



Master Thesis

im Rahmen des
Universitätslehrganges „Geographical Information Science & Systems“
(UNIGIS MSc) am Zentrum für GeoInformatik (Z_GIS)
der Paris Lodron-Universität Salzburg

zum Thema

Entwicklung einer Webapplikation zur Dokumentation von Laserscanning-Projekten und Visualisierung von Laserscanning-Daten

vorgelegt von

Mario Schedler
U1392, UNIGIS MSc Jahrgang 2008

Zur Erlangung des Grades
„Master of Science (Geographical Information Science & Systems) – MSc(GIS)“

Gutachter:
Ao. Univ. Prof. Dr. Josef Strobl

Halle, 6. Oktober 2010

Danksagung

An dieser Stelle möchte ich mich bei den Personen bedanken, die zum Gelingen dieser Masterthesis beigetragen und mich bei der Entstehung dieser Arbeit unterstützt haben. Mein spezieller Dank gilt deshalb:

- Ao. Univ. Prof. Dr. Josef Strobl, UNIGIS-Lehrgangleiter an der Universität Salzburg und dem gesamten UNIGIS-Team,
- Klaus Engelmann und Romy Ludwig für die Unterstützung bei der Dow Chemical Company und
- meinem Kommilitonen Marco Seyße für die aufmunternden Gespräche und Diskussionen nicht nur während der Zeit der Masterthesis.

Ganz besonders möchte ich mich bei meiner Frau Juliane bedanken. Sie hat mich während des gesamten Zeitraums, in dem ich die Masterthesis geschrieben habe – in der Sie zusätzlich ihre Diplomprüfungen absolvierte und in das Berufsleben eintauchte – vehement unterstützt. Auch bei meiner Familie und meinen Freunden möchte ich mich für die Rücksichtnahme während dieser Phase ganz herzlich bedanken.

Erklärung über die eigenständige Arbeit

Ich versichere, diese Masterthesis ohne fremde Hilfe und ohne Verwendung anderer als der angeführten Quellen angefertigt zu haben, und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat. Alle Ausführungen der Arbeit die wörtlich oder sinngemäß übernommen wurden sind entsprechend gekennzeichnet.

Ort und Datum

Unterschrift

Kurzfassung

Terrestrisches Laserscanning wird seit etwa zehn Jahren in verschiedenen Bereichen wie beispielsweise der Archäologie, dem Anlagen- und Maschinenbau, dem Bauwesen, der Denkmalpflege und der Architektur zur Datenerfassung eingesetzt. Die aus dem Laserscanning resultierenden Punktwolken können auf verschiedene Weise und mit verschiedenen Softwareprodukten visualisiert und verarbeitet werden. Einige dieser Softwareprodukte ermöglichen das Betrachten von Laserscanning-Daten in Form eines Panoramas im World Wide Web mithilfe eines Webbrowsers.

Auf die Panoramen kann durch Übersichtskarten zugegriffen werden. Übersichtskarten sind häufig verweissensitive Grafiken, sogenannte Image Maps oder CAD-Dateien. Sie stellen sich in verschiedenen Bereichen als sehr unflexibel dar, weisen kartografische Defizite und eine geringe Usability auf. An diesen Punkten setzt diese Masterthesis an: auf Grundlage einer Anforderungsanalyse wird eine Webapplikation entwickelt, die versucht die Nachteile der bisherigen Lösungen zu beseitigen. Hierbei werden beispielsweise aktuelle JavaScript-Frameworks wie GeoExt und OpenLayers sowie Basiskarten von Google Maps und Bing Maps verwendet und somit ein ansprechendes User Interface im Sinne des „Web 2.0“ entwickelt. Das Ziel dieser Masterthesis ist somit die Entwicklung einer Webapplikation mit der auf Laserscanning-Daten zugegriffen werden kann, Laserscanning-Daten verwaltet und Daten von Laserscanning-Projekten abgerufen werden können.

Schlussendlich soll diese Webapplikation bei der Dow Chemical Company eingeführt werden um Laserscanning-Daten zu verwalten und auf diese mithilfe von webbasierten Viewern, wie beispielsweise TruView von Leica Geosystems, zuzugreifen.

Abstract

Terrestrial laser scanning has been established as a data acquisition technology in many areas such as archaeology, plant engineering, mechanical engineering, civil engineering, monument preservation and architecture during the last decade. The data collected using laser scanning technology are called point clouds. These point clouds can be visualized and processed with different software. Some of these software products enable panoramic viewing of the point clouds using a simple web browser like Microsoft's Internet Explorer.

The access to these panoramas is provided by overview maps which are almost all HTML image maps or CAD files. Due to inflexibility, missing cartographic principles and poor usability, overview maps could not be used in each area where laser scanning was performed. This is where this master thesis becomes necessary. Based on a requirements analysis, a web application will be developed trying to improve current solutions. For development purposes JavaScript frameworks such as GeoExt and OpenLayers and maps from web services like Google Maps or Bing Maps will be integrated in the web application to achieve a pleasing and appealing user interface according to „Web 2.0“. This master thesis aims at developing a web application that provides capabilities to manage laser scanning data, shows information related to laser scanning projects and enables the visualization of laser scanning data.

Finally, the web application is to be released at the Dow Chemical Company to manage and access laser scanning data using a web-based viewer such as TruView from Leica Geosystems.

Inhaltsverzeichnis

Abbildungsverzeichnis	XI
Tabellenverzeichnis	XIII
Quellcode-Verzeichnis.....	XIV
Abkürzungsverzeichnis	XV
1 Einführung	1
1.1 Motivation	1
1.2 Ziele und deren Realisierung.....	3
1.3 Nicht behandelte Themen.....	4
1.4 Beabsichtigtes Publikum	4
1.5 Struktur der Masterthesis.....	5
2 Theoretische und technische Grundlagen	7
2.1 Web-Mapping.....	7
2.1.1 Begriffsbestimmung	7
2.1.2 Klassifikation von Internet-GIS	8
2.1.3 Architektur.....	10
2.1.4 Datenspeicherung und Datenaustausch	11
2.1.4.1 Datenbankbasierte Speicherung.....	11
2.1.4.2 Dateibasierte Speicherung.....	12
2.1.5 Tools zum Erstellen von Web-Mapping Applikationen.....	13
2.1.6 JavaScript-Frameworks	14
2.1.6.1 OpenLayers	14

2.1.6.2	Ext JS	15
2.1.6.3	GeoExt.....	16
2.2	Terrestrisches Laserscanning.....	17
2.2.1	Begriffsbestimmung und Funktionsweise	17
2.2.2	Visualisierung von Laserscanning-Daten.....	18
2.2.2.1	Punktwolken.....	19
2.2.2.2	Orthofotos	20
2.2.2.3	Panoramabilder	20
2.2.3	Viewer-Software.....	21
2.2.3.1	TruView	21
2.2.3.2	PlantView	23
2.2.3.3	LFM NetView	25
2.3	Metadaten	26
2.3.1	Begriffsbestimmung	27
2.3.2	Standardisierung von Metadaten	27
2.3.2.1	Allgemeines.....	27
2.3.2.2	ISO 19115	29
2.3.2.3	Dublin Core.....	31
3	Anforderungen an die Webapplikation	33
3.1	Nutzeranalyse	33
3.2	Anwendungsfälle der Webapplikation	34
3.3	Anforderungen im Detail.....	37
3.3.1	Allgemeines	37
3.3.2	User Interface	38
3.3.3	Anforderungen an die Daten.....	38
3.3.4	TruView.....	39
3.4	Resultat der Anforderungsanalyse.....	39

4	Datenmodellierung	41
4.1	Datenmodellierung für den Prototyp	41
4.2	Konzeptuelles Datenmodell.....	41
4.2.1	Entity-Relationship Model	42
4.2.2	Gemeinsame Attribute der Entitäten	43
4.2.3	Entität Laser Scanner Position.....	44
4.2.4	Entität Area.....	44
4.2.5	Entität Database	44
4.2.6	Entität Person.....	45
4.2.7	Entität Laser Scanning Project	45
4.2.8	Entitäten Side und Continent.....	46
4.2.9	Entität Layer	46
4.3	Logisches und physisches Datenmodell.....	47
5	Die Webapplikation	50
5.1	Grundgedanken.....	50
5.1.1	JavaScript-Frameworks und der Prototyp	50
5.1.2	Basislayer und Overlays	51
5.1.3	Bezugssystem	52
5.1.4	Laserscanner-Standpunkte.....	53
5.1.5	Datenabfrage.....	53
5.2	Struktur der Webapplikation	54
5.3	Prinzipielle Funktionsweise.....	55
5.4	User Interface	57
5.4.1	Layout und Ext JS	57
5.4.2	Region North	59
5.4.3	Region West	59
5.4.3.1	Projects-Tab	59
5.4.3.2	Maps-Tab	61

5.4.3.3	Symbolik	62
5.4.4	Region Center	63
5.4.5	Region East.....	64
5.5	Der Prototyp	65
6	Abschließende Betrachtungen	67
6.1	Zusammenfassung	67
6.2	Ergebnisse.....	68
6.3	Ausblick.....	69
	Literaturverzeichnis	71
	Anhang.....	77

Abbildungsverzeichnis

Abbildung 1-1: Verweissensitive Karte mit wenigen Laserscanner-Standpunkten (Quelle: LEICA 2009a).....	2
Abbildung 1-2: Verweissensitive Karte mit vielen Laserscanner-Standpunkten (Quelle: DOW 2009a)	2
Abbildung 1-3: CAD-Datei als Übersichtskarte (Quelle: Z+F UK 2009)	3
Abbildung 2-1: Klassifikation von <i>Web Maps</i> nach KRAAK (2001).....	9
Abbildung 2-2: Client-Server Architektur.....	11
Abbildung 2-3: User Interfaces OpenLayers (Quelle: http://www.openlayers.com).....	15
Abbildung 2-4: User Interfaces ExtJS (Quelle: http://www.sencha.com/products/js/)..	16
Abbildung 2-5: User Interfaces GeoExt (Quelle: http://www.geoext.org)	16
Abbildung 2-6: Funktionsprinzip des Laserscanning (Quelle: FARO 2010).....	18
Abbildung 2-7: Flächenhafte Darstellung der Punkt-wolke, die in Abbildung 2-8 vergrößert ist (Rahmen)	20
Abbildung 2-8: Verlust der flächenhaften Darstellung durch Vergrößerung des Bereiches aus Abbildung 2-7	20
Abbildung 2-9: Farbiges TruView in der Version 2.0 (Quelle: http://www.bsp-osnabrueck.de)	22
Abbildung 2-10: Navigationsoberfläche PlantView (Quelle: DOW 2009b)	24
Abbildung 2-11: LFM NetView mit Übersichtsplan (Key Plan) (Quelle: Z+F UK 2010d)	25
Abbildung 3-1: Anwendungsfalldiagramm der Webapplikation	35
Abbildung 4-1: Entity-Relationship Model der Webapplikation	42
Abbildung 4-2: Logisches Datenmodell für eine MySQL Datenbank	48
Abbildung 5-1: Struktur der Dateien für die Webapplikation.....	54
Abbildung 5-2: Client-Server Prinzip der Webapplikation.....	56

Abbildung 5-3: Funktionsweise der Webapplikation (Client-Server und mobile Lösung)	57
Abbildung 5-4: Grundgerüst der Webapplikation im Border Layout	58
Abbildung 5-5: Nördlicher Bereich der Webapplikation mit Auswahlliste zur Suche von geographischen Namen	59
Abbildung 5-6: Baumstruktur für den Zugriff auf die Laserscanning-Daten.....	60
Abbildung 5-7: Baumstruktur für die Projekt-bezogenen Layer.....	61
Abbildung 5-8: Kartenfenster mit Laserscanner-Standpunkten und einem Basislayer von Google Maps	63
Abbildung 5-9: Daten eines Laserscanning-Projekts in Region East.....	64
Abbildung 5-10: Hinweisfenster der Webapplikation.....	65
Abbildung 5-11: Prototyp des Laser Scanning Information Systems im Internet Explorer	66

Tabellenverzeichnis

Tabelle 2-1: TruView Funktionen (Quelle: LEICA 2009b).....	23
Tabelle 2-2: LFM NetView Funktionen (Quelle: Z+F 2010b)	26
Tabelle 2-3: Kernelemente des ISO 19115 Standards (Quelle: ISO/TC 211 2003; KST. GDI-DE 2008).....	30
Tabelle 2-4: Dublin Core Metadata Element Set Version 1.1 (Quelle: DCMI 2009b, WIKIPEDIA 2009)	31
Tabelle 3-1: Beschreibung des Use Case „Suche nach Laserscanning-Daten“ (UseCase1)	35
Tabelle 3-2: Beschreibung des Use Case „Informieren über Laserscanning-Daten und - Projekte“ (UseCase2)	35
Tabelle 3-3: Beschreibung des Use Case „Visualisieren von Laserscanning-Daten mittels eines Viewers“ (UseCase3).....	36
Tabelle 3-4: Beschreibung des Use Case „Administration der Webapplikation“ (UseCase4).....	37
Tabelle 3-5: Anforderungen der User an die Webapplikation	40
Tabelle 4-1: Gemeinsame Attribute aller Entitäten der Webapplikation	43
Tabelle 4-2: Zusätzliche Attribute der Entität Laser Scanner Position	44
Tabelle 4-3: Zusätzliche Attribute der Entität Database	45
Tabelle 4-4: Zusätzliche Attribute der Entität Person	45
Tabelle 4-5: Zusätzliche Attribute der Entität Layer.....	46
Tabelle 4-6: Mögliche CHECK-Bedingungen im Datenmodell	49
Tabelle 5-1: Kartendienste und deren Bezeichnung in der Webapplikation.....	51
Tabelle 5-2: Darstellung der Laserscanner-Standpunkte in der Karte	53
Tabelle 5-3: Verwendete Symbole in den Baumstrukturen.....	62

Quellcode-Verzeichnis

Listing 2-1: Gegenüberstellung XML-Format (links) und JSON-Format (rechts)	12
Listing 2-2: Beispiel einer XHTML-Notation von DC-Metadaten	32
Listing 5-1: Metadaten-Elemente der Startseite der Webapplikation	55

Abkürzungsverzeichnis

AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
BSD	Berkeley Software Distribution
CAD	Computer Aided Design
CRS	Coordinate Reference System
CSV	Comma Separated Values
DBMS	Datenbank-Managementsystem
DC	Dublin Core
DCMI	Dublin Core Metadata Initiative
DXF	Drawing Interchange Format
EPSG	European Petroleum Survey Group
ERD	Entity Relationship Diagram
ERM	Entity-Relationship Modell
GeoRSS	Geographically Encoded Objects for RSS Feeds
GML	Geography Markup Language
HTML	HyperText Markup Language
HTTPS	HyperText Transfer Protocol Secure
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
KML	Keyhole Markup Language
LFM	Light Form Modeller
MIME	Multipurpose Internet Mail Extensions
OGC	Open Geospatial Consortium
PHP	PHP Hypertext Preprocessor
PNG	Portable Network Graphics
RDF	Resource Description Framework

RIA	Rich Internet Application
RSS	Really Simple Syndication (RSS 2.0)
SQL	Structured Query Language
TIFF	Tagged Image File Format
UML	Unified Modeling Language
URL	Uniform Resource Locator
WMS	Web Map Service
XML	eXtensible Markup Language

1 Einführung

Terrestrisches Laserscanning wird seit etwa zehn Jahren in verschiedenen Bereichen wie zum Beispiel der Archäologie, der Denkmalpflege, dem Bauwesen, dem Anlagen- und Maschinenbau und der Architektur zur Datenerfassung eingesetzt. Die aus dem Laserscanning resultierenden Punktwolken können auf verschiedene Weise und mit verschiedenen Softwareprodukten visualisiert und verarbeitet werden. Einige dieser Softwareprodukte ermöglichen das Betrachten von Laserscanning-Daten in Form eines Panoramas im World Wide Web mithilfe eines Browsers. Die Verwendung solcher webbasierter Software erfordert beim Anwender kein Expertenwissen im Bereich des Laserscannings.

Der Zugriff auf die Panoramen erfolgt durch Übersichtskarten wie verweissensitive Grafiken (Image Maps) oder CAD-Dateien in den die Standpunkte des Laserscanners dargestellt und verlinkt sind. Die Übersichtskarten beinhalten in der Regel mehrere Standpunkte eines Laserscanners und stellen so ein einfaches Werkzeug für den Zugriff auf die Laserscanning-Daten dar. Häufig werden die Übersichtskarten aus den Laserscanning-Daten generiert. Die Perspektive der Übersichtskarte kann dabei frei gewählt werden, so dass neben der klassischen Draufsicht auch isometrische Ansichten verwendet werden können. In verschiedenen Bereichen stellen sich Übersichtskarten als sehr unflexibel dar, weisen kartografische Defizite und eine geringe Usability auf (siehe Kapitel 1.1).

1.1 Motivation

Die Darstellung der Laserscanner-Standpunkte in einer verweissensitiven Grafik ist für Projekte mit einer sehr geringen räumlichen Ausdehnung sowie einer niedrigen und homogenen Standpunktdichte ausreichend (siehe Abbildung 1-1). Sofern aber größere Bereiche mit sehr vielen Standpunkten oder Bereiche mit einer heterogenen Standpunktverteilung abgebildet werden sollen, sind verweissensitive Karten nicht mehr

geeignet (siehe Abbildung 1-2). Ein weiterer Nachteil verweissensitiver Grafiken ist, dass bei isometrischen Ansichten bzw. Draufsichten nur sehr unzureichend erkannt wird, ob sich ein Standpunkt in Bodennähe oder beispielsweise auf einem Gebäude befindet. Die Abbildung 1-1 und Abbildung 1-2 stellen Laserscanner-Standpunkte, die durch gelbe Dreiecke symbolisiert werden, in verweissensitive Grafiken am Beispiel der Software TruView dar. Bei Abbildung 1-1 handelt es sich um eine isometrische Darstellung, Abbildung 1-2 ist eine Draufsicht auf eine Industrieanlage mit sehr vielen und heterogen verteilten Standpunkten. Es wird deutlich, dass sich die Standpunkte in Abbildung 1-2 teilweise überlagern und man könnte vermuten, dass sich die Standpunkte in Bodennähe befinden. Tatsächlich lässt die Karte in Abbildung 1-2 aber nicht erkennen, wo sich ein Standpunkt in der Örtlichkeit befindet.



Abbildung 1-1: Verweissensitive Karte mit wenigen Laserscanner-Standpunkten (Quelle: LEICA 2009a)

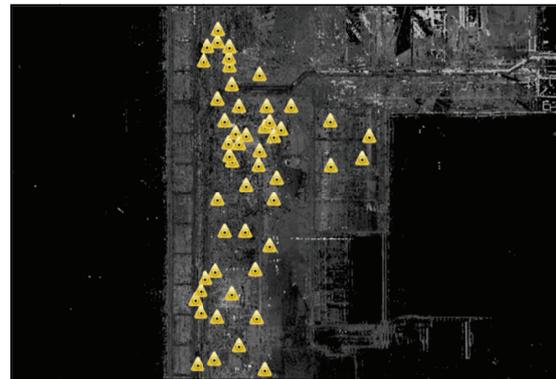


Abbildung 1-2: Verweissensitive Karte mit vielen Laserscanner-Standpunkten (Quelle: DOW 2009a)

Der in Abbildung 1-2 dargestellte Bereich ist nur ein Teil eines Projekts, das sich über eine größere Fläche erstreckt. Schon in dieser Grafik werden Laserscanner-Standpunkte verdeckt. Würden alle Standpunkte des Projektes in einer verweissensitiven Grafik abgebildet werden, wäre die Grafik entweder exorbitant groß, so dass am Bildschirm sehr viel gescrollt werden müsste, um an die gewünschten Daten zu gelangen oder die Grafik wäre sehr klein und Laserscanner-Standpunkte wären verdeckt und eine Suche nach Daten wäre nur eingeschränkt möglich.

Verweissensitive Karten sind auf Grund ihrer statischen Natur nicht veränderbar. Sofern Teilbereiche auf Grund von Umbauarbeiten oder Ähnlichem neu gescannt werden, muss die gesamte Karte neu generiert werden. Es ist nicht ohne weiteres möglich einen neuen Standpunkt in eine statische Karte zu integrieren, da eine solche Karte letztlich den gescannten Bereich abbildet und dieser auf Grund der Veränderungen nicht mehr identisch mit der vorherigen Situation ist. Bei dicht beieinanderliegenden Laserscanner-Standpunkten fehlt eine Zoom-Funktion.

Anstelle von verweissensitiven Karten können auch CAD-Pläne verwendet werden um auf die Laserscanner-Standpunkte zuzugreifen. Diese CAD-Pläne sind dann häufig in Websites eingebettet. Sie können zwar einfacher aktualisiert werden, als es mit verweissensitiven Karten möglich ist, jedoch weisen diese Übersichtskarten eine ähnlich schlechte Usability wie verweissensitive Karten und auch erhebliche kartographische Mängel auf. In den Abbildung 1-3 und Abbildung 2-11 verdrängen die Laserscanner-Standpunkte fast die gesamten Elemente der Übersichtskarte. Zudem ermöglichen es bisherige webbasierte Lösungen nicht, mehrere Laserscanning-Projekte mit einer Applikation zu verwalten; stattdessen muss für jedes Projekt eine neue Übersichtskarte erzeugt werden. Dies kann bei mehreren Projekten zu einem hohen Aufwand sowohl bei der Generierung der Übersichtskarten als auch bei der Verwaltung der Projekte führen. Eine Webapplikation, die an den beschriebenen Kritikpunkten ansetzt und auch global für die Dokumentation von Laserscanning-Projekten und der Visualisierung von Laserscanning-Daten eingesetzt werden kann, vereinfacht das Arbeiten mit solchen Daten und gestaltet dies effizienter.

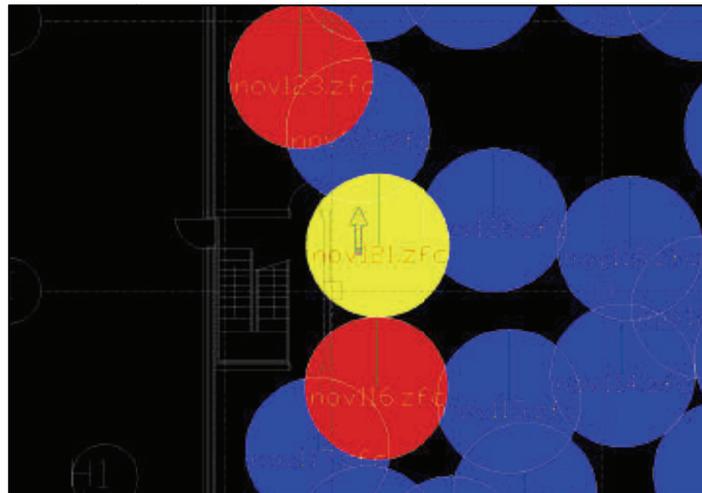


Abbildung 1-3: CAD-Datei als Übersichtskarte (Quelle: Z+F UK 2009)

1.2 Ziele und deren Realisierung

Ziel dieser Masterthesis ist die Entwicklung einer Webapplikation zur Dokumentation von Laserscanning-Projekten und Visualisierung von terrestrischen Laserscanning-Daten, die die unter Punkt 1.1 aufgezählten Nachteile beseitigt. Um diese Masterthesis zu realisieren werden theoretische und technische Grundlagen erarbeitet und diskutiert. Zudem wird dargestellt, wie diese Grundlagen im Rahmen der Masterthesis angewendet werden können. Dabei soll die Visualisierung der Laserscanning-Daten mittels beste-

hender Viewer-Software, die in einem Browser, wie beispielsweise dem Microsoft Internet Explorer, ausgeführt werden können, erfolgen. Die Webapplikation soll über alle Laserscanning-Projekte informieren und als zentrale Anlaufstelle fungieren sowie Fragen, nach dem Wann, Wo und Wie Laserscanning-Projekte durchgeführt wurden, beantworten. Mithilfe der Webapplikation sollen Laserscanning-Daten einem größeren Benutzerkreis zur Verfügung gestellt werden. Die Webapplikation soll dabei intuitiv bedienbar sein und ohne spezielle Fertigkeiten im Bereich des Laserscannings genutzt werden können.

Die während dieser Masterthesis entwickelte Webapplikation wird prototypisch bei der Dow Chemical Company (Dow) als *Laser Scanning Information System* eingeführt und getestet. Hierfür stehen Laserscanning-Daten aus bereits erfolgten Projekten bei Dow zur Verfügung, die in die Webapplikation überführt werden. Als webbasierter Viewer wird TruView von Leica Geosystems eingesetzt.

1.3 Nicht behandelte Themen

Bei den theoretischen Betrachtungen wird auf die Grundlagen des Internets wie beispielsweise das World Wide Web, Internet-Protokolle, HyperText Markup Language, Cascading Style Sheets und Webbrowser nicht eingegangen. Lediglich das Client-Server Prinzip wird kurz erläutert. Es wird vielmehr davon ausgegangen, dass die angestrebte Zielgruppe der Masterthesis über ausreichend Grundwissen in diesen Bereichen verfügt.

Bei der Entwicklung des Prototyps liegt das Hauptaugenmerk auf der Implementierung grundlegender Funktionalität – auf die Implementierung darüber hinausgehender Funktionalität wird verzichtet. Dies würde den zeitlichen Rahmen der Masterthesis sprengen und kann zudem durch zukünftige Updates realisiert werden.

1.4 Beabsichtigtes Publikum

Diese Masterthesis ist im Wesentlichen an zwei Personengruppen adressiert: zum einen an Personen, die sich mit der Entwicklung von Webapplikationen beschäftigen (Webentwickler) und zum anderen an die zukünftigen Nutzer der Webapplikation (User), die sich für die Entwicklung der Webapplikation interessieren und gegebenenfalls eigene Projekte initiieren wollen. Zu den Webentwicklern werden auch Personen gezählt, die keine oder nur rudimentäre Kenntnisse im Bereich des Laserscannings

aufweisen, sich aber mit der Anwendungsentwicklung im World Wide Web (WWW) beschäftigen.

User sind in der Regel Personen die Laserscanning als Datenerfassungsmethode verwenden und eine Möglichkeit suchen allen Projektbeteiligten die Laserscanning-Daten mittels einer Webapplikation zur Verfügung zu stellen. Aber auch Nutzer, die ihre Daten mittels eines Browsers verwalten möchten, zählen zu diesem Personenkreis. Einer der wichtigsten User ist die Dow Chemical Company, bei der die Webapplikation getestet und im Anschluss an die Masterthesis als Software zum Verwalten von Laserscanning-Daten eingesetzt werden soll. Hier partizipieren insbesondere Anlagenplaner, die Instandhaltung, Anlagenbetreiber, Facility Management und das Management von Infrastruktureinrichtungen von der Webapplikation.

1.5 Struktur der Masterthesis

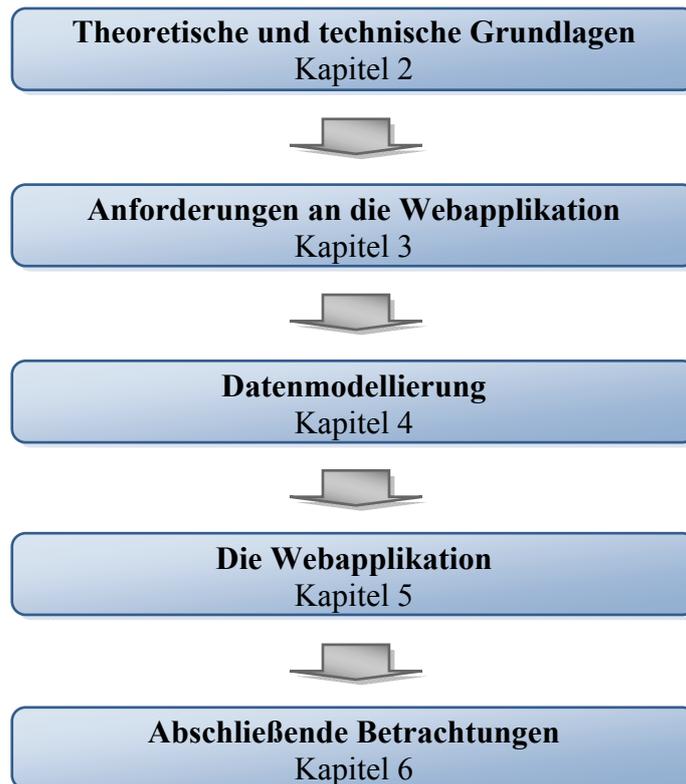
Die vorliegende Masterthesis gliedert sich im Wesentlichen in fünf Kapitel. Sie beginnt mit der Erläuterung der theoretischen und technischen Grundlagen in *Kapitel 2*. Diesem Abschnitt kommt große Bedeutung für das Verständnis der Masterthesis zu. Es wird ein Überblick über die Bereiche Internet-GIS, Laserscanning und Metadaten im Kontext einer Webapplikation bzw. dem World Wide Web gegeben.

Der Frage nach den Anforderungen, die eine solche Webapplikation mit sich bringen muss, wird im *Kapitel 3* nachgegangen. Hier wird der zukünftige User und dessen Bedürfnisse an die Webapplikation beleuchtet, um die Webapplikation effizient und benutzerfreundlich zu entwickeln.

In *Kapitel 4* wird die Datenmodellierung für den Prototyp beschrieben. Es wird erläutert, welche Entitäten und Attribute notwendig sind und wie die verschiedenen Entitäten in Beziehung stehen. Weiterhin wird beschrieben, wie aus diesem konzeptuellen Datenmodell ein logisches Datenmodell und schließlich ein physisches Datenmodell entwickelt werden.

Die eigentliche Entwicklung der Webapplikation wird im *Kapitel 5* abgehandelt. Hier werden der Aufbau und die Funktionsweise beschrieben sowie das User Interface und dessen Umsetzung im Detail erläutert. Anschließend wird der Prototyp vorgestellt.

Abschließend wird im *Kapitel 6* der beschrittene Weg bis zur Webapplikation betrachtet, erörtert und mit den in *Kapitel 1* gestellten Zielen verglichen. Im Ausblick werden dann Optimierungsmöglichkeiten für die Webapplikation aufgezeigt.



2 Theoretische und technische Grundlagen

Kapitel 2 widmet sich den theoretischen und technischen Grundlagen, die für die Bearbeitung der Masterthesis bedeutend sind. Im ersten Abschnitt wird auf das Thema Internet-GIS bzw. Web-Mapping eingegangen. Hierbei wird eine allgemeine Einführung in die Thematik gegeben und anschließend werden die Methoden und Werkzeuge vorgestellt, die während der Entwicklungsphase des Prototyps eingesetzt werden. Der zweite Abschnitt behandelt das Thema terrestrisches Laserscanning. Es wird die Funktionsweise des Messverfahrens erläutert und auf die Software, die zur Visualisierung von Laserscanning-Daten benötigt wird, näher eingegangen. Im dritten und letzten Abschnitt wird das Thema Metadaten behandelt. Es wird definiert was Metadaten sind und welchen Zweck sie erfüllen sowie die verschiedenen Metadaten-Standards kurz vorgestellt.

2.1 Web-Mapping

In diesem Abschnitt wird das Thema Web-Mapping behandelt. Neben der Darstellung zahlreicher Begrifflichkeiten und Klassifikationsmöglichkeiten in diesem Bereich, wird auch auf die Architektur von Web-Mapping Applikationen eingegangen. Weiterhin wird erläutert wie die Daten in einem Internet-GIS gespeichert und über einen Server ausgetauscht werden können. Abschließend werden Tools vorgestellt, die es ermöglichen eine Webapplikation zur Verwaltung und Visualisierung von terrestrischen Laserscanning-Daten zu entwickeln.

2.1.1 Begriffsbestimmung

Für die Darstellung von Geoinformationen bzw. digitalen Karten im Internet existieren in der Literatur zahlreiche Begriffe wie *Internet-GIS*, *Rich Web-Mapping Application*, *Mapserver*, *Web-GIS*, *Online-GIS*, *web-based Map*, *Web-Karte* und *Internet-Karte*.

Einige dieser Begriffe wie beispielsweise *Internet-GIS* beziehen sich mehr auf die Gesamtheit der Dienste des Internets. Wohingegen sich andere Begriffe wie *Web-GIS* mehr auf einen einzigen Dienst des Internets, das World Wide Web und die Darstellung bzw. Bearbeitung von Geoinformationen in einem Browser beziehen. Der Begriff GIS, wie er beispielsweise in *Web-GIS* vorkommt, könnte neben der bloßen Darstellung und Ansichts-Manipulationen von Karten auch auf zusätzliche Funktionalitäten wie Datenbank gestützte Abfragen, GIS-Operationen etc. hindeuten (DICKMANN 2004). Unter *Internet-GIS* verstehen KORDUAN & ZEHNER (2008) alles, was auf der Internet-Technologie basiert und Geoinformationen darstellen, verteilen und bearbeiten kann. Viele der eingangs aufgeführten Begriffe, können unter diesem Begriff subsummiert werden. Diese Definition ist sehr umfassend und sie bezieht sich auf alles, was mit Geoinformationen und dem Internet in Verbindung steht. Für PENG & TSOU (2003) ist *Internet-GIS* ein netzwerkbasierendes GIS, das das Internet verwendet um auf Geoinformationen und Geoprocessing-Tools zuzugreifen.

Begriffe wie *Web-GIS*, *Internet-GIS* etc. beziehen sich mehr auf Karten – Begriffe wie *Web-Mapping* oder *Web-Kartographie* hingegen, beziehen sich mehr auf den Herstellungsprozess digitaler Karten. Häufig wird hierunter mehr verstanden als das kartographische Gestalten und das anschließende Publizieren von Karten im Internet (DICKMANN 2004). Es wird deutlich, wie unterschiedlich Geoinformationen bzw. digitale Karten im Kontext von Internet und Web bezeichnet werden. Im Rahmen dieser Masterthesis müssen einige Festlegungen bezüglich der Begrifflichkeiten getroffen werden: alle Geoinformationen im Internet werden im Sinne von KORDUAN & ZEHNER (2008) als Internet-GIS verstanden. Der Begriff Web-Mapping wird im Zusammenhang mit der Herstellung von solchen Internet-GIS gebraucht. Der zu entwickelnde Prototyp wird allerdings nicht als Internet-GIS sondern als Webapplikation bezeichnet, da dieser Begriff vom beabsichtigten Publikum besser verstanden wird.

2.1.2 Klassifikation von Internet-GIS

Neben den zahlreichen Begrifflichkeiten, existieren auch zahlreiche Klassifikationen für die Darstellung von Geoinformationen bzw. digitalen Karten im Internet. Diese Klassifikationen können nach verschiedenen Kriterien wie Interaktionsgrad, Technologie, Anwendungsbereich, Funktionsumfang etc. erfolgen, wobei häufig nach der erwarteten Funktionalität klassifiziert wird (KORDUAN & ZEHNER 2008).

PENG & TSOU (2003) sprechen im Kontext von verteilten GI-Systemen auch von einer technologischen Evolution dieser Systeme. Sie beziehen sich dabei auf die Funktionalität und Interaktivität von verteilten GIS, die beispielsweise bei verweissensitiven Karten niedrig und bei Anwendung von Java-Applets relativ hoch ist. Im Folgenden werden exemplarisch einige Klassifikationen vorgestellt, um den Begriff Internet-GIS deutlicher zu präsentieren. Auf eine ausführliche Darstellung der verschiedenen Klassifikationen wird verzichtet, es wird vielmehr auf die Literatur wie beispielsweise FITZKE (1999), ASCHE (2001), PENG & TSOU (2003), MITCHELL (2005) verwiesen.

KRAAK (2001) verwendet den Begriff *Web Maps*, den er ausschließlich auf das World Wide Web bezieht und unterteilt diese in statische Karten (*static maps*) und dynamische Karten (*dynamic maps*). Beide Kategorien unterteilt er wiederum in Karten, die nur betrachtet werden können (*view only*) und Karten mit interaktiver Schnittstelle und/oder interaktivem Inhalt (*interactive interface and/or contents*). Er fügt dieser Klassifizierung eine zeitliche Abhängigkeit aufgrund der schnellen Entwicklung des WWW hinzu und versucht mit dieser Klassifikation aufzeigen, wie Karten genutzt werden können. Die Webapplikation ist demnach eine *statische Karte* mit interaktiver Schnittstelle und interaktivem Inhalt.

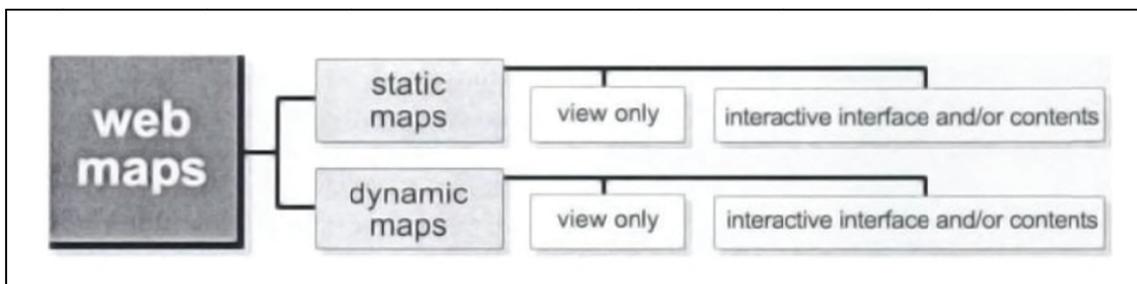


Abbildung 2-1: Klassifikation von *Web Maps* nach KRAAK (2001)

Eine weitere Einteilung zur Darstellung von raumbezogenen Daten im Internet wird von DICKMANN (2004) vorgenommen. Er unterteilt elektronische, also digitale Karten, in *Kartenvorstufen*, *statische Karten* und *interaktive Karten*. *Kartenvorstufen* sind digitale Rohdaten, die sich nur mit geeigneter Software wie beispielsweise einem GIS nutzen lassen. Unter *statischen Karten* versteht er vorgefertigte Karten, die sich mit gängigen Browsern auf dem Bildschirm darstellen lassen. *Interaktive Karten* werden in verweissensitive Karten (*sensitive Karten*), individuell generierbare Karten (*maps on demand*), animierte Karten zur Darstellung von dynamischen Prozessen (*animated maps*), räumliche Darstellungen (*3D maps*) und Karten als Resultat von GIS-Analysen (*Web-*

GIS) untergliedert. Die Webapplikation kann demnach als *interaktive Karte* klassifiziert werden.

KORDUAN & ZEHNER (2008) klassifizieren Internet-GIS unter anderem hinsichtlich der Anwendungsbereiche in *einfache Auskunftssysteme*, *spezialisierte georeferenzierte Auskunftssysteme*, *internetbasierte Geoinformatik-Clients* und *Geodatenportale*. Die *einfachen Auskunftssysteme* laufen in einem Standard-Browser oder mithilfe eines üblichen Plug-Ins, erzeugen interaktive Karten und dienen weiterhin der Präsentation von Sachdaten. *Spezialisierte georeferenzierte Auskunftssysteme* sind dynamische Systeme, die einem Personenkreis Dienste wie beispielsweise Adresssuchen bereitstellen. *Internetbasierte Geoinformatik-Clients* besitzen erweiterte Funktionalität und ermöglichen somit komplexe Analysen, Export von Daten und das Ändern von Sach- und Geometriedaten. Mit *Geodatenportalen* werden Daten, meist von unterschiedlichen Geodatenservern bereitgestellt. Die Webapplikation, die in dieser Arbeit entwickelt wird, kann hiernach in die Kategorie der *einfachen Auskunftssysteme* eingeordnet werden.

2.1.3 Architektur

Internet-GIS basiert in der Regel auf dem Client-Server-Prinzip. Der Server ist hierbei ein Rechner oder ein Programm, der/das Anfragen von Clients entgegen nimmt und diese beantwortet (BILL 1999). Im Kontext von Internet-GIS bieten Server den Clients vielfältige Dienste wie beispielsweise Druckausgabe, Koordinatentransformationen und Raster- und Vektordatenauslieferung an. Bei der Auslieferung von digitalen Karten werden vorrangig Rasterdaten verwendet. Diese Karten können auf dem Server in Dateiform (z. B. im JPG- oder PNG-Format) bereits vorliegen und bei Anfragen des Clients in diesem Format ausgeliefert werden. Diese Art der Datenspeicherung liegt auch bei verweissensitiven Karten vor. Die Rasterdaten können weiterhin in Echtzeit durch einen Mapserver oder mittels Standards des Open Geospatial Consortiums (OGC) beispielsweise durch einen Web Map Service (WMS) generiert werden. Ein WMS ist eine einfache HTTP-Schnittstelle mit der georeferenzierte Karten von einer oder mehreren räumlichen Datenbanken angefragt werden können (OGC 2010).

Der Client ist ein Rechner oder ein Programm, der/das Anfragen an den Server sendet und auf die angefragten Daten wartet (BILL 1999). Im Bereich des Web-Mapping werden Webbrowser als Clients verwendet um die digitalen Karten bei Servern anzufragen und schließlich darzustellen. Diese Daten können durch unterschiedliche

Technologien wie HTML, JavaScript, SVG, Flash, Java-Applets etc. wiedergegeben werden.

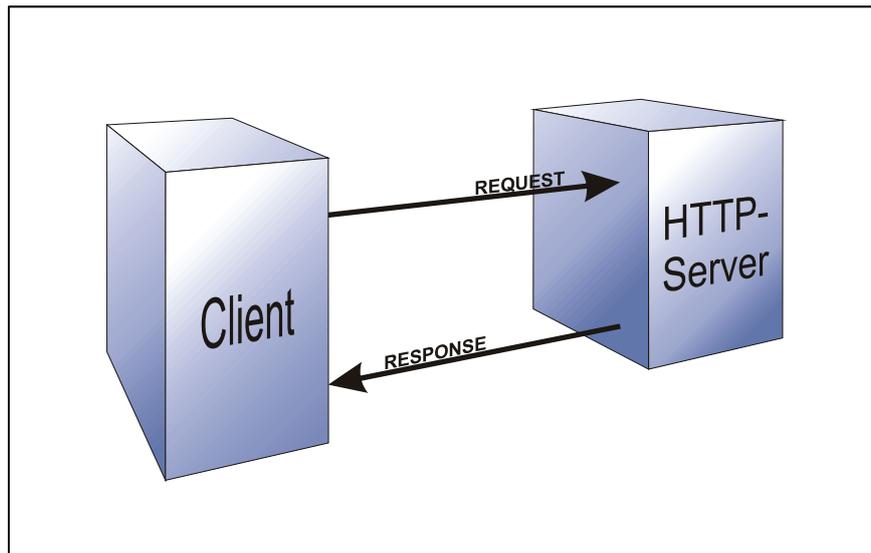


Abbildung 2-2: Client-Server Architektur

2.1.4 Datenspeicherung und Datenaustausch

Geoinformationen können in einem Internet-GIS sowohl client- als auch serverseitig gespeichert werden, wobei die Daten eines Internet-GIS im Allgemeinen auf einem Server gespeichert und von diesem auch ausgeliefert werden (KORDUAN & ZEHNER 2008). Die Speicherung der Daten kann sowohl datenbankbasiert als auch dateibasiert geschehen. In den folgenden beiden Abschnitten werden diese Speichermöglichkeiten kurz vorgestellt.

2.1.4.1 Datenbankbasierte Speicherung

Vor allem bei großen und komplexen GI-Systemen werden die Daten in einer Geodatenbanken gespeichert (LONGLEY et al. 2005) und von Datenbank-Management-systemen (DBMS) mit einer räumlichen Erweiterung (Spatial Extension) verwaltet. Diese DBMS werden dann als *GeoDBMS* oder *räumliche DBMS* bezeichnet (vgl. BRINKHOFF 2005). Die Definition, Abfrage und Manipulation der Daten erfolgt mit einer Datenbanksprache. Hierbei kommt der *Structured Query Language (SQL)* aufgrund der hohen Verbreitung eine große Bedeutung zu. Für räumliche Daten existiert *SQL/MM Spatial*, das Bestandteil des *SQL Multimedia and Application Packages*, einer Sammlung von SQL-Erweiterungen, ist. SQL/MM legt fest wie zweidimensionale,

räumliche Daten mittels SQL gespeichert, abgefragt und verwaltet werden (KOTHURI, GODFRIND & BEINAT 2004).

Ein räumliches DBMS ist aber nicht zwangsweise für die datenbankbasierte Speicherung notwendig. Bei kleinen GI-Systemen bzw. kleinen Internet-GIS ist ein räumliches DBMS meist zu komplex und auch überdimensioniert. In solchen Fällen kann beispielsweise auf ein relationales Datenbank-Managementsystem (RDBMS) zurückgegriffen werden.

2.1.4.2 Dateibasierte Speicherung

KORDUAN & ZEHNER (2008) nennen wichtige dateibasierte Datenformate, die vorwiegend zur Speicherung von Geodaten in einem Internet-GIS verwendet werden. Hierzu zählen sowohl reine Textformate wie zum Beispiel TXT und CSV (Comma Separated Values) als auch Rasterdatenformate wie TIFF, PNG, JPG etc. sowie Vektordatenformate wie beispielsweise SHP, DGN und DXF. Für die Beschreibung und den Austausch von Geodaten sind XML-basierte Formate wie die Geography Markup Language (GML) und die Keyhole Markup Language (KML) weit verbreitet (STEFANAKIS & PATROUMPAS 2008). XML-basierte Formate können ebenfalls zu den textbasierten Formaten gezählt werden (W3C 2009).

Ein weiteres Format für den Austausch von Daten bzw. Geodaten ist die JavaScript Object Notation (JSON). Hierbei handelt es sich um ein schlankes und textbasiertes Datenformat, das von Maschinen einfach zu parsen und zu generieren ist (JSON 2010). Die Struktur von JSON kann man sich beispielsweise als assoziatives Array vorstellen. Es handelt sich um eine Kombination eines Namens und einem zugehörigen Wert, was auch als Name/Wert-Paar bezeichnet wird. Für den Austausch von geographischen Informationen kann das JSON Derivat GeoJSON verwendet werden. Im folgenden Listing sind Daten sowohl im XML-Format als auch im JSON-Format dargestellt.

Listing 2-1: Gegenüberstellung XML-Format (links) und JSON-Format (rechts)

01 <project>	01 {
02 <scan>	02 project: {
03 <id>001</id>	03 scan: [
04 <name>Scan 01</name>	04 { id:1, name:'Scan 01' },
05 </scan>	05 { id:2, name:'Scan 02' }
06 <scan>	06]
07 <id>002</id>	07 }
08 <name>Scan 02</name>	08 }
09 </scan>	
10 </project>	

2.1.5 Tools zum Erstellen von Web-Mapping Applikationen

Gegenwärtig existieren zahlreiche client- und serverseitige Möglichkeiten, mit denen Web-Mapping Applikationen realisiert werden können. Auf proprietäre Tools wie ArcGIS Server, GeoMedia WebMap, MapInfo MapXtreme etc. wird an dieser Stelle nicht eingegangen, da diese Tools nicht frei verfügbar sind und nicht ohne weiteres implementiert werden können. Für freie Software bietet das FreeGIS-Projekt einen Überblick bezüglich Software im GIS-Sektor. Allerdings werden in der Rubrik Web-GIS ca. 70 Tools gelistet mit denen es möglich ist Web-Mapping Applikationen zu entwickeln (FOSSGIS 2009). Diese Tools zu analysieren und ein geeignetes herauszufiltern ist nur mit sehr hohem Aufwand realisierbar.

SCHÜTZE (2007) hat elf Tools, die im FreeGIS-Projekt gelistet sind, wie CartoWeb, Chameleon, Mapbender, OpenLayers, p-mapper etc. nach verschiedenen Kriterien verglichen und bewertet. Hierbei erfasste er neben allgemeinen Informationen wie Beschreibung, Lizenzierung, Architektur und unterstützte Browser auch Informationen zur Community und deren Aktivität. Der Fokus der Analyse wurde auf die Usability der Tools gesetzt, wo die einzelnen Komponenten wie Hauptkarte, Werkzeugleisten, Legende, Zooming und Panning etc. bewertet wurden. Weitere Kriterien geben Auskunft über vorhandene Dokumentationen, zusätzliche Features sowie Bemerkungen und Screenshots der untersuchten Beispiel-Anwendungen. Google Maps wurde aufgrund der großen Bedeutung und Beliebtheit als einziges proprietäres Tool in den Vergleich mit aufgenommen.

Die Auswertung der Analyse nimmt SCHÜTZE (2007) unter dem Gesichtspunkt der Usability vor. Er definiert in diesem Zusammenhang den Begriff *Smart Map Browsing* und versteht hierunter „eine für den Benutzer effektive, effiziente und zufriedenstellende Gebrauchstauglichkeit von WebMapping-Anwendungen“. In Bezug auf die zu entwickelnde Webapplikation, die sowohl Personen mit als auch ohne GIS-Background adressiert, wird dieses Kriterium als sehr bedeutend angesehen. Aufgrund seiner Vergleiche kommt SCHÜTZE (2007) zu dem Ergebnis, dass OpenLayers in fast allen untersuchten Kriterien überzeugt und eine der aktivsten Communities besitzt. Weiterhin sieht er hohes Potential hinsichtlich der zukünftigen Entwicklung für OpenLayers.

KORDUAN & ZEHNER (2008) verwenden einen ähnlichen Begriff wie SCHÜTZE (2007) namens *Smart Mapping Clients*, verstehen darunter ebenfalls JavaScript-Frameworks wie OpenLayers sowie Dienste wie Google Maps und schreiben diesen Clients einen großen Verbreitungsgrad zu. Bei der Entwicklung der Webapplikation

wird, basierend auf den Ergebnissen von SCHÜTZE (2007), auf OpenLayers und andere JavaScript-Frameworks die mit OpenLayers in Verbindung stehen, zurückgegriffen. Zuvor wird jedoch der Begriff JavaScript-Framework erläutert sowie JavaScript-Frameworks im Umfeld von OpenLayers und OpenLayers selbst vorgestellt.

2.1.6 JavaScript-Frameworks

Unter einem *Framework*¹ versteht man nach EICHINGER (2004) ein wiederverwendbares Softwaresystem in dem grundlegende Funktionalitäten implementiert sind und das durch Spezialisierung in eine Anwendung überführt werden kann. Es handelt sich hierbei also nicht um ein fertiges Programm sondern um einen Rahmen (Frame), mit dem Applikationen erstellt werden können. Ein Framework umfasst in der Regel eine Bibliothek auf die durch eine Programmierschnittstelle (Application Programming Interface, kurz *API*) zugegriffen werden kann (STEYER 2008). Auf diese Bibliotheken kann entsprechend mit der Skriptsprache JavaScript zugegriffen werden.

Häufig werden diese Frameworks auch als JavaScript-Toolkits, JavaScript-Bibliotheken oder auch als AJAX-Frameworks etc. bezeichnet, da sie häufig auf der AJAX-Technologie basieren (MINTERT & LEISEGANG 2007).² AJAX steht für *Asynchronous JavaScript and XML* und wird zur Kommunikation des Clients mit dem Server verwendet, wobei diese Kommunikation im Hintergrund stattfindet. Der Nutzer einer Webseite muss diese nicht neu laden wenn er mit dem Server kommuniziert, lediglich Teile dieser Webseite werden mittels AJAX nachgeladen. Der Datenaustausch mit dem Server muss nicht zwangsweise mit XML realisiert werden. Auch wenn der Name AJAX darauf hindeutet, kann beispielsweise auch JSON (siehe Kapitel 2.1.4.2) verwendet werden.

2.1.6.1 OpenLayers

OpenLayers ist ein JavaScript-Framework und ein Projekt der Open Source Geospatial Foundation mit dem Geodaten in einem Browser dargestellt werden können (OPENLAYERS 2010a). Hierzu steht eine API, die sogenannte *OpenLayers JavaScript Mapping Library* zur Verfügung. Mit dieser können Internet-GIS ähnlich wie mit der Google Maps-, Bing Maps- oder Yahoo Maps-API erstellt werden (siehe Abbildung

¹ Die Definition des Begriffs Framework variiert in der Literatur. Man findet Definitionen die auf die Struktur von Frameworks abzielen und andere bei denen der Zweck der Frameworks im Vordergrund steht (FAYAD, SCHMIDT & JOHNSON 1999).

² In der Literatur bestehen bezüglich der Begrifflichkeiten unterschiedliche Ansichten. STEYER (2008) trennt das Framework vom Toolkit und versteht unter dem Toolkit eine Sammlung von Programmen (Tools). Allerdings stellt er auch dar, dass die Termini nur schwer zu trennen sind. In dieser Masterthesis wird sowohl für das Framework als auch für das Toolkit einheitlich der Begriff JavaScript-Framework verwendet.

2-3). Allerdings bringt OpenLayers im Gegensatz zu Google Maps und Bing Maps keine eigenen Geodaten mit. OpenLayers ist in der Skriptsprache JavaScript entwickelt, nutzt Komponenten des *Prototype.js*- und des *Rico-Frameworks* und unterstützt Standards des Open Geospatial Consortiums wie beispielsweise den Web Map Service und den Web Feature Service. Weiterhin sind in OpenLayers Schnittstellen zum UMN MapServer, GML, GeoRSS, ka-Map, World Wind, Google Maps, Bing Maps etc. implementiert. OpenLayers läuft vollständig auf dem Client und kann somit unabhängig von Serversoftware eingesetzt werden.

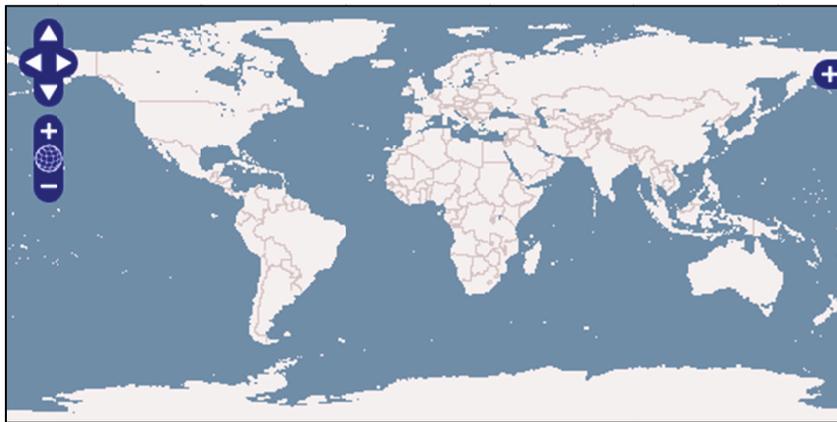


Abbildung 2-3: User Interfaces OpenLayers (Quelle: <http://www.openlayers.com>)

OpenLayers stellt typische Komponenten für die grafische Benutzeroberfläche (Graphical User Interface) von Webapplikation wie beispielsweise Übersichtskarte, Maßstab, Layer-Übersicht, Werkzeugleiste und Zoomnavigationsleiste zur Verfügung. Neben der reinen Darstellung von Geodaten können beispielsweise auch Marker und Popups dargestellt werden. OpenLayers wurde unter einer BSD-Lizenz veröffentlicht (OPENLAYERS 2010a).

2.1.6.2 Ext JS

Ext JS ist ein JavaScript-Framework, mit dem Rich Internet Applications³ entwickelt werden können. Es wurde in 2006 als YUI-Ext, eine Erweiterung der Yahoo User Interface Library, entwickelt und später unabhängig von dieser weiter entwickelt (ZAMMETTI 2009). Seit Juni 2010 ist Ext JS ein Produkt des Unternehmens Sencha. Ext JS steht sowohl als Open Source Lizenz als auch als kommerzielle Lizenz zur

³ Rich Internet Applications (RIAs) sind Webapplikationen die eine sehr intuitive, ansprechende und effektive Benutzerschnittstelle mit sich bringen; schnelle Antwortzeiten bei Serveranfragen ohne neuen Seitenaufbau aufweisen und häufig einen hohen („rich“) Funktionsumfang mit sich bringen. RIAs können in der Regel schnell und weltweit bereitgestellt werden, sind unabhängig vom verwendeten Webbrowser und ermöglichen progressiven Download (DUHL 2003).

Verfügung (EXT JS 2010). Dieses JavaScript-Framework bietet neben zahlreichen Layout-Möglichkeiten zur Gestaltung der Benutzerschnittstelle (User Interface) auch zahlreiche weitere Komponenten wie beispielsweise Formular-Elemente (Datumfelder, Auswahllisten etc.), Menüleisten, Diagramme basierend auf Flash und Menübäume für die Entwicklung von Webapplikation. In Abbildung 2-4 ist ein typisches Ext JS Interface durch ein Grid-Panel (Daten in tabellarischer Form) dargestellt. Ext JS ist ein sehr umfangreiches JavaScript-Framework mit dem auch Desktop-ähnliche Browseranwendungen erstellt werden können.



Company	Price	Change	% Change	Last Updated
3m Co	\$71.72	0.02	0.03	09/01/2010
AT&T Inc.	\$31.61	-0.48	-1.54	09/01/2010
Alcoa Inc	\$29.01	0.42	1.47	09/01/2010
Altria Group Inc	\$83.81	0.28	0.34	09/01/2010

Abbildung 2-4: User Interfaces ExtJS (Quelle: <http://www.sencha.com/products/js/>)

2.1.6.3 GeoExt

Mit der OpenLayers API können sehr umfassende Web-Mapping Applikationen entwickelt werden. Allerdings weist OpenLayers bei der grafischen Benutzeroberfläche einige Mankos auf. An dieser Stelle knüpft das JavaScript-Framework GeoExt an, dass die beiden JavaScript-Frameworks OpenLayers und Ext JS zusammen bringt (GEOEXT 2010). GeoExt ist ein noch relativ junges JavaScript-Framework, das am 15. Juni 2009 in der Version 0.5 veröffentlicht wurde (GEOEXT 2009).



Abbildung 2-5: User Interfaces GeoExt (Quelle: <http://www.geoext.org>)

Im Gegensatz zu OpenLayers hat Ext JS eine sehr beachtliche Benutzeroberfläche für die Entwicklung von Rich Internet Applications, bietet aber keine Möglichkeit Karten im Stil von OpenLayers einzubinden (HOCEVAR & MOULLET 2009). GeoExt vereint somit die Vorteile der beiden JavaScript-Frameworks OpenLayers und Ext JS.

2.2 Terrestrisches Laserscanning

Dieser Abschnitt gibt einen Einblick in die Thematik des terrestrischen Laserscannings in Bezug auf die Visualisierung von Laserscanning-Daten innerhalb des World Wide Web. Neben der Erläuterung des Messverfahrens und der Klassifikation von Laserscanning-Software wird gezeigt wie Laserscanning-Daten mithilfe eines Webbrowsers (Browser) wie beispielsweise dem Internet Explorer visualisiert werden können.

2.2.1 Begriffsbestimmung und Funktionsweise

Terrestrisches Laserscanning ist ein Messverfahren zur berührungslosen Erfassung von Oberflächen und Objekten mit Hilfe eines Laserscanners (INGENSAND & SCHULZ 2005; KRAUS 2007). Ausgehend vom Standpunkt des Laserscanners wird mit Hilfe eines stark gebündelten Laserstrahls die Oberfläche von Objekten durch eine Vielzahl von Messpunkten diskretisiert (siehe Abbildung 2-6). Die Erfassung erfolgt systematisch nach dem Phasenvergleichsverfahren oder dem Pulslaufzeitverfahren. Die Objekte bzw. Objektoberflächen werden dabei spalten- oder zeilenförmig abgetastet, wobei die Erfassung nicht auf markante Punkte abzielt (KERN 2007; STAIGER 2005; WÖLFELSCHNEIDER 2009).

Beim Phasenvergleichsverfahren wird die Entfernung vom Laserscanner zum Messobjekt durch Phasenvergleich des ausgesandten und reflektierten Signals bestimmt. Mit diesem Verfahren können sehr hohe Messfrequenzen von bis zu 1,2 MHz erzielt werden. Jedoch ist die Reichweite gegenwärtig auf ca. 80 m begrenzt. Beim Pulslaufzeitverfahren wird die Entfernung aus der Laufzeit sehr kurzer Intensitätsimpulse zwischen Laserscanner und Messobjekt abgeleitet. Mit diesem Messverfahren können Messfrequenzen von bis zu 125 kHz erzielt werden. Eine Entfernungsmessung kann über mehrere Kilometer erfolgen (PFEIFER et al. 2008; STAIGER 2005). Die von einem Standpunkt des Laserscanners erfassten Punkte werden im Allgemeinen als Scan (KRAUS 2007) oder als Punktwolke (INGENSAND & SCHULZ 2005) bzw. als 3D-

Punktwolke (STAIGER & WUNDERLICH 2006) bezeichnet. Im Rahmen dieser Masterthesis wird hierfür der Begriff Scan verwendet.

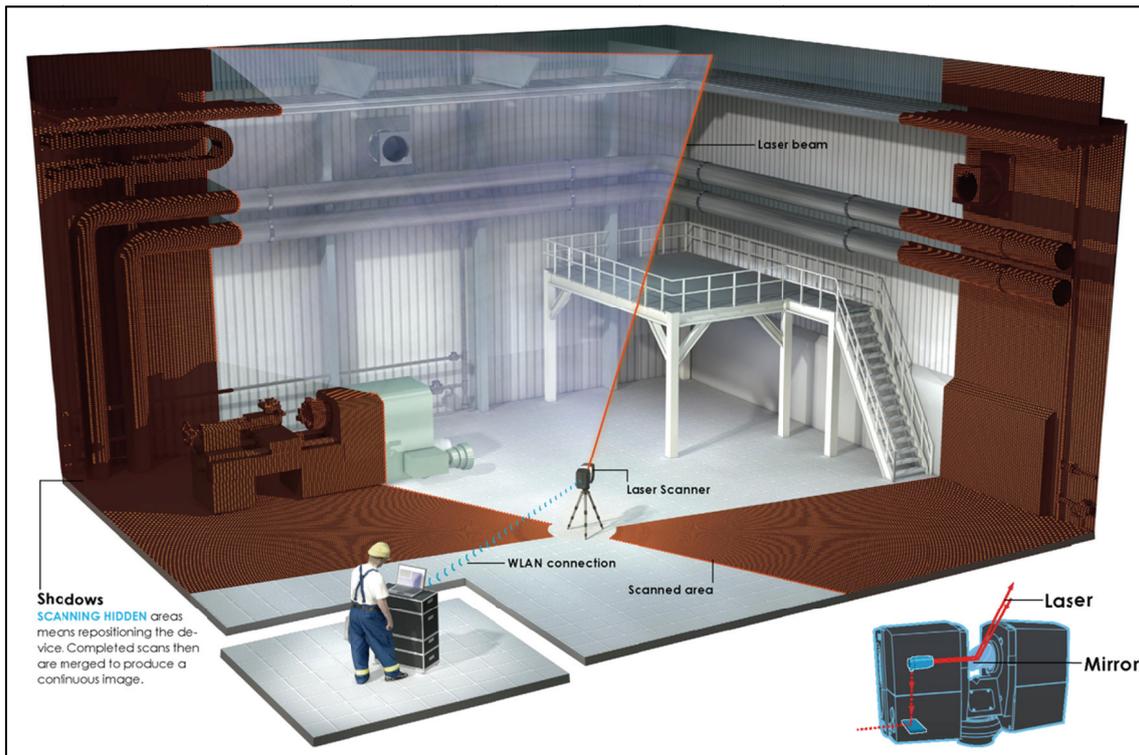


Abbildung 2-6: Funktionsprinzip des Laserscanning (Quelle: FARO 2010)

Durch Laserscanner werden im Allgemeinen vier Messwerte erfasst: ein Horizontalwinkel, ein Vertikalwinkel, die Entfernung zum Messobjekt und ein Intensitätswert, der auch als Remissionswert bezeichnet wird und die Reflektivität des Objektes beschreibt. Aus den ersten drei aufgeführten, polaren Messwerten werden kartesische Koordinaten bestimmt, die sich auf ein internes Laserscanner-Koordinatensystem beziehen (STAIGER 2005). Sofern der Laserscanner über eine interne oder externe Kamera verfügt, kann jedem Punkt auch ein RGB-Wert (Rot, Grün, Blau; additives Farbschema) zugewiesen werden. Somit kann ein einzelner Messpunkt über sieben Werte verfügen.

2.2.2 Visualisierung von Laserscanning-Daten

Laserscanning-Daten können verschiedenartig verarbeitet werden. Nach SCHÄFER (2008) ist die Verarbeitung der Punktwolken abhängig von der Aufgabenstellung, die letztlich zu unterschiedlichen Endprodukten führt. Die Verarbeitungsmöglichkeiten von Punktwolken können in Visualisierung, numerische Analyse, Digitalisierung und Modellierung sowie Veredelung der Modellierung differenziert werden (SCHÄFER 2008; STAIGER & WUNDERLICH 2006). STAIGER (2005) verfolgt einen anderen

Ansatz: er bezieht sich auf Software und unterscheidet bei der Verarbeitungsmöglichkeit von Punktwolken zwischen Plug-Ins für CAD-Programme und Software zur Auswertung und Visualisierung. Er subsummiert dabei unter Auswertung auch die Veredelung der Punktwolken.

Innerhalb dieser Masterthesis steht die Visualisierung der Laserscanning-Daten im Vordergrund, weshalb ausschließlich diese näher betrachtet wird. Der Begriff Visualisierung im Kontext des Laserscannings wird in der Literatur unterschiedlich dargestellt. STAIGER (2005) beschreibt die Visualisierung innerhalb des Ablaufs eines Laserscanning-Projekts und versteht unter Visualisierung die Darstellung von Punktwolken, von Ergebnissen der Auswertung und die hybride Darstellung beider. Beispiele hierfür sind Punktwolken deren Farbwerte nach der Höhe oder der Entfernung vom Laserscanner codiert sind, Punktwolken mit geometrischen Elementen oder Digitalbildern, 3D-Modelle, animierte Punktwolken und Höhenlinienpläne.

SCHÄFER (2008) unterteilt die Visualisierung von Punktwolken in drei Gruppen und betrachtet sie im Zusammenhang mit den Verarbeitungsschritten der Punktwolken. Er bezieht sich bei der Visualisierung ausschließlich auf die Ergebnisse der Datenerfassung eines Laserscanners, d. h. auf die Punktwolke und die Fotos einer internen oder externen Digitalkamera. Im Rahmen dieser Masterthesis wird unter Visualisierung das Betrachten von Punktwolken mit einer Viewer-Software im Sinne von SCHÄFER (2008) verstanden. Viewer ermöglichen grundsätzlich das Rotieren, Verschieben und Zoomen von Punktwolken (BOEHLER et al. 2005; SCHÄFER 2008); darüber hinaus bieten sie aber auch zusätzliche Funktionalität wie zum Beispiel Messen, Bemaßen und Markieren. In den folgenden Abschnitten wird die Klassifizierung nach SCHÄFER (2008) ausführlicher betrachtet.

2.2.2.1 Punktwolken

Die erste Kategorie, die SCHÄFER (2008) aufführt, bezieht sich auf die Visualisierung der reinen Punktwolke. Man kann sich in der Punktwolke frei bewegen und bei hinreichender Punktdichte wirken die Punktwolken sehr flächenhalt und die gescannten Bereiche sehr realitätsnah. Sofern man in die Punktwolke hineinzoomt, werden die einzelnen Punkte ersichtlich und die flächenhafte Darstellung geht verloren, da nur noch ein Punktraster erkennbar ist (siehe Abbildung 2-7 und Abbildung 2-8).

Sofern während der Datenerfassung neben den Koordinaten auch Farbinformationen mit einer externen oder internen Kamera erfasst werden, können den Punkten Farbinformationen als RGB-Werte zugewiesen werden. Die Punktwolken können auch auf Grund-

lage des Intensitätswerts, der von den meisten Laserscannern mit erfasst wird, durch einen Grauwert eingefärbt werden. Weiterhin können den Punkten, aufgrund ihrer Entfernung vom Laserscanner oder aufgrund ihrer Höhe, tiefen- oder höhenrelevante Farbinformationen zugewiesen werden. Beispiele für Viewer, die diese Art der Visualisierung ermöglichen sind der Cyclone Viewer von Leica Geosystems, der LFM Viewer von Z+F, Trimble RealWorks Viewer und SmartPlant Review von Intergraph mit zusätzlichem Plug-In.

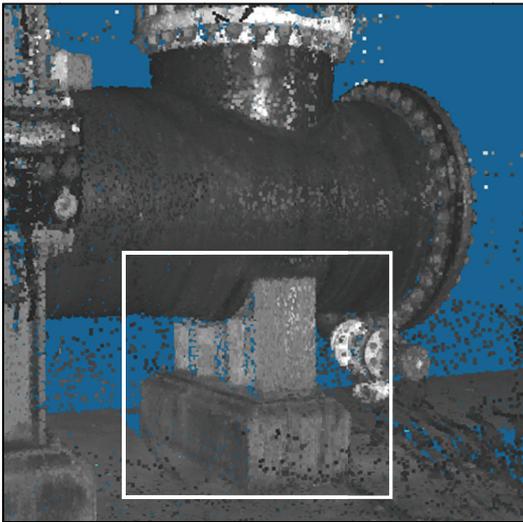


Abbildung 2-7: Flächenhafte Darstellung der Punktwolke, die in Abbildung 2-8 vergrößert ist (Rahmen)

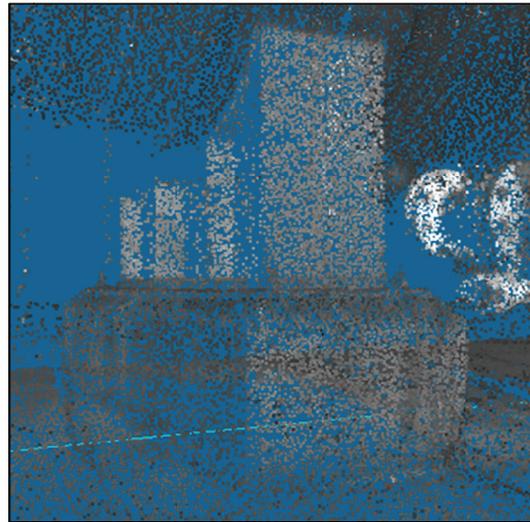


Abbildung 2-8: Verlust der flächenhaften Darstellung durch Vergrößerung des Bereiches aus Abbildung 2-7

2.2.2.2 Orthofotos

Als zweite Visualisierungsmöglichkeit nennt SCHÄFER (2008) Orthofotos mit zusätzlicher Tiefeninformation, die aus den Punktwolken erzeugt werden. Hierzu wird die Punktwolke trianguliert und anschließend als digitales Oberflächenmodell zur differentiellen Entzerrung der Digitalbilder verwendet. Die Digitalbilder werden von einer externen Kamera, die auf dem Laserscanner installiert ist, erzeugt. Die Software RiSCAN Pro Viewer des österreichischen Unternehmens Riegli bietet diese Art der Visualisierung an.

2.2.2.3 Panoramabilder

Eine weitere Möglichkeit der Visualisierung nach SCHÄFER (2008) besteht darin, eine Serie von Digitalbildern zu einem Panoramabild zusammensetzen und dieses mit der Punktwolke zu georeferenzieren. Es ist nicht möglich sich frei in der Punktwolke zu bewegen; der Betrachter befindet sich vielmehr im Aufnahmestandpunkt des Laserscanners. Das Panoramabild ist um 3D-Information bereichert, was die Daten-

menge der originären Punktwolke reduziert und eine webbasierte Betrachtung der Laserscanning-Daten ermöglicht. Hierfür werden Plug-Ins benötigt, die kostenlos zur Verfügung gestellt werden. Beispiele für diese Art von Viewern sind LFM NetView von Z+F und TruView von Leica Geosystems.

Panoramabilder stellen gegenwärtig die einzige Möglichkeit dar, Laserscanning-Daten im World Wide Web zu betrachten.⁴ Die Panoramabilder werden somit in der Webapplikation verwendet um Laserscanning-Daten zu visualisieren. In den folgenden Abschnitten wird bestehende Viewer-Software näher erläutert.

2.2.3 Viewer-Software

Dieses Kapitel gibt einen Überblick über bereits existierende Software zum Betrachten von Punktwolken über das World Wide Web und stellt Software vor, die mit diesen Tools in Zusammenhang steht. Wie bereits unter Punkt 2.2.2.3 erwähnt, existieren zwei proprietäre und einander sehr ähnliche Softwareprodukte namens TruView und LFM NetView. Diese Produkte ermöglichen es, Punktwolken über das WWW zu betrachten und mit ihnen zu arbeiten. Beide Softwareprodukte stellen Laserscanning-Daten in einem unverzerrten Panorama aus dem Blickwinkel des Laserscanners dar. Diese Panoramansicht kann sowohl als Graustufenbild als auch als Echtfarbbild vorliegen.

2.2.3.1 TruView

TruView ist eine lizenzfreie Software mit der Scans betrachtet werden können. Es handelt sich bei dieser Software um einen Viewer, der es ermöglicht Punktwolken im Microsoft Internet Explorer⁵ zu betrachten. Im Internet Explorer wird ein fotorealistisches Panoramabild der Laserscanning-Daten ausgehend vom Aufnahmestandpunkt des Laserscanners dargestellt (LEICA 2009b). TruView bezieht sich immer auf die Laserscanner-Standpunkte eines Projekts. Zusätzlich kann innerhalb eines TruViews gemessen werden und Beschriftungen können ebenfalls angebracht werden. Aktuell existiert die Version 2.

⁴ Es gibt weitere Möglichkeiten Laserscanning-Daten im Internet darzustellen. Beispielsweise können in Bing Maps Punktwolken angezeigt werden. Allerdings handelt es sich zum gegenwärtigen Zeitpunkt mehr um Staffagen, weshalb auf diese Art der Visualisierung nicht eingegangen wird.

⁵ Da TruView ein Plug-In für den Microsoft Internet Explorer ist, können die Panoramabilder nicht ohne weiteres in anderen Browsern verwendet werden. Hierfür sind zusätzliche Plug-Ins notwendig. Beispielsweise kann Mozilla Firefox mit dem Add-On *IE Tab* Webseiten mit der Internet Explorer Engine aufrufen und somit auch TruViews darstellen.

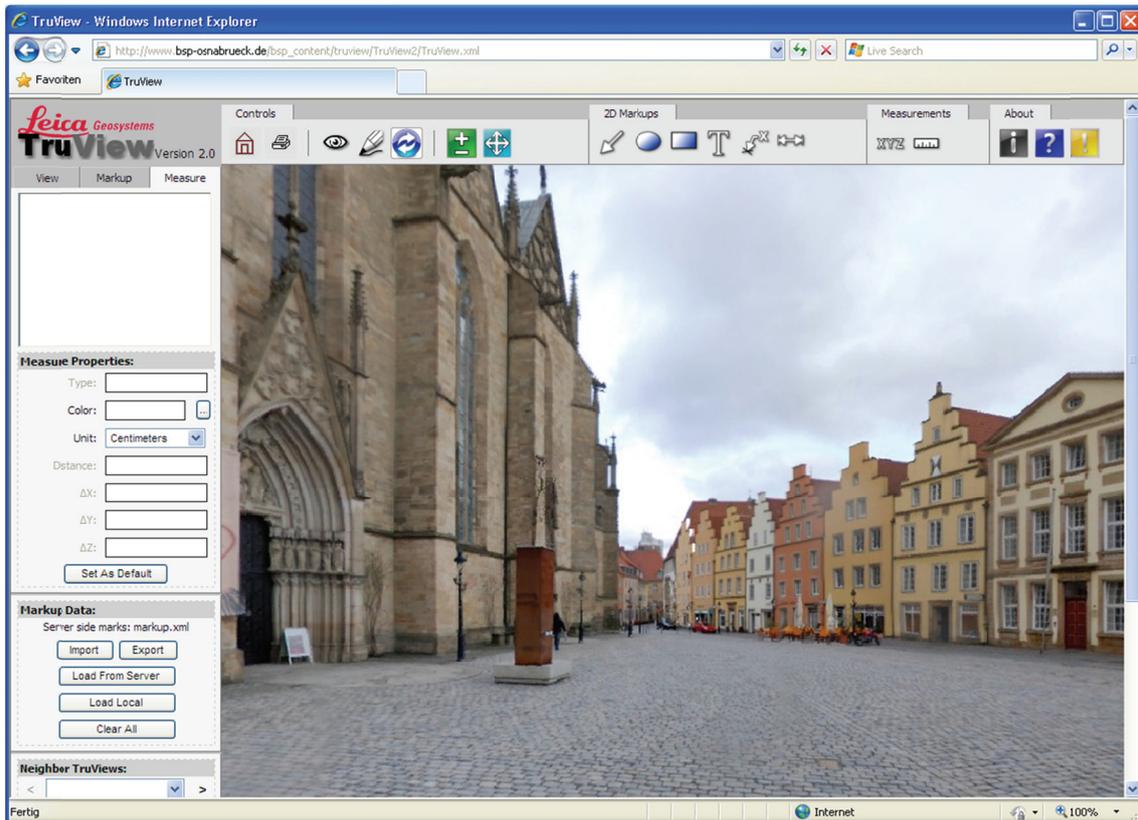


Abbildung 2-9: Farbiges TruView in der Version 2.0 (Quelle: <http://www.bsp-osnabrueck.de>)

Die TruView-Software wird von Leica Geosystems entwickelt und kostenfrei für alle Nutzer zur Verfügung gestellt. Um die TruViews zu erzeugen, muss die lizenzpflichtige Software Cyclone Publisher verwendet werden. Der Cyclone Publisher greift auf die Laserscanning-Daten zu, die in einer Cyclone-Datenbank gespeichert sind und erzeugt anschließend die TruViews. Bestandteile der TruViews sind auch Metadaten eines jeden Laserscanner-Standpunkts und ein Übersichtsplan, der in eine Webseite als Bild integriert ist. Die Metadaten beinhalten im Wesentlichen Informationen zum Scan, zur Panoramansicht, zum Publisher und zum Eigentümer der Daten. Der Übersichtsplan, auch als *Site Map* bezeichnet, ist eine verweissensitive Grafik, die wie ein Screenshot aus den Laserscanning-Daten der Cyclone Datenbank erzeugt wird. Der Screenshot kann aus einem beliebigen Blickwinkel (Draufsicht, Isometrische Ansicht, Seitenansicht etc.) generiert werden. Die Laserscanner-Standpunkte werden durch kleine gelbe Dreiecke in der Übersichtskarte symbolisiert. Diese Dreiecke sind mit dem jeweiligen TruView verlinkt. Dieser Hyperlink führt zu einer XML-Datei (Truview.xml), die das TruView Plug-In und das entsprechende Panorama im Internet Explorer lädt. Zwei Darstellungen einer solchen Site Map befinden sich im ersten Kapitel (siehe Abbildung 1-1 und Abbildung 1-2).

Auf die Panoramabilder kann über ein Netzwerk wie beispielsweise ein Intranet oder auch das Internet zugegriffen werden. Damit kann also jeder Nutzer auch ohne fundierte Kenntnisse in den Bereichen Laserscanning, CAD oder 3D auf die Punktwolken zugreifen und mit den TruViews arbeiten. Ein solches TruView ist in Abbildung 2-9 dargestellt.

Mit dem TruView Plug-In lassen sich verschiedene Funktionen wie beispielsweise Markieren, Messen und Beschriften ausführen. Die Funktionen sind in Tabelle 2-1 aufgeführt.

Tabelle 2-1: TruView Funktionen (Quelle: LEICA 2009b)

Funktion	Beschreibung
Betrachten	Von jedem Scanner-Standpunkt aus, 360° horizontale und 360° vertikale Drehung
Markieren/Beschriften	Pfeil, Rechteck, Kreis und Ellipse mit Hinweislinien und Text Angabe des Nutzers mit Stempel (Zeit/Datum) Eigenschaften ändern: Farbe, Schrift, Transparenz, Liniendicke, Maßeinheit
Messen	Distanz zwischen zwei Punkten, Koordinaten
Hyperlinks	Alle Grafiken/Beschriftungen, Hinweislinien und Texte können verlinkt werden
Gespeicherte Ansichten	Automatisch durch Erstellen, jederzeit abrufbar
Zusammenarbeit/ Datenaustausch	Gespeicherte Ansichten und Ergänzungen sowie Standardeinstellungen können importiert und exportiert werden

2.2.3.2 PlantView

PlantView ist eine webbasierte Navigationsoberfläche, die zur Darstellung und Strukturierung einer gescannten Anlage im Microsoft Internet Explorer dient (UNISON ENGINEERING 2009). PlantView wird vom Berliner Ingenieurunternehmen Unison Engineering entwickelt und stellt einen Übersichtsplan dar, der es ermöglicht auf die TruViews von Leica Geosystems zuzugreifen.

Unison versucht mit dem selbst entwickelten Übersichtsplan PlantView die Site Map (siehe Kapitel 2.2.3.1), die mit dem Cyclone Publisher erzeugt wird, übersichtlicher zu gestalten. PlantView verwendet, wie auch die Site Map von TruView, eine verweissensitive Grafik in der alle Laserscanner-Standpunkte durch blaue Kreise dargestellt sind. Die Grafik stellt in der Regel eine Draufsicht auf das gescannte Areal dar, die durch eine topographische Karte, einen horizontalen Schnitt durch die Laserscanning-

Daten oder Ähnliches charakterisiert ist. Der Zugriff auf die TruViews kann auf verschiedenen Wegen erfolgen. So ist, wie auch bei TruView, ein Zugriff über die verweissensitive Karte (blau markierte Punkte) möglich. Weiterhin bietet PlantView den Zugriff über ein Menü, das in Abbildung 2-10 auf der linken Seite ersichtlich ist. Ein zusätzliches Menü befindet sich in der oberen Zeile derselben Abbildung. Hier kann man die TruViews anhand von kleinen Vorschaubildern, die der Laserscanner erzeugt, auswählen. Unison gruppiert hierbei die verschiedenen Laserscanner-Standpunkte nach der Höhe. Befinden sich Laserscanner-Standpunkte in etwa auf gleicher Höhe beispielsweise im Erdgeschoss eines Gebäudes, werden diese zu einer Ebene zusammengefasst, das Obergeschoss wird zu einer neuen Ebene gruppiert.

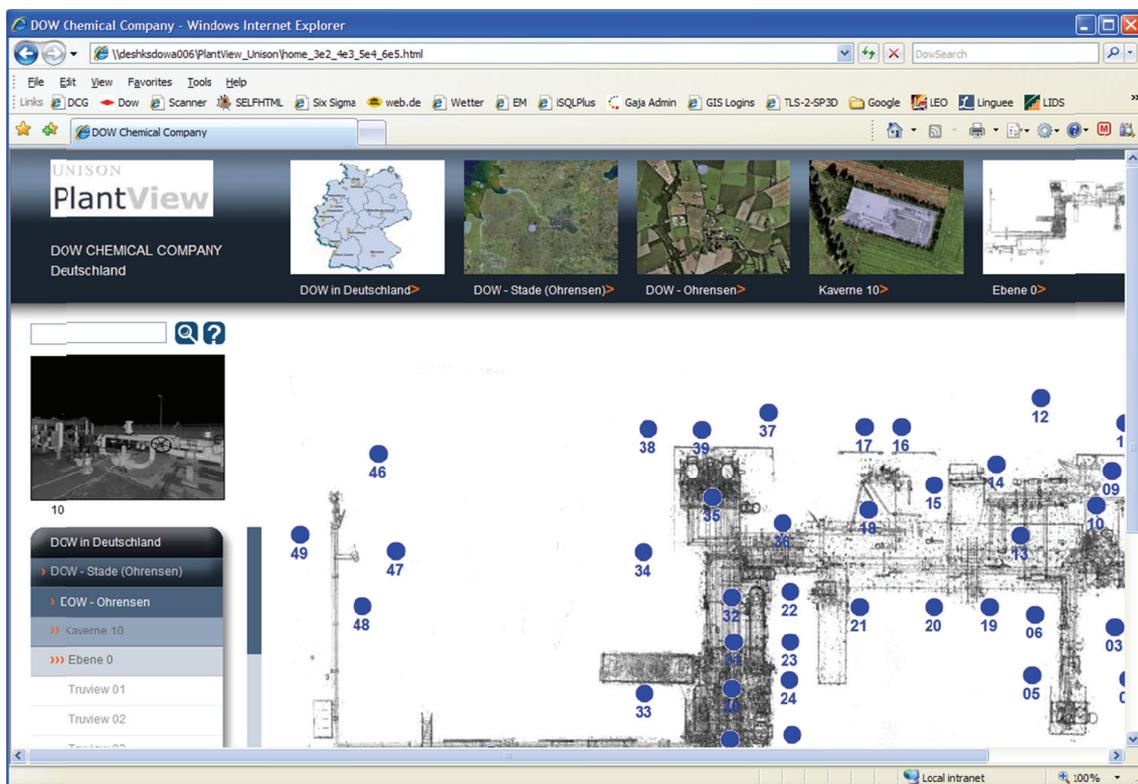


Abbildung 2-10: Navigationsoberfläche PlantView (Quelle: DOW 2009b)

Somit hat PlantView im Vergleich zu TruView zwar eine übersichtlichere und auch anwenderfreundlichere Lösung in petto, allerdings handelt es sich bei der Übersichtskarte immer noch um eine verweissensitive Karte, mit der nur kleinere Laserscanning-Bereiche dargestellt werden können. Auch die Pflege des Systems und das Entwickeln neuer PlantView-Projekte werden als sehr zeitaufwendig eingeschätzt.

2.2.3.3 LFM NetView

LFM NetView ist eine kostenpflichtige Software mit der auf Punktwolken über ein Netzwerk wie beispielsweise das Internet zugegriffen werden kann (Z+F 2010a). Sie wird vom englischen Unternehmen Z+F entwickelt und kann als eigenständige Software oder als Plug-In für einen Webbrowser verwendet werden. Neben dem Betrachten, Messen und Beschriften von Punktwolken sind in LFM NetView Funktionen wie das Umschalten zwischen verschiedenen Ansichten der Übersichtskarte, das Bestimmen von Rohrdurchmessern und Master-Slave Sitzungen beispielsweise für Präsentationen implementiert. Momentan existiert die Version 1.1. Die Daten für LFM NetView werden mit der eigenständigen Software LFM Server erzeugt. Dieser überführt die Laserscanning-Daten eines Projekts in eine Datenbank, durch die sehr schnell auf Laserscanning-Daten zugegriffen werden kann (Z+F 2010e).

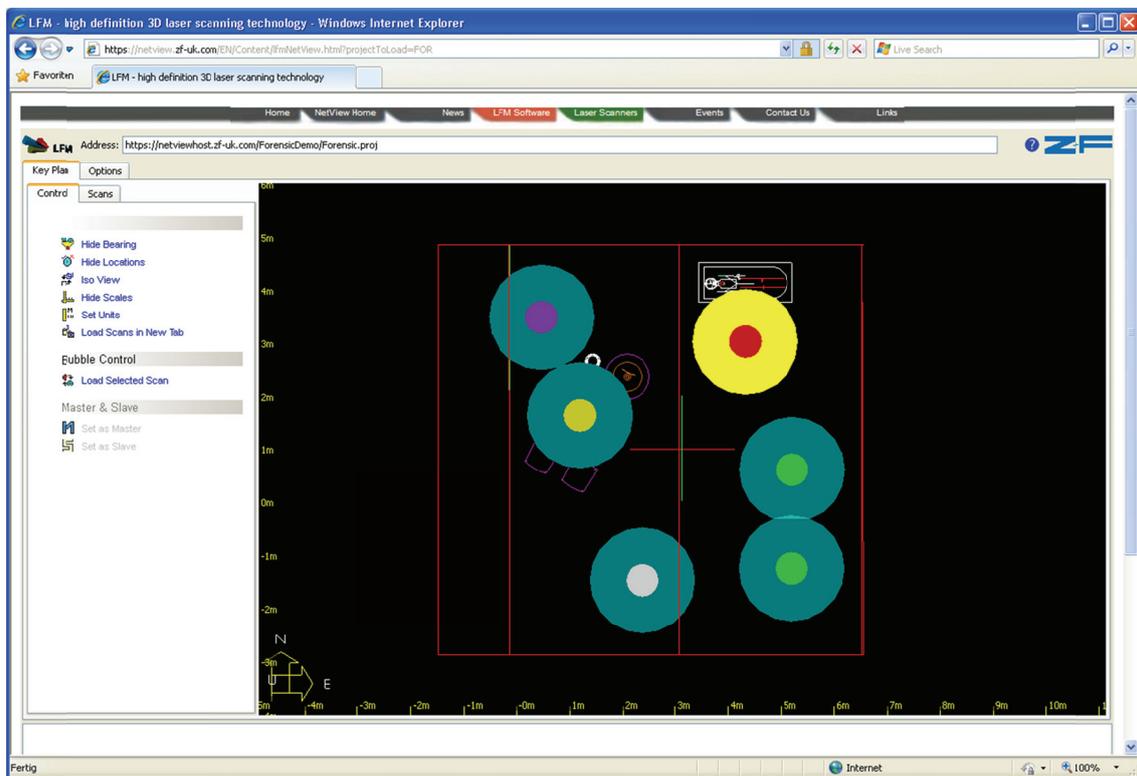


Abbildung 2-11: LFM NetView mit Übersichtsplan (Key Plan) (Quelle: Z+F UK 2010d)

Auch LFM NetView bietet einen Übersichtsplan (*Key Plan*) für den Zugriff auf die Panoramasichten (siehe Abbildung 2-11), die bei Z+F als *BubbleViews* bezeichnet werden. Der Übersichtsplan kann beispielsweise eine CAD-Zeichnung sein, die sowohl in der Draufsicht als auch in der isometrischen Ansicht dargestellt werden kann. Um auf die *BubbleViews* zugreifen zu können, muss zuerst der Übersichtsplan durch die Eingabe einer URL geladen werden. Der Zugriff selbst basiert auf dem HyperText

Transfer Protocol Secure (*HTTPS*), wodurch die Daten auch sicher im WWW betrachtet werden können. Die Usability von LFM NetView ist relativ gut, allerdings ist der Übersichtsplan sehr trivial und kartografisch stark verbesserungswürdig. Die Möglichkeit der Einbindung von Web Map Services wäre an dieser Stelle sehr konstruktiv, da dadurch die Orientierung des Nutzers erleichtert werden kann. Mit LFM NetView kann in einem Projekt eine unbegrenzte Anzahl von BubbleViews vorgehalten werden (Z+F 2010c). Ein direkter Zugriff auf ein einzelnes BubbleView ist bislang nicht möglich, aus Sichtweise des Autors jedoch empfehlenswert um eine Einbindung der Punktwolken in ein GI-System zu ermöglichen. Nach Gesprächen mit Z+F soll dieses Feature mit dem nächsten Release von LFM NetView bereitgestellt werden (COPPLE 2010). Bis dato kann ein *BubbleView* nur durch den *Key Plan* geladen werden. LFM NetView bietet die in der nachstehenden Tabelle aufgeführten Funktionen.

Tabelle 2-2: LFM NetView Funktionen (Quelle: Z+F 2010b)

Funktion	Beschreibung
Betrachten	Von jedem Scanner-Standpunkt aus, 360° horizontale und 360° vertikale Drehung
Markieren/Beschriften	Punkt, Rechteck und Polygon mit Hinweislinien und Text, Eigenschaften ändern: Position, Größe, Farbe, Schrift, Maßeinheit
Messen	Distanz zwischen zwei Punkten, Koordinaten, orthogonale Distanz zwischen Punkt und Ebene, Rohrdurchmesser
Zusammenarbeit/ Datenaustausch	Master-Slave Modus (erlaubt verschiedenen Personen das Betrachten der NetView-Daten, geführt durch einen „Master“)

2.3 Metadaten

Für die Organisation und die Verwaltung der Laserscanning-Projekte und -Daten werden Metadaten benötigt. Metadaten beschreiben die Geodaten und liefern Informationen mit denen beispielsweise die Zweckmäßigkeit der Daten für ein Projekt besser eingeschätzt werden kann. Zu Beginn dieses Abschnitts wird der Begriff Metadaten kurz erläutert und anschließend werden die für die Masterthesis bedeutenden Standards und deren Anwendung für die Umsetzung innerhalb der Webapplikation betrachtet. Hierbei wird ausschließlich auf die Kernelemente dieser Standards eingegangen.

2.3.1 Begriffsbestimmung

Der Begriff Metadaten wurde erstmals 1968 in informationswissenschaftlicher Literatur in unterschiedlichen Schreibweisen wie „meta data“ oder „meta-data“ verwendet (AALDERS 2007). Metadaten werden häufig als Daten über Daten beschrieben (BARTELME 2005; LONGLEY et al. 2005; STROBL 1994). Allerdings wird der Begriff Metadaten in der Literatur nicht einheitlich definiert und vielmehr kontextabhängig verwendet (KORDUAN & ZEHNER 2008; NISO 2004). In dieser Arbeit werden Metadaten im Kontext von Geoinformationen verstanden, die deren Eigenschaften und Inhalt beschreiben. Daten und Metadaten können nicht exakt voneinander getrennt werden, da eine Abhängigkeit vom Benutzerverständnis und vom Kontext vorhanden ist (KORDUAN & ZEHNER 2008). Die in dieser Arbeit verwendeten Metadaten, können somit in anderen Zusammenhängen nicht als Metadaten sondern als Daten bezeichnet werden.

Neben der relativen kurzen Definition von Metadaten liefert BARTELME (2005) eine sehr anschauliche Erläuterung des Begriffs. Geodaten stellen hiernach eine Abstraktion der realen Welt dar und müssen somit erfasst werden. Die Erfassung von Geodaten ist abhängig von verschiedenen Kriterien wie der Person, die die Daten erfasst, der Aufgabenstellung, Umwelteinflüssen etc. Bei der Datenerfassung werden verschiedene Methoden angewendet, die letztlich zu einem gewünschten Ergebnis führen. Die erfassten Daten sind somit nicht für alle Aufgaben verwertbar, sondern vielmehr zweckgebunden. Beispielsweise können Daten, die mit einem hohen Generalisierungsgrad erfasst wurden, nicht in Bereichen verwendet werden, wo eine geringe Erfassungsgeneralisierung notwendig ist. Um eine Zweckentfremdung der Daten zu verhindern, die unter Umständen zu falschen Resultaten führt, müssen die Annahmen und Einschränkungen während der Datenerfassung dokumentiert werden. Der Inhalt dieser Dokumentation wird als Metadaten bezeichnet. Am Beispiel des Laserscannings können das der Zeitpunkt der Messung; die klimatischen Verhältnisse während der Messung; die Person, die Daten erfasst und der Zweck des Laserscannings sein.

2.3.2 Standardisierung von Metadaten

2.3.2.1 Allgemeines

Metadaten-Standards beschreiben wie und mit welchen Elementen, Daten beschrieben werden sollen. Dabei werden obligatorische (verbindliche) und optionale Metadaten festgesetzt sowie Bedingungen festgelegt, unter welchen diese Elemente als verbindlich

anzusehen sind (AALDERS 2007). AALDERS (2007) gibt einen minimalen Metadatensatz für Geoinformationen an und bezieht sich dabei sowohl auf Studien die bei der Entwicklung des *ISO-Standards 19115* verwendet wurden, als auch auf den *Dublin Core Metadaten-Standard*. Dieser minimale Metadatensatz sollte folgendes beinhalten:

- Identifikation des Datensatzes – eindeutiger Name,
- Datenanbieter – Wer stellt Daten zur Verfügung?,
- Datenproduzent – Wer hat Daten erfasst?,
- Bezugssystem – semantische Definition mittels Thesaurus,
- Datenausdehnung – geografisch, semantisch und zeitlich,
- Geltungsdauer der Daten,
- Sprache der Metadaten,
- Syntax des Datentransfers zur Identifikation der Software zum Lesen der Daten
- Qualität der Daten – räumlich, semantisch, zeitlich und
- Rechte und Management – Kopierrechte, Nutzungsbedingungen etc.

Im Bereich der Geoinformatik existieren zahlreiche Metadaten-Standards wie beispielsweise der *Dublin Core Metadaten-Standard*, *ISO-Standard 19115*, *ISO-Standard 19119*, *ISO-Standard 19139*, der *Content Standard for Digital Geospatial Metadata*, das *Resource Description Framework (RDF)*⁶ etc. Vor allem der *ISO-Standard 19115* (Geographic information – Metadata) ist dabei von wesentlicher Bedeutung. Aber auch der Umweltdatenkatalog, der in vielen Bundesämtern und der Mehrheit der Bundesländer angewendet wird, kann als Standard für Metadaten zur Beschreibung von Umweltdaten betrachtet werden (KORDUAN & ZEHNER 2008). Für diese Masterthesis bzw. die Entwicklung der Webapplikation sind Metadaten beispielsweise für die Beurteilung der Laserscanning-Daten sehr bedeutsam. Aus diesem Grund werden die zwei bedeutenden Standards *ISO-Standard 19115* (siehe Kapitel 2.3.2.2) und *Dublin Core Metadaten-Standard* (siehe Kapitel 2.3.2.3) näher erläutert.

⁶ Das Resource Description Framework ist eine Sammlung von Standards des World Wide Web Consortium und dient der Beschreibung von Ressourcen die durch eindeutige Bezeichner (URIs) identifiziert werden (W3C 2009). Nach KORDUAN & ZEHNER (2008) ist die Verwendung von RDF dann zweckmäßig, wenn der Nutzer Metadatenelemente selbst definieren oder vorhandene Strukturen verändern soll. Sofern Metadatenmodelle nur sehr selten verändert werden müssen, also annähernd statischer Natur sind, bedarf es nicht dem RDF.

2.3.2.2 ISO 19115

Der Standard *ISO 19115: Geographic Information - Metadata* wurde im Mai 2003 vom Technical Committee 211 der International Organisation for Standardization (ISO) veröffentlicht und definiert eine Struktur zur Beschreibung von digitalen Geoinformationen und Services. Mit diesem Standard werden Informationen zur Identifikation, Ausdehnung, Qualität, räumlichen und zeitlichen Schemas, Bezugssystemen und der Verteilung von Geoinformationen bestimmt (ISO/TC 211 2003). Der *ISO 19115 Standard* ist für alle Geoinformationen anwendbar und unterstützt Sammlungen von Datensätzen, einzelne Datensätze sowie Features und deren Attribute.

Es werden sowohl verbindliche Elemente festgesetzt, um möglichst viele Metadaten-Anwendungen anzusprechen (AALDERS 2007) als auch optionale Metadaten-Elemente festgesetzt um Geoinformationen umfangreich zu beschreiben. Weiterhin werden Metadaten-Elemente festgelegt, die nur unter bestimmten Voraussetzungen anzuwenden sind (bedingt-notwendige Elemente). Insgesamt umfasst der Standard mehr als 400 Metadaten-Elemente, wovon 22 Metadaten-Elemente (Kernelemente; Core metadata) den minimalen Datensatz definieren. Von diesen Kernelementen sind sieben verbindlich, die anderen entweder optional oder bedingt notwendig. Die Kernelemente sind in Tabelle 2-3 aufgeführt und dort kurz beschrieben. Sofern das generische Modell dieses Standards nicht ausreichend ist, kann das Modell mithilfe eines Regelwerks standardgerecht erweitert werden (NONN & ZIPF 2007).

Innerhalb des Standards werden Metadaten mithilfe der Unified Modeling Language (UML) in *UML-Packages* dargestellt. Jedes *Package* kann eine oder mehrere Metadaten-Entitäten (UML: *Class*) besitzen. Metadaten-Entitäten wiederum beinhalten Metadaten-Elemente (UML: *Class attributes*), wobei eine Metadaten-Entität bestimmte Metadaten-Elemente mit gleichen Aspekten zusammenfasst. Das *Package Metadata entity set information* besteht aus der obligatorischen Entität *MD_Metadata*, die sowohl obligatorische als auch optionale Metadaten-Elemente enthält. Diese Entität stellt die oberste Einheit dar und ist ein Aggregat verschiedener Entitäten. Die einzelnen Packages bzw. Entitäten, deren Beziehungen und Attribute können an dieser Stelle nicht aufgeführt werden, da dies den Rahmen der Masterthesis sprengen würde. Es wird vielmehr auf die Dokumentation des Standards verwiesen.

Tabelle 2-3: Kernelemente des ISO 19115 Standards (Quelle: ISO/TC 211 2003; KST. GDI-DE 2008)

Typ	Element-Name	Beschreibung
Verbindliche Elemente	Dataset title	Bezeichnung unter der die Ressource bekannt ist
	Dataset reference date	Datum der Erzeugung, Veröffentlichung oder Überarbeitung der Ressource
	Dataset language	Im Datenbestand verwendete Sprache
	Dataset topic category	Thematische Einordnung des Datenbestands
	Abstract describing the dataset	Kurze, beschreibende Zusammenfassung des Inhalts der Ressource
	Metadata point of contact	Für die Metadaten verantwortliche Stelle
	Metadata date stamp	Datum, zu dem der Metadatensatz erzeugt bzw. geändert wurde
Optionale Elemente	Dataset responsible party	Kontaktinformation zur Person und Organisation, welche im Bezug zur Ressource steht
	Spatial resolution of the dataset	Angaben über die räumliche Auflösung der geografischen Informationen
	Distribution format	Bezeichnung und Version des Datenformats
	Additional extent information	Vertikale und zeitliche Komponente der Ausdehnung des betreffenden Objekts
	Spatial representation type	Methode, mit der geografische Informationen räumlich dargestellt werden
	Reference system	Information zum Referenzsystem
	Lineage	Information über die Datenerzeugung und die dafür genutzten Quellen oder Kenntnismangel der Herkunft
	On-line resource	Information zur Online-Ressource
	Metadata file identifier	Eindeutiger Identifikator für diesen Metadatensatz
	Metadata standard name	Bezeichnung des verwendeten Metadaten-Standards
Metadata standard version	Version des verwendeten Metadaten-Standards bzw. des genutzten Profils	
Bedingt notwendige Elemente	Geographic location	Geografisches Gebiet des Datenbestands
	Dataset character set	Exakte Bezeichnung des Zeichencode-Standards, der im Datenbestand verwendet wird
	Metadata language	Zur Dokumentation der Metadaten verwendete Sprache
	Metadata character set	Exakte Bezeichnung des Zeichencode-Standards, der im Metadatensatz verwendet wird

Da der Standard ISO 19115 vorrangig Vektordaten adressiert, wurde der ergänzende Standard ISO 19115-2 definiert; in diesem werden Metadaten zur Beschreibung von

Bildern und Rasterdaten festgesetzt (KRESSE & FADAIE 2004). Für Rasterdaten werden innerhalb der Webapplikation keine Metadaten verwendet, da primär auf Daten von Webservices wie Google Maps und Bing Maps gesetzt wird.

2.3.2.3 Dublin Core

Dublin Core (DC) stellt eine Sammlung von Konventionen zur Beschreibung von elektronischen Ressourcen dar, dessen Urheber die *Dublin Core Metadata Initiative* (DCMI) ist. DCMI ist eine offene Organisation die interoperable Metadaten-Standards entwickelt und dabei versucht einen möglichst breiten Einsatzbereich abzudecken und viele Geschäftsmodelle zu unterstützen (DCMI 2009a). Vorrangig wird dieser Standard im World Wide Web angewendet um Internetseiten zu beschreiben. Er wird von vielen Internet-Suchmaschinen unterstützt und führt bei Verwendung zu einer höheren Treffsicherheit bei der Suche im Internet (KORDUAN & ZEHNER 2008).

Unter Dublin Core wird häufig das *Dublin Core Metadata Element Set* verstanden, das aus 15 Metadaten-Elementen besteht (LONGLEY et al. 2005; SHEKHAR & XIONG 2008). Diese Sammlung ist aber nur ein Teil des gesamten Standards des DCMI, der als *DCMI Metadata Terms* (DCMI-TERMS) bezeichnet wird und weit mehr als die 15 Kernelemente beinhaltet. Im Rahmen dieser Masterthesis wird nur auf das *DC Metadata Element Set* der Version 1.1 vom 14. Januar 2008 näher eingegangen, da es für die Umsetzung der Webapplikation bedeutend ist und in dieser auch verwendet werden soll.

Tabelle 2-4: Dublin Core Metadata Element Set Version 1.1 (Quelle: DCMI 2009b, WIKIPEDIA 2009)

Typ	Element-Name	Beschreibung
ID	Identifizier	Eindeutiger Bezeichner des Dokuments
Technische Daten	Format	Dateiformat, physisches Medium oder Dimension des Dokuments wie beispielsweise Internet Media Type (MIME-Typen)
	Type	Gattung oder Typ des Dokuments
	Language	Sprache des Dokumenteninhalts
Beschreibung des Inhalts	Title	Titel des Dokuments
	Subject	Thema des Dokuments beispielsweise durch Schlüsselwörter
	Coverage	Räumliche oder zeitliche Eingrenzung des Dokuments
	Description	Kurze Beschreibung des Dokuments

Personen und Rechte	Creator	Verantwortlicher Verfasser oder Urheber des Dokuments
	Publisher	Person, die das Dokument veröffentlicht hat
	Contributor	Person, die zur Erstellung des Dokuments beigetragen hat
	Rights	Informationen über die Rechte, die am Dokument gehalten werden bzw. das Dokument betreffen
Vernetzung und Lebenszyklus	Source	Verweis auf ein Dokument, von dem das aktuelle Dokument abgeleitet wurde
	Relation	Verweis auf ein Dokument, das mit dem aktuellen Dokument in Beziehung steht
	Date	Datum oder Zeitspanne, welche mit einem Ereignis im Lebenszyklus des Dokuments verbunden sind

Die Dublin Core Metadaten können verschiedenartig in HTML/XHTML-Webseiten eingebunden werden. Eine Möglichkeit besteht darin, die Beschreibung der jeweiligen Webseite direkt in den Header mittels eines meta-Tags einzubetten. Dieses Tag besitzt das Attribut `name`, in dem ein Metadaten-Element wie beispielsweise *Description*, *Creator* und *Title* mit dem Präfix „DC.“ angegeben werden kann. Das Attribut `content` beinhaltet die entsprechende Beschreibung als freien Text. Weiterhin können im meta-Tag das Schema, die Sprache und das Protokoll, das verwendet wird um die jeweilige Webseite zu empfangen, angegeben werden. Alternativ zu dieser Möglichkeit HTML/XHTML-Webseiten zu beschreiben, kann auch eine Beschreibung mittels RDF/XML (siehe Kapitel 2.3.2.1) erfolgen. Eine solche Beschreibung wird mittels des `link`-Tags in die HTML/XHTML-Webseite referenziert (DCMI 2009c).

Listing 2-2: Beispiel einer XHTML-Notation von DC-Metadaten

```

01 <meta name = "DC.title" content = "Laser Scanning Information System" />
02 <meta name = "DC.creator" content = "Mario Schedler" />
03 <meta name = "DC.language" scheme = "ISO639-2" content = "eng" />

```

3 Anforderungen an die Webapplikation

In diesem Kapitel werden die funktionalen und technischen Anforderungen aufgezeigt, die an die Webapplikation gestellt werden. Zu Beginn dieses Kapitels werden die Nutzer charakterisiert (siehe Kapitel 3.1) und anschließend wird mithilfe eines Use Case Diagramms dargestellt, welche Aufgaben die zukünftigen Nutzer mit der Webapplikation durchführen werden (siehe Kapitel 3.2). Die einzelnen Use Cases werden anschließend mit einer Use Case Beschreibung spezifiziert. Eine detaillierte Beschreibung der Anforderungen, in der auch die Machbarkeit mit in Betracht gezogen wird, wird anschließend dargestellt (siehe Kapitel 3.3). Diese Anforderungen entspringen verschiedenen Quellen: zum einen sind diese in der Motivation der Masterthesis begründet und zum anderen stammen die Anforderungen aus Gesprächen mit zukünftigen Nutzern. Im letzten Abschnitt werden die Anforderungen zusammengefasst und tabellarisch aufgeführt (siehe Kapitel 3.4).

Es sei noch einmal darauf hingewiesen, dass die Webapplikation prototypisch bei der Dow Chemical Company eingeführt wird und daher einige Adaptionen vorgenommen werden müssen. Somit fließen in diesem Abschnitt sowohl Anforderungen ein die allgemeingültig sind, als auch Anforderungen, die speziell für Dow gelten. Auf verschiedene Teilbereiche einer Anforderungsanalyse wie einer Kosten-Nutzen-Analyse, Ressourcen- und Zeitplanung etc. wird an dieser Stelle bewusst verzichtet, da die Webapplikation im Rahmen einer Masterthesis entwickelt wird und hier andere Bedingungen vorherrschen. Auf eine Ist-Analyse wird an dieser Stelle ebenfalls verzichtet und vielmehr auf das einführende Kapitel 1 verwiesen.

3.1 Nutzeranalyse

Bevor die Anforderungen an die Webapplikation aufgeführt und analysiert werden, wird untersucht wer die zukünftigen Nutzer der Webapplikation sind und aus welchen

Arbeitsbereichen sie kommen. Mit diesem Wissen kann die Webapplikation dann zielgerichtet entwickelt werden. Nach GRÜNBACHER (2004) und BEHR (2000) bilden die zukünftigen Anwender der Webapplikation mit ihren Zielen und Erwartungen den Ausgangspunkt für eine Anforderungsanalyse. Die zukünftigen Nutzer werden die Webapplikation verwenden, um nach Laserscanning-Daten und -Projekten zu suchen, sich über diese Daten zu informieren, sie zu visualisieren und mit diesen Daten zu arbeiten. Da sehr viele unterschiedliche Fachdisziplinen wie beispielsweise Archäologie, Maschinen- und Anlagenbau, Architektur, Bauwesen und Facility Management mit Laserscanning-Daten arbeiten, kann die Anwendergruppe als sehr heterogen eingeschätzt werden. Die Webapplikation sollte somit ein breites Publikum ansprechen und ergo eine hohe Usability aufweisen, d. h., die Verständlichkeit, Erlernbarkeit und Bedienbarkeit der Webapplikation sollte bezüglich des Lernaufwands sehr gering sein (vgl. DUMKE, LOTHER, WILLE & ZBROG 2003).

3.2 Anwendungsfälle der Webapplikation

Für die Beschreibung der funktionalen Anforderungen, die an ein System gestellt werden eignen sich besonders Anwendungsfälle (GRÜNBACHER 2004). Anwendungsfälle werden im Umfeld der Unified Modeling Language (UML) als Use Cases bezeichnet. Mit einem Use Case werden Aktivitäten eines Systems aus Sicht seiner Akteure beschrieben. Akteure (UML: Actors) befinden sich außerhalb des zu entwickelnden Systems und interagieren mit diesem (OESTEREICH 2001). Als Akteure kann man sich Personen aber auch andere Systeme vorstellen. Die Beziehungen zwischen Akteuren und Anwendungsfällen werden in einem Anwendungsfalldiagramm (UML: Use Case Diagram) dargestellt. In Abbildung 3-1 ist ein solches Diagramm für die Webapplikation dargestellt.

Aus Sichtweise des Systems existieren zwei Akteure: der User und der Administrator. Der User wird mit diesem System drei Hauptaufgaben erledigen:

- nach Laserscanning-Daten suchen (siehe Tabelle 3-1),
- sich über Laserscanner-Standpunkte bzw. -Projekte informieren (siehe Tabelle 3-2) und letztlich
- Laserscanning-Daten mithilfe eines Viewers visualisieren (siehe Tabelle 3-3 und Kapitel 2.2.3).

Der Administrator hat die Aufgabe dieses System zu pflegen und zu warten (siehe Tabelle 3-4).

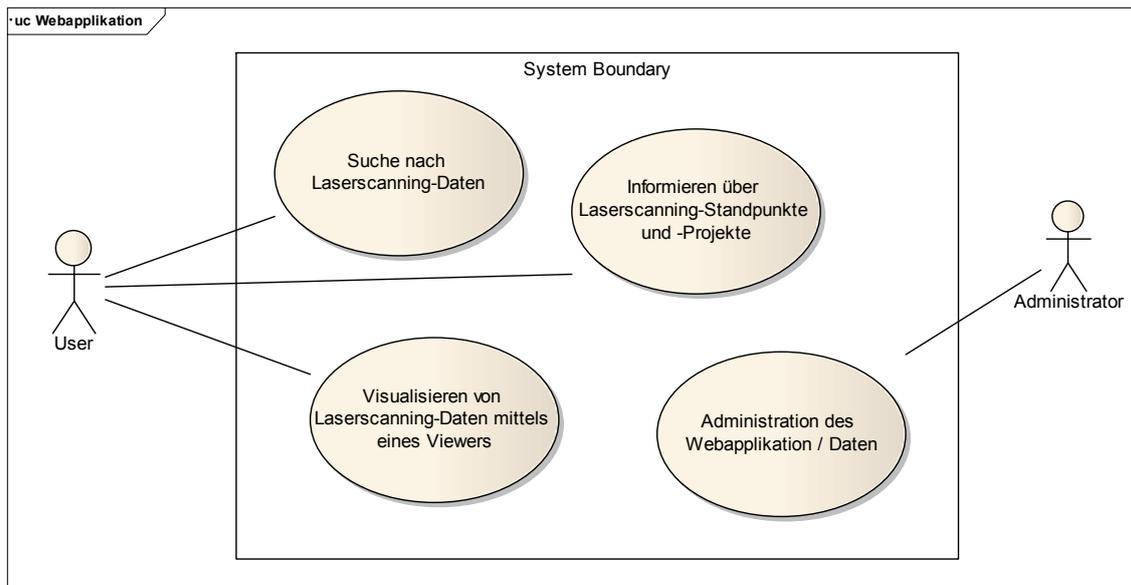


Abbildung 3-1: Anwendungsfalldiagramm der Webapplikation

Die im Anwendungsfalldiagramm (siehe Abbildung 3-1) dargestellten Anwendungsfälle sind in den Tabelle 3-1 bis Tabelle 3-4 beschrieben.

Tabelle 3-1: Beschreibung des Use Case „Suche nach Laserscanning-Daten“ (UseCase1)

ID	UseCase1
Name	Suche nach Laserscanning-Daten
Akteur	User
Vorbedingung	User hat Zugang zur Webapplikation
Auslösendes Ereignis	User benötigt Informationen zu Laserscanning in einem bestimmten Bereich
Hauptszenario	1a. User navigiert anhand einer Baumstruktur zu gewünschten Knoten und wählt diesen aus 1b. System stellt Informationen in einem Fenster dar
Alternativ- und Ausnahmeszenario	2a. User navigiert anhand einer Baumstruktur zu Knoten, der keine Informationen enthält und wählt diesen aus 2b. System stellt in Fenster dar, dass keine Informationen zu diesem Knoten vorhanden sind und führt Knotenbereiche auf, die mit Sachdaten hinterlegt sind 2c. Weiter mit Schritt 1 im Hauptszenario

Tabelle 3-2: Beschreibung des Use Case „Informieren über Laserscanning-Daten und -Projekte“ (UseCase2)

ID	UseCase2
Name	Informieren über Laserscanner-Standpunkte und -Projekte

Akteur	User
Vorbedingung	User hat nach Laserscanning-Daten gesucht und UseCase1 wurde erfolgreich ausgeführt
Auslösendes Ereignis	User benötigt Informationen zu Laserscanning-Daten oder -Projekten um beispielsweise zu prüfen ob vorhandene Daten für Realisierung eines Projektes ausreichend sind
Hauptszenario	1a. User betrachtet Daten, die in einem eigenständigen Fenster aufgelistet sind 1b. User entnimmt die für ihn bedeutenden Informationen
Alternativ- und Ausnahmeszenario	N/A

Tabelle 3-3: Beschreibung des Use Case „Visualisieren von Laserscanning-Daten mittels eines Viewers“ (UseCase3)

ID	UseCase3
Name	Visualisieren von Laserscanning-Daten mittels eines Viewers
Akteur	User
Vorbedingung	User hat nach Laserscanning-Daten gesucht und UseCase1 wurde erfolgreich ausgeführt
Auslösendes Ereignis	User muss die Laserscanning-Daten in einem Viewer betrachten beispielsweise um etwas zu messen
Hauptszenario	1a. User sucht im Fenster, in dem die gesuchten Daten aufgelistet sind, nach einem Hyperlink zu einem Viewer 1b. Der User klickt auf diesen Hyperlink 1c. Ein neues Fenster öffnet sich mit den Laserscanning-Daten
Alternativ- und Ausnahmeszenario	2a. User sucht im Fenster, in dem die gesuchten Daten aufgelistet sind, nach einem Hyperlink zu einem Viewer 2b. Der User klickt auf diesen Hyperlink 2c. Ein neues Fenster öffnet sich und die Laserscanning-Daten werden aufgrund eines fehlenden Plug-Ins nicht dargestellt 2d. Fenster schließen und Laserscanning Plug-In installieren 2e. Nach Installation Browser schließen 2f. Szenario von vorne beginnen 3a. User sucht im Fenster, in dem die gesuchten Daten aufgelistet sind, nach einem Hyperlink zu einem Viewer 3b. Der User klickt auf diesen Hyperlink 3c. Es öffnet sich kein neues Fenster oder das neue Fenster ist leer (Hyperlink ist nicht korrekt) 3d. Administrator über dieses Verhalten informieren 3e. Administrator prüft dies und meldet sich beim User 4a. User sucht im Fenster, in dem die gesuchten Daten aufgelistet sind, nach einem Hyperlink zu einem Viewer 4b. User findet Hyperlink nicht 4c. User stellt Anfrage an Administrator warum kein Hyperlink vorhanden ist 4d. Administrator prüft dies und meldet sich beim User

Tabelle 3-4: Beschreibung des Use Case „Administration der Webapplikation“ (UseCase4)

ID	UseCase4
Name	Administration der Webapplikation
Akteur	Administrator
Vorbedingung	Webapplikation wurde erfolgreich implementiert
Auslösendes Ereignis	Notwendigkeit der System- oder Datenpflege
Hauptszenario	1a. User übergibt neue Daten an Administrator 1b. Administrator pflegt diese in die Datenbank ein 1c. Administrator benachrichtigt User über Fertigstellung der Dateneingabe
Alternativ- und Ausnahmeszenario	2a. User tritt mit Administrator in Kontakt um zum Beispiel einen Verbesserungsvorschlag am System zu unterbreiten 2b. Administrator prüft dies und entscheidet über Implementierung 2c. Administrator implementiert Vorschlag oder sagt User ab 3a. Administrator erkennt Verbesserungspotential am System 3b. Administrator updatet System 3c. Administrator schickt Nachricht über Update an User

3.3 Anforderungen im Detail

3.3.1 Allgemeines

Aufgrund der Anwendung in einem globalen Intranet, soll die Webapplikation als Client-Server Lösung umgesetzt werden. Allerdings soll auch eine mobile Variante der Webapplikation entwickelt werden, was unter bestimmten Voraussetzungen einige Vorteile mit sich bringt. Beispielsweise ist ein schneller Zugriff innerhalb eines Intranets nicht immer gewährleistet und auf die Laserscanning-Daten kann somit nicht oder nur eingeschränkt zugegriffen werden. Zudem können Personen, die keinen Zugriff zu dem jeweiligen Intranet besitzen, diese Daten nicht nutzen. Eine mobile Variante der Webapplikation behebt diese Restriktionen.

Aufgrund des unbestimmten Einsatzbereiches der Webapplikation wird diese komplett in englischer Sprache entwickelt. Das bedeutet, dass sowohl das User Interface als auch die Daten selbst in englischer Sprache vorgehalten werden. Dies ist vor allem bei der Entwicklung des Prototyps sinnvoll, da die Webapplikation Daten über Laserscanning-Projekte aus verschiedenen Ländern beinhaltet. Auf eine Anmeldeprozedur wird bei der Entwicklung des Prototyps verzichtet, da die Anwendung des Prototyps im Intranet

erfolgen soll. Sofern die Webapplikation allerdings über das Internet verfügbar ist, sollte eine Anmeldeseite entwickelt werden.

3.3.2 User Interface

Das User Interface (Benutzerschnittstelle) sollte einfach zu bedienen sein, sich funktional an bestehenden und bekannten Systemen orientieren sowie optisch und funktional im Bereich Web 2.0 angesiedelt sein. Um dies zu erreichen, werden für die Entwicklung der Webapplikation JavaScript-Frameworks GeoExt, Ext JS und OpenLayers verwendet (siehe Kapitel 2.1.6). Diese Frameworks bieten häufig sehr umfassende Komponenten, die auch als Widgets bezeichnet werden (ZAMMETTI 2009) und zudem weit verbreitet sind (vgl. KORDUAN & ZEHNER 2008). Mit diesen Widgets können optisch sehr ansprechende Benutzerschnittstellen entwickelt werden. Die Frameworks haben gleichfalls Vorteile aus Sicht eines Webentwicklers. So kann beispielsweise viel Zeit und Aufwand gespart werden, da auf Bibliotheken mit vorgefertigten Objekten zurückgegriffen werden kann und Applikationen besser modularisiert werden können (WENZ 2007).

Der Zugriff auf die Laserscanning-Daten soll verschiedenartig möglich sein. Primär soll mittels einer Baumstruktur auf diese Daten zugegriffen werden (siehe Kapitel 5.4.3.1). Diese Baumstruktur muss daher dynamisch erzeugt werden, um flexibel gegenüber Veränderungen zu sein. Innerhalb der Baumstruktur soll eine Gruppierung der Daten nach verschiedenen Kriterien wie zum Beispiel Gebäuden, Etagen, Arealen oder Höhenbereichen möglich sein. Neben diesem Zugriff mittels einer Baumstruktur soll auch eine Suche nach geographischen Bezeichnungen wie Städtenamen implementiert werden (siehe Kapitel 5.4.2).

3.3.3 Anforderungen an die Daten

Für die Webapplikation werden flächendeckend Geobasisdaten als Datengrundlage für die Laserscanning-Daten benötigt. Für diese Aufgabe sind vor allem Web-Services wie zum Beispiel Google Maps oder Bing Maps prädestiniert (siehe Kapitel 5.1.2). Vor allem die Orthofotos sind für die Webapplikation bedeutsam, da hier mehr Details zu erkennen sind, als in topografischen Karten oder Straßenkarten. Zusätzlich sollen für die Webapplikation auch eigene Geodaten verwendet werden können. Diese Geodaten sollen dann entweder als Rasterdaten in Form einfacher Bilder oder mithilfe eines Web Map Services in die Webapplikation eingebunden werden. Durch die Verwendung von Geodaten aus Web-Services wird die Administration der Webapplikation erheblich

vereinfacht. Durch die Einbindung mehrere Dienste kann der Nutzer entscheiden welche Basisdaten in der entsprechenden Region besser geeignet sind.

Die eigentlichen Laserscanning-Daten wie Laserscanner-Standpunkte oder Projekt-Daten werden aufgrund ihrer Menge und aus Gründen der Verwaltung in einer Datenbank abgelegt (siehe Kapitel 4). Eine Speicherung der Daten in einer Datei und die Abfrage der Daten mittels der verwendeten JavaScript-Frameworks führen ansonsten zu enormen Performance-Problemen und erschwert zudem die Verwaltung der Daten (siehe Kapitel 5.3).

3.3.4 TruView

Die Webapplikation soll den Zugriff auf die Laserscanning-Daten mit dem webbasierten Viewer TruView (siehe Kapitel 2.2.3.1) ermöglichen. Bevor diese Daten betrachtet werden können, muss ein Plug-In installiert werden. Dies muss in der Webapplikation zum Download angeboten werden. Jeder Nutzer muss zwangsläufig dieses Plug-In installieren, um die Laserscanning-Daten betrachten zu können.

Die TruView-Daten können zentral oder verteilt auf verschiedenen Servern gespeichert werden. Die Art der Datenspeicherung ist abhängig von vielen Faktoren wie zum Beispiel der Anzahl und der Größe der Laserscanning-Daten, dem Zugriffsort und der Bandbreite. Es kann somit keine spezifische Aussage darüber getroffen werden, wo die Daten gespeichert werden sollen. Dies muss vielmehr im Einzelfall entschieden werden. Für die mobile Variante der Webapplikation werden diese Daten auf das Medium kopiert, auf dem sich auch die Webapplikation befindet.

3.4 Resultat der Anforderungsanalyse

Die zu entwickelnde Webapplikation soll entsprechend der heterogenen Nutzergruppe nicht zu kompliziert sein und eine hohe Usability aufweisen. Um dies zu gewährleisten, werden JavaScript-Frameworks (siehe Kapitel 2.1.6) eingesetzt. Die Webapplikation soll auch als mobile Variante Verwendung finden. Geobasisdaten werden von Kartendiensten wie Google Maps bezogen. Daten über Laserscanner-Standpunkte und andere Laserscanning-Daten wie beispielsweise projektrelevante Daten werden in einer Datenbank gespeichert. Als Betriebssystem wird Microsoft Windows vorausgesetzt. In Tabelle 3-5 sind noch einmal alle Anforderungen, die an die Webapplikation gestellt

werden, aufgeführt. Diese Anforderungen lassen sich während der Entwicklung der Webapplikation sowohl zeitlich als auch technisch umsetzen.

Tabelle 3-5: Anforderungen der User an die Webapplikation

Anforderung	Umsetzung
Lösungen	Client-Server und Mobil
Sprache	Englisch
Anmeldeprozedur	Keine notwendig
User Interface / Funktionalität	Web 2.0
Zugriff auf Laserscanning-Daten	Baumstruktur und Gruppierung der Daten nach verschiedenen Gesichtspunkten
Suchfunktion	Suche nach geographischen Namen wie Städten
Geobasisdaten	Flächendeckend wie bei Google Maps und Bing Maps
Sonstige Daten	Einbindung von WMS-Diensten und einfachen Bildern
Speicherung der Daten (Standpunkte und Projekte)	Datenbank
Plug-In für Visualisierung	TruView
Speicherung der TruView-Daten	Unterschiedlich (verteilt, auf USB-Stick etc.)

4 Datenmodellierung

Kapitel 4 widmet sich der Entwicklung eines Datenmodells für die Webapplikation. Zu Beginn dieses Kapitels wird auf die Datenmodellierung bezüglich des Prototyps eingegangen. Anschließend wird das konzeptuelle Datenmodell mithilfe des Entity-Relationship Modells erarbeitet und die einzelnen Entitäten sowie deren Attribute werden erläutert. Aus diesem Entity-Relationship Modell wird das logische Datenmodell entwickelt, das anschließend in das physische Datenmodell für eine MySQL-Datenbank umgesetzt wird.

4.1 Datenmodellierung für den Prototyp

Die Modellierung der Daten ist von verschiedenen Faktoren wie der Datenmenge und der Datenkomplexität abhängig (KORDUAN & ZEHNER 2008). Zudem ist der zukünftige Anwendungsbereich für die Datenmodellierung von entscheidender Bedeutung. So kann sich ein Datenmodell beispielsweise für den Bereich Archäologie von dem des Anlagenbaus erheblich unterscheiden. Das für den Prototyp zu entwickelnde Datenmodell ist ergo auch nur für diesen speziellen Fall gültig und wird in anderen Anwendungsfeldern gegebenenfalls angepasst werden müssen. Um die Webapplikation ohne großen Änderungsaufwand auch in anderen Bereichen anwenden zu können, wird darauf geachtet, dass die Datenlogik von der Applikationslogik weitestgehend getrennt ist.

4.2 Konzeptuelles Datenmodell

Im konzeptuellen Datenmodell werden die Anforderungen der Benutzer an die Daten sowie Entitätstypen, Beziehungen und Einschränkungen in übersichtlicher und kompakter Weise beschrieben (ELMASRI & NAVATHE 2009). Das konzeptuelle Daten-

modell der Webapplikation ist mit Hilfe des Entity-Relationship Modells (ERM) in Information-Engineering Notation dargestellt (siehe Abbildung 4-1). Im Folgenden werden die Entitäten und deren Attribute näher beschrieben.

4.2.1 Entity-Relationship Model

Bei der Erstellung des ERM werden Metadaten in Anlehnung an den ISO 19115 Standard berücksichtigt (siehe Kapitel 2.3.2.2). Es wird darauf geachtet, dass die Administration des Systems und insbesondere die Datenpflege möglichst einfach sind und zugleich wenig Aufwand verursachen. Hiermit wird auch impliziert, dass das Datenmodell auf ein Minimum an Entitäten und Attributen beschränkt wird. In einigen Bereichen wird denormalisiert, da hierdurch das Modellieren von unnötigen und kleinen Entitäten vermieden wird und das Datenmodell übersichtlicher und einfach strukturiert bleibt. Die Entität *Laser Scanning Project* beispielsweise könnte in der Realität mehrere Entitäten *Database* besitzen. Bei der Datenmodellierung für den Prototyp wird hierauf verzichtet, da dies im entsprechenden Textfeld vermerkt werden kann und innerhalb eines Projektes stets nur eine Datenbank genutzt wird. Zudem könnten mehrere Personen ein Projekt verwalten bzw. Daten erfassen. Allerdings ist es ausreichend nur einen Ansprechpartner zu verwalten.

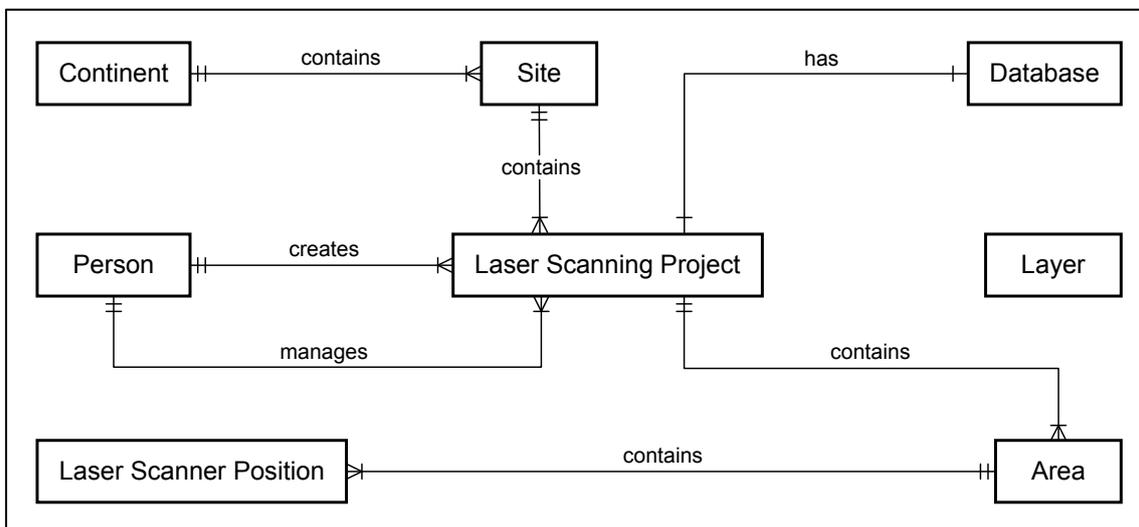


Abbildung 4-1: Entity-Relationship Model der Webapplikation

Bei Attributen, die ein Datum darstellen, wird innerhalb der ISO 19115 auf die ISO 8601 (Data elements and interchange formats - Information interchange - Representation of dates and times) verwiesen. Als numerisches Datenformat für Zahlen wird *JJJJ-MM-TT* verwendet.

4.2.2 Gemeinsame Attribute der Entitäten

Alle Entitäten haben diverse Attribute wie beispielsweise Identifier (*id*) und Namen (*name*) gemeinsam. Um diese nicht bei jeder Beschreibung der Entitäten aufzuführen, sind die Attribute in Tabelle 4-1 aufgeführt. Dennoch werden bei einigen Entitäten Attribut wie beispielsweise *refdate* aufgefasst und genauer erläutert, was an dieser Stelle darunter zu verstehen ist. Für die Navigation zwischen den verschiedenen Laserscanning-Projekten wird eine Baumstruktur, ähnlich der des Windows Explorers, verwendet. Hierfür werden bei allen Entitäten, außer *Database*, *Person* und *Layer* zusätzlich drei weitere Attribute namens *leaf*, *parent* und *iconcls* modelliert. Mithilfe der beiden Attribute *leaf* und *parent* kann die Baumstruktur (siehe Kapitel 5.4.3.1) weitestgehend unabhängig von der Programmierung mittels JavaScript realisiert werden. Das Attribut *iconcls* dient der Speicherung von CSS-Klassen, durch die den verschiedenen Knoten des Baumes charakteristische Symbole zugewiesen werden können (siehe Kapitel 5.4.3.3).

Bei dem Attribut *language* verweist die ISO 19115 auf die ISO 639-2 (Codes for the representation of names of languages - Part 2: Alpha-3 code), wo jeder Sprache ein Code bestehend aus drei Buchstaben zugeordnet wird. Bei der englischen Sprache wird beispielsweise als Code *ENG* angegeben.

Tabelle 4-1: Gemeinsame Attribute aller Entitäten der Webapplikation

Attribut	Beschreibung
<u>id</u>	Primärschlüssel
name	Bezeichnung, unter der die Ressource bekannt ist
topic	Thematische Einordnung des Datenbestands
description	Kurze, beschreibende Zusammenfassung des Inhalts der Ressource
dsdate	Datum, zu dem der Metadatensatz erzeugt bzw. geändert wurde
language	Im Projekt verwendete Sprache
refdate	Datum der Erzeugung oder Überarbeitung des Datensatzes
leaf	Angabe ob Element ein Kind-Element besitzt
iconcls	Charakteristisches Symbol für Element bzw. Gruppe von Elementen
parent	Identifier des Elternelements

4.2.3 Entität Laser Scanner Position

Im Mittelpunkt des ERM steht die Entität *Laser Scanner Position*, die den Standpunkt eines Laserscanners mittels Längengrad (*longitude*) und Breitengrad (*latitude*) repräsentiert. Die Koordinaten der Laserscanner-Standpunkte werden mit einer Genauigkeit von sechs Nachkommastellen in Dezimalschreibweise angegeben. Somit werden die Standpunkte lagerichtig, im Dezimeterbereich dargestellt. Das Scandatum wird im Attribut *refdate* erfasst um später Aussagen zur Aktualität und Verwertbarkeit der Scans treffen zu können. Das Attribut *viewer_url* beinhaltet die URL zu den Laserscanning-Daten, die mittels webbasierten Viewer dargestellt werden können. Die zusätzlichen Eigenschaften eines Laserscanner-Standpunktes werden in der folgenden Tabelle dargestellt und kurz erläutert.

Tabelle 4-2: Zusätzliche Attribute der Entität Laser Scanner Position

Attribut	Beschreibung
longitude	Längengrad
latitude	Breitengrad
viewer_url	URL zum webbasierten Viewer des Laserscanner-Standpunktes

4.2.4 Entität Area

Jeder Laserscanner-Standpunkt wird einem Bereich (*Area*) zugeordnet. Hierbei kann es sich um Teilflächen des gescannten Areals handeln aber auch um bestimmte Höhenbereiche. Teilflächen können beispielsweise nach Gebäudenamen, Höhenbereichen nach Höhenintervallen wie Level 100 m, Level 110 m etc. benannt werden. Die Entität *Area* besitzt keine zusätzlichen Attribute.

4.2.5 Entität Database

Mit der Entität Database werden Daten der Laserscanning-Datenbanken gespeichert. Diese Datenbanken beinhalten in der Regel alle Scans in Form eines registrierten Gesamtmodells. Das Attribut *crs* bezeichnet das Bezugssystem (engl. Coordinate Reference System) auf das sich sowohl die Standpunkte als auch die Datenbank beziehen. Wenn möglich, sollten hier immer EPSG-Codes angegeben werden. Dieses Attribut ist transitiv nicht abhängig und sollte gemäß der dritten Normalform als eigständige Entität modelliert werden. Das gilt analog für das Attribut *format*, welches das Datenbankformat spezifiziert. Aus den in Kapitel 4.2.1 aufgeführten Gründen

werden diese Attribute nicht in Form von Entitäten modelliert. Bei der Angabe des Referenzsystems wird häufig anstelle eines globalen Bezugssystems ein lokales Bezugssystem verwendet, das zudem häufig von Projekt zu Projekt unterschiedlich definiert ist. Die Entität Database besitzt zusätzlich die in Tabelle 4-3 aufgeführten Attribute.

Tabelle 4-3: Zusätzliche Attribute der Entität Database

Attribut	Beschreibung
size	Speicherbedarf der Datenbank in GB
crs	Referenzsystem in denen die Scans registriert sind
format	Herstellerspezifische Bezeichnung der Datenbank

4.2.6 Entität Person

Mit der Entität *Person* sind Personen gemeint, die an Laserscanning-Projekten beteiligt sind. Das sind zum einen der Verantwortliche für die Datenerfassung (*Originator*) und zum anderen der Verantwortliche für die Verwaltung der Daten nach der Erfassung (*Provider*). Das zusätzliche Attribut *organization* weist keine transitive Abhängigkeit bezüglich der Person auf und müsste gemäß der dritten Normalform als eigenständige Entität modelliert werden. An dieser Stelle wird zur Vereinfachung des Datenmodells und der Verwaltung der Daten denormalisiert. Für die Kontaktaufnahme der User der Webapplikation mit einer Person wird das Attribut *email* modelliert. Auf die Modellierung umfangreicher Kontaktdaten wie Telefonnummer, Webseite etc. wird verzichtet.

Tabelle 4-4: Zusätzliche Attribute der Entität Person

Attribut	Beschreibung
email	Email-Adresse der Person
organization	Name des Unternehmens

4.2.7 Entität Laser Scanning Project

Die Entität *Laser Scanning Project* stellt ein Laserscanning-Projekt dar, welches einer *Site*, also einem Werk, zugeordnet wird. Zu einem Projekt gehört eine Datenbank (Entität *Database*) und ein registriertes Gesamtmodell bestehend aus allen Scans. Das Attribut *refdate* hält hier fest wann das Laserscannings-Projekt begonnen hat.

4.2.8 Entitäten *Site* und *Continent*

Die Entitäten *Site* und *Continent* dienen der Strukturierung der Laserscanner-Standpunkte in einem globalen Kontext. Diese Entitäten besitzen keine zusätzlichen Attribute.

4.2.9 Entität *Layer*

Die Entität *Layer* dient der Speicherung von Daten, die benötigt werden um Datenquellen in OpenLayers mithilfe der Klasse `OpenLayers.Layer` und abgeleiteten Klassen darzustellen. OpenLayers unterscheidet *Layer* in zwei Kategorien: *Basislayer* und *Overlays* (siehe Kapitel 5.1.2). *Basislayer* sind *Layer* von denen immer nur einer zur gleichen Zeit dargestellt werden kann. *Overlays* hingegen sind *Layer*, die oberhalb eines *Basislayers* dargestellt werden. Zur gleichen Zeit können mehrere *Overlays* dargestellt werden. Ob es sich um einen *Basislayer* oder einen *Overlay* handelt, wird im Attribut *baselayer* festgesetzt. Innerhalb der Webapplikation stellen die Kartendaten von Google Maps und Bing Maps *Basislayer* dar. Kleinere Bilder und die Laserscanner-Standpunkte hingegen werden durch *Overlays* dargestellt. Zusätzlich zu den Attributen aus Tabelle 4-1 beinhaltet die Entität *Layer* die in Tabelle 4-5 aufgeführten Attribute.

Tabelle 4-5: Zusätzliche Attribute der Entität *Layer*

Attribut	Beschreibung
class	Layer-Klasse
type	Vordefinierte Kartentypen von Google Maps und Bing Maps
url	URL des WMS-Layers oder Pfadangabe eines Bildes
spherical_mercator	Wird der Layer in Mercator-Projektion dargestellt oder nicht
bounds	Räumliche Ausdehnung des Layers
zoomlevels	Anzahl der Zoomstufen
baselayer	Ist der Layer ein Basislayer oder nicht
size	Größe des Bildes in Pixeln

Die in Tabelle 4-5 dargestellten Attribute der Entität *Layer* können jedoch nicht auf alle Datenquellen angewendet werden, da jeder `OpenLayers.Layer` eigene und individuelle Eigenschaften besitzt. Ein WMS-Dienst beispielsweise benötigt das Attribut *size* nicht, da dieses nur benötigt wird, wenn ein Bild (`OpenLayers.Layer.Image`) als Datenquelle verwendet wird.

Die Entität *Layer* besitzt im konzeptuellen Datenmodell (siehe Abbildung 4-1) keine Beziehungen zu anderen Entitäten, da diese Entität auf die Speicherung von „Geometriedaten“ und nicht wie die anderen Entitäten auf die Speicherung von Sachdaten abzielt. Eine Beziehung zu Entitäten wie *Project* oder *Area* ist zwar möglich, stellt sich aber als nicht sinnvoll dar und ist für die Webapplikation unnötig.

4.3 Logisches und physisches Datenmodell

Das konzeptuelle Datenmodell wird durch ein logisches Datenmodell für eine MySQL Datenbank realisiert (siehe Abbildung 4-2). Hierzu wurde eine separate Datenbank erzeugt in der insgesamt acht Tabellen und ein View enthalten sind. Jeder Tabelle und auch dem View wurde das Präfix „TLS_“ vorangestellt, um die Zusammengehörigkeit der Tabellen zu einer Applikation zu verdeutlichen. Die Datentypen wurden entsprechend den praktischen Gegebenheiten definiert.

Der oben genannte View dient der Navigation innerhalb der Baumstruktur (siehe Kapitel 5.4.3.1). Um eine solche Navigation zu ermöglichen, müssen die relational gespeicherten Daten in eine hierarische Struktur überführt werden. Hierzu existieren zwei bedeutende Modelle bzw. Herangehensweisen: das *Nested-Set Model* und das *Adjacency List Model* (VAN TULDER 2003). Beim *Nested-Set Model* wird bei jedem Datensatz sowohl der Vorgänger als auch der Nachfolger gespeichert um eine Baumstruktur zu erzeugen. Das *Adjacency List Model* verfolgt einen anderen Ansatz. Hier wird für jeden Datensatz lediglich ein Zeiger (Attribut *id*) auf das Elternelement gespeichert. Das Wurzelement besitzt entsprechend einen Null-Wert. Für die Webapplikation wird das *Adjacency List Model* verwendet, da es relativ einfach zu implementieren und zu verwalten ist. Das *Adjacency List Model* ist gegenüber dem *Nested-Set Model* weniger performant, was bei der Webapplikation allerdings nicht sehr bedeutsam ist. Bei jedem Datensatz wird somit das Elternelement (*parent*) gespeichert. In der folgenden Abbildung ist das logische Datenmodell, das für die Webapplikation verwendet wird, dargestellt.

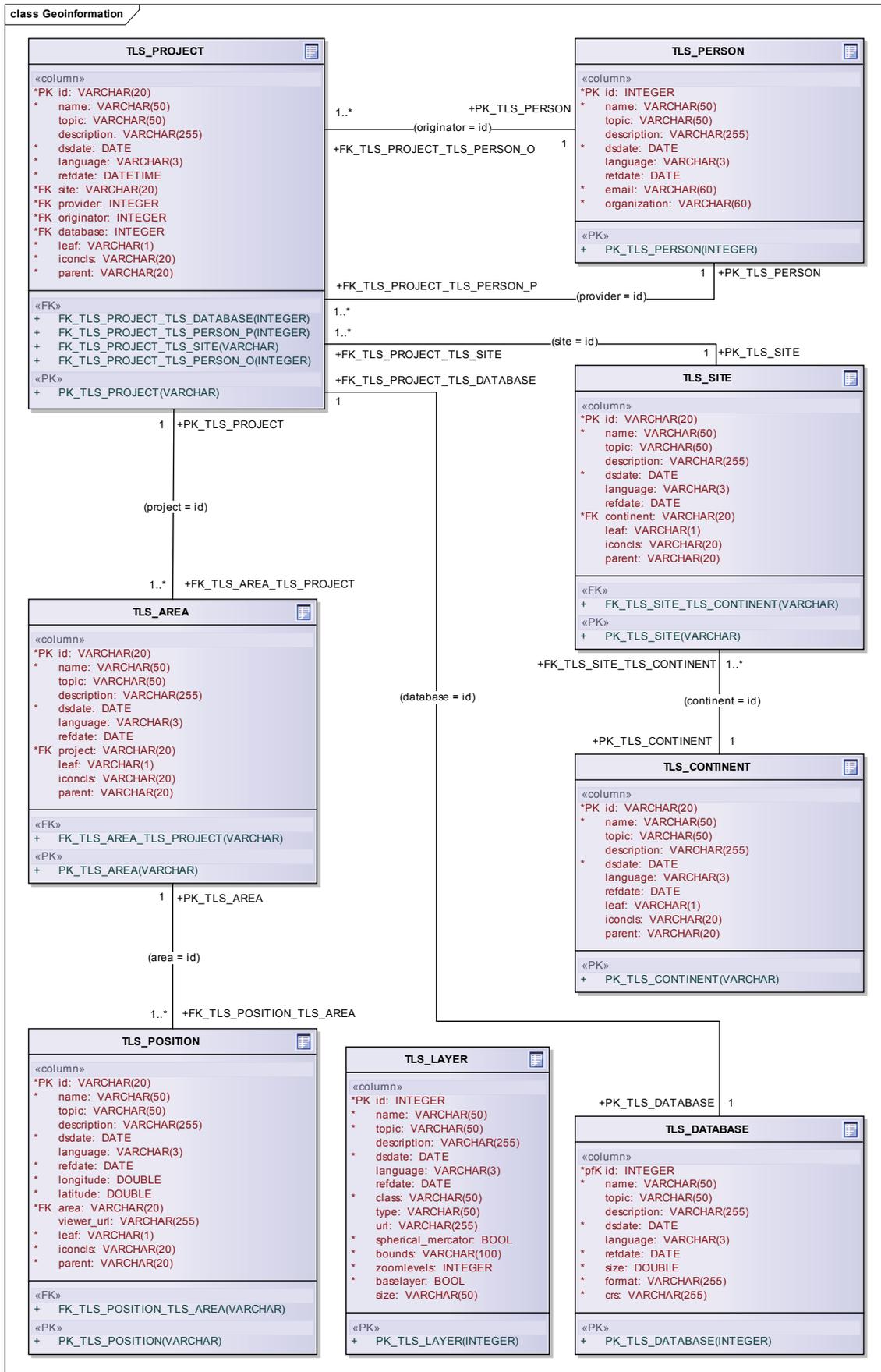


Abbildung 4-2: Logisches Datenmodell für eine MySQL Datenbank

In Abbildung 4-2 sind nicht alle Festlegungen des Datenmodells aufgeführt. Um bei Anfragen des Clients an die Datenbank die Antwortzeiten möglichst gering zu halten, wurden die Attribute *id* und *parent* bei den entsprechenden Tabellen indexiert. Zur Vermeidung von fehlerhaften Eingaben wurden CHECK-Bedingungen eingeführt, obwohl diese vom MySQL-DBMS zwar geparkt aber anschließend ignoriert werden (MYSQL 2010). In der folgenden Tabelle sind CHECK-Bedingungen aufgeführt, die somit implementiert werden könnten aber keine Wirkung zeigen würden. Bei Bedarf müssen einige Wertebereiche wie beispielsweise die Spalte *class* in der Tabelle *TLS_LAYER* erweitert werden.

Tabelle 4-6: Mögliche CHECK-Bedingungen im Datenmodell

Tabelle	Spalte	Bedingung
TLS_LAYER	class	Wertebereich: „WMS“, „Image“, „Google“, „GML“, „VirtualEarth“
	topic	darf nur Wert „Base Layer“ oder „Overlay“ annehmen
TLS_PERSON	email	muss Symbol „@“ und einen Punkt beinhalten
TLS_PROJECT	language	Bis dato nur Wert „ENG“ zulässig
TLS_POSITION	longitude	Wertebereich zwischen -180° und 180°
	latitude	Wertebereich zwischen -90° und 90°

5 Die Webapplikation

In diesem Kapitel wird die Entwicklung der Webapplikation beschrieben. Im ersten Abschnitt wird Grundlegendes beispielsweise zu Bezugssystemen, Kartendiensten und der Darstellung der Laserscanner-Standpunkt besprochen. Die folgenden beiden Abschnitte erläutern den groben Aufbau und die Funktionsweise des Prototyps. Der detaillierte Aufbau und die Benutzeroberfläche wird im Abschnitt User Interface beschrieben. Hier wird auch erläutert, wie die einzelnen Bereiche der Webapplikation miteinander interagieren. Abschließend wird der Prototyp in seiner Gesamtheit vorgestellt.

Innerhalb dieses Kapitels wird weitgehend auf die Angabe von Quelltext verzichtet, da dies den Lesefluss stört. Exemplarisch werden jedoch Auszüge zum besseren Verständnis eingefügt. Wenn in diesem Teil der Arbeit Klassen der oben genannten JavaScript-Frameworks aufgeführt werden, wird immer das entsprechende JavaScript-Framework, getrennt durch einen Punkt, der Klasse vorangestellt und durch eine eigene Schriftart hervorgehoben. Somit ist es für den Leser nachvollziehbar, welchem Framework die Klasse zugeordnet ist.

5.1 Grundgedanken

5.1.1 JavaScript-Frameworks und der Prototyp

Die Verwendung von JavaScript-Frameworks wie beispielsweise ExtJS und GeoExt ermöglicht eine weitgehende browser- und browserversionsunabhängige Entwicklung von webbasierten Applikationen. Eine solche Applikation kann ergo in allen gängigen Webbrowser verwendet werden, wobei der Fokus bei der prototypischen Entwicklung auf den Internet Explorer und dem Mozilla Firefox gelegt wird. Die Webapplikation per se ist auch in anderen Browsern, die von den verwendeten JavaScript-Frameworks

unterstützt werden, lauffähig. Allerdings wird der Viewer TruView nur von diesen beiden Browserderivaten unterstützt. Die Entwicklung der Webapplikation erfolgt mit den JavaScript-Frameworks ExtJS 3.2, OpenLayers 2.9 und GeoExt 0.7.

5.1.2 Basislayer und Overlays

In die Webapplikation wurden als Kartengrundlage (Basislayer) Webservices wie Bing Maps und Google Maps integriert. Somit konnten die Laserscanner-Standpunkte von den Basislayern getrennt werden, was das Hinzufügen bzw. Löschen von Standpunkten erleichtert. Sie werden mittels der Klasse `OpenLayers.Layer` in die Webapplikation eingebunden. Wenn in bestimmten Bereichen die Auflösung der Rasterdaten zu gering, kein Kartenmaterial vorhanden oder veraltet ist, können WMS-Dienste in die Webapplikation eingebunden werden. Exemplarisch wurde der Layer `Bluemarble`, der durch einen WMS-Dienst der NASA bereitgestellt wird, in die Webapplikation eingebunden. Auch das Einbinden von zusätzlichen Daten in Form von Rasterdateien (`OpenLayers.Layer.Image`) oder GML-Dateien (`OpenLayers.Layer.GML`) wird ermöglicht. Diese Daten werden in einem Unterordner der Webapplikation abgelegt und wie die anderen `OpenLayers.Layer` in der Datenbank verwaltet (siehe Kapitel 4.2.9). In Tabelle 5-1 sind alle Rasterdatensätze, die in der Webapplikation verfügbar sind, aufgelistet. Exemplarisch wurden auch einige Rasterdatensätze (`OpenLayers.Layer.Image`) in die Webapplikation eingefügt. Diese Rasterdaten sind projektspezifisch und werden deshalb nicht an dieser Stelle aufgeführt.

Tabelle 5-1: Kartendienste und deren Bezeichnung in der Webapplikation

Kartendienst	Layerotyp		Bezeichnung
Bing Maps	Karte (Shaded)		Bing Maps – Road
	Luftbild		Bing Maps – Aerial
	Hybrid		Bing Maps – Hybrid
Google Maps	Karte		Google Maps – Streets
	Luftbild		Google Maps – Satellite
	Hybrid		Google Maps – Hybrid
	Physical		Google Maps – Physical

Global Imagery	Bluemarble		NASA – Bluemarble
----------------	------------	---	-------------------

Alle `OpenLayers.Layer` werden zentral in der MySQL-Datenbank gespeichert, um die Anwendungslogik von der Datenlogik zu trennen und somit Erweiterungen der Webapplikation zu vereinfachen (siehe Kapitel 4.1 und 4.2.9). Die Layer werden beim Starten der Webapplikation mithilfe eines PHP-Skripts automatisch geladen.

Die Verwendung der Kartendienste Google Maps und Bing Maps ist nicht ohne weiteres möglich – es müssen die Nutzungsbedingungen von Google bzw. Microsoft beachtet werden. Da die Webapplikation in einem Intranet – also einem nicht frei zugänglichen Bereich – genutzt werden soll, fallen entsprechend den Geschäfts- und Nutzungsbedingungen von Google Maps bzw. Bing Maps Kosten an. Bei Google Maps wird eine kostenpflichtige Google Maps API Premier Lizenz und für die Verwendung von Bing Maps wird eine Bing Maps Enterprise Lizenz benötigt. Sofern diese erworben wird, kann die Webapplikation im Intranet verwendet werden.

5.1.3 Bezugssystem

Die Kartendienste Bing Maps und Google Maps verwenden für die Anzeige und Eingabe von Koordinaten grundsätzlich geographische Längen- und Breitenangaben, angegeben in der Einheit Grad in Dezimalschreibweise. Die Koordinaten beziehen sich dabei auf das geodätische Referenzsystem World Geodetic System 1984 (*WGS84*; EPSG-Code: 4326). Die zugrundeliegenden Rasterdaten sind mittels Mercator-Projektion in der Ebene abgebildet. Allerdings wird bei dieser Projektion kein Ellipsoid sondern ein Sphäroid verwendet. Im Umfeld von OpenLayers wird von einer sphärischen Mercator-Projektion (*Spherical Mercator*) gesprochen, die inoffiziell mit dem EPSG-Code 900913 und offiziell mit dem EPSG-Code 3857 bezeichnet wird (OPENLAYERS 2010b).

Mit der Mercator-Projektion werden alle Bereiche der Erdoberfläche, ausgenommen die der Pole, dargestellt. Da an den Polen in der Regel sehr selten Laserscanning-Projekte stattfinden, ist die Mercator-Projektion für die Webapplikation sehr gut geeignet. Es sei noch darauf hingewiesen, dass OpenLayers bis dato keine Umprojektion der Daten von Google Maps und Bing Maps bietet; ergo muss ohnehin die sphärische Mercator-Projektion für die Webapplikation verwendet werden. Die Koordinaten der Laserscanner-Standpunkte und die Overlays (siehe Kapitel 4.2.9) werden ebenfalls im geodäti-

schen Bezugssystem *WGS84* vorgehalten und bei der Visualisierung entsprechend in das sphärische System mittels der Klasse `OpenLayers.Projection` transformiert.

5.1.4 Laserscanner-Standpunkte

Die Lage der Laserscanner-Standpunkte wird in der Webapplikation symbolisch durch einen gelben Kreis dargestellt. Selektierte Standpunkte hingegen werden durch einen blauen Kreis repräsentiert. Durch diese Farbdarstellung können die Standpunkte, insbesondere wenn sich Orthofotos im Hintergrund befinden, relativ gut erkannt werden. Aber auch bei Straßenkarten sind diese Punkte gut erkennbar. Der Durchmesser dieser Kreise beträgt 6 mm (siehe Tabelle 5-2).

Tabelle 5-2: Darstellung der Laserscanner-Standpunkte in der Karte

Laserscanner-Standpunkt	Darstellung
Nicht selektiert	
Selektiert	

Die Laserscanner-Standpunkte befinden sich häufig in unterschiedlichen Bereichen wie beispielsweise verschiedene Etagen eines Gebäudes oder unterschiedlichen Baufeldern. Eine Gruppierung der Standpunkte nach örtlichen Gegebenheiten wird somit erfolgen, um Überlagerungen von Standpunkten zu vermeiden. Auf eine explizite Darstellung von Höhenbereichen beispielsweise durch geometrisch- oder farblich unterschiedliche Darstellungen der Standpunkte wird verzichtet. Die Einteilung der Laserscanner-Standpunkte in *Areas* erscheint für die Webapplikation als sehr zweckmäßig, da die Teilbereiche häufig auch den Bereichen entsprechen, in den Umbaumaßnahmen stattfinden. Auch die Performance von `OpenLayers` sinkt bei der Darstellung von mehreren Hundert Standpunkten und würde sich negativ auf die Performance auswirken.

5.1.5 Datenabfrage

Für die Kommunikation mit der MySQL-Datenbank wird PHP verwendet. PHP ist ein rekursives Akronym und steht für PHP Hypertext Preprocessor. Es handelt sich um eine weit verbreitete Open-Source Skriptsprache, die speziell im Bereich der Webprogrammierung eingesetzt wird und in HTML eingebettet werden kann (PHP 2010). Als Datenaustauschformat wird die JavaScript Object Notation (JSON) verwendet, da ExtJS

zum Datenaustausch vorrangig *JSON* unterstützt. Durch die Verwendung von AJAX-Technologie werden nur Daten abgefragt, die der User auch tatsächlich anfragt. Mit der Datenbank wird immer dann kommuniziert, wenn in der Region *West* ein noch nicht aufgeklappter Knoten aufgeklappt wird (siehe Kapitel 5.4.3). Ein serverseitiges PHP-Skript fragt die Kindknoten des angeklickten Knotens aus der Datenbank ab und liefert diese zurück an den Baum. Hierbei wird mittels des Knoten-Attributes *leaf* überprüft ob die Kindknoten weitere Kindknoten haben (*true*) oder nicht (*false*). Haben sie keine Kindknoten, dann handelt es sich um einen Laserscanner-Standpunkt.

Bei der Initialisierung der Webapplikation werden standardmäßig alle `OpenLayers.Layer` (siehe Kapitel 5.4.3.2) und die Kontinente (siehe Kapitel 5.4.3.1) geladen und anschließend in den entsprechenden Baumstrukturen der Region West (siehe Kapitel 5.4.3) dargestellt.

5.2 Struktur der Webapplikation

Die Webapplikation setzt sich, neben den zahlreichen Dateien die von den JavaScript-Frameworks verwendet werden, aus drei eigenen Dateien zusammen, die die gesamte Applikationslogik beinhalten. Dies sind: *tls.php*, *tls.css* und *tls_db.php* (siehe Abbildung 5-1). Das eigentliche Herzstück der Webapplikation ist die Datei *tls.php*, in der fast die gesamte Anwendungslogik der Webapplikation enthalten ist. Hier werden im Header alle benötigten JavaScript-Bibliotheken, CSS-Dateien (Cascading Style Sheets) und Metadaten der Webseite eingebunden. Zudem sind hier die notwendigen Skripte für Google Maps und Bing Maps enthalten. Kapitel 5.4 erläutert die oben genannten Dateien respektive die Funktionsweise der Webapplikation im Detail.



Abbildung 5-1: Struktur der Dateien für die Webapplikation

In der Datei *tls.php* wird die Datei *tls_db.php* aufgerufen um die Baumstruktur, die für die Navigation innerhalb der Laserscanning-Projekte zuständig ist, zu erzeugen (siehe Kapitel 5.4.3.1). Die Datei *tls.css* beinhaltet alle Stylesheets (CSS-Datei), mit denen verschiedene Elemente bzw. Bereiche der Webapplikation in ihrem Erscheinungsbild verändert werden. Dies sind im Einzelnen:

- die Koordinatenanzeige der Mausposition,
- die grafischen Symbole innerhalb der beiden Baumstrukturen (siehe Kapitel 5.4.3) und
- die Formatierung der Regionen *North* und *West* (siehe Kapitel 5.4.2 und 5.4.3).

Zur Beschreibung der Webapplikation mittels Metadaten wird das *Dublin Core Metadata Element Set* in der Version 1.1 verwendet (siehe Kapitel 2.3.2.3). Da es sich bei der Webapplikation um lediglich eine einzige Webseite handelt, werden die Metadaten in XHTML-Notation entsprechend der Tabelle 2-4 direkt im Header der Datei *tls.php* integriert. Auf das Metadaten-Element *DC.source* wird verzichtet, da die Webapplikation nicht von einem anderen Dokument abstammt. Exemplarisch sind nachstehend einige meta-Tags, wie sie in der zentralen PHP-Datei definiert sind, aufgeführt.

Listing 5-1: Metadaten-Elemente der Startseite der Webapplikation

```
01 <head>
02 <meta name="DC.title" content="Laser Scanning Information System" />
03 <meta name="DC.creator" content="Mario Schedler" />
04 <meta name="DC.type" content="Laser Scanning" />
05 <meta name="DC.language" scheme="ISO639-2" content="ENG" />
06 ...
07 </head>
```

5.3 Prinzipielle Funktionsweise

Die Webapplikation basiert auf dem Client-Server Prinzip, wobei die Anfragen eines Clients an mehrere Server gestellt werden können (siehe Abbildung 5-2). In der Webapplikation fragt der Client einerseits den Server an, auf dem sich die Webapplikation und die MySQL-Datenbank befinden. Andererseits stellt der Client Anfragen an Server auf denen die grundlegenden Kartendaten wie Basislayer von Google Maps oder die TruView-Daten für die Visualisierung gespeichert sind. Die Daten können somit verteilt sein, was entsprechend in der Datenbank beispielsweise in Form einer URL festgesetzt werden muss. Eine Internet-Verbindung ist für die Webapplikation somit

sehr bedeutend, allerdings nicht unbedingt notwendig, da auch Intranet-basierte WMS-Dienste und digitale Karten in die Webapplikation eingebunden werden können.

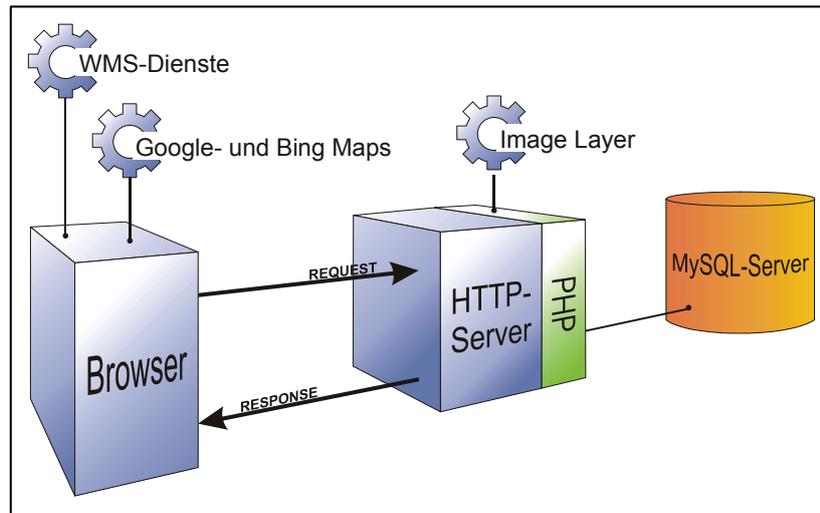


Abbildung 5-2: Client-Server Prinzip der Webapplikation

Zu dieser Client-Server-Lösung existiert noch eine mobile Variante der Webapplikation, die ebenfalls als Client-Server-Lösung betrachtet werden kann. Hierfür wird die Software *Server2Go* verwendet. *Server2Go* ist ein Webserver der ohne Installation und Konfiguration auch auf schreibgeschützten Medien wie DVDs genutzt werden kann (HABERKERN 2010). Er verwendet gängige Open Source Software wie Apache, PHP, SQLite und MySQL. Dies hat den Vorteil, dass die Webapplikation nur einmal entwickelt bzw. der Quelltext für die mobile Nutzung nicht geändert werden muss. Einmalig muss der *Server2Go* konfiguriert werden. Zukünftig muss lediglich die Webapplikation auf das mobile Medium kopiert werden, die Datenbank exportiert und anschließend importiert werden. Die mobile Webapplikation ist somit auf vielen Medien wie USB-Sticks, CD-ROMs und DVDs funktionsfähig, was einen erheblichen Mehrwert in sich birgt.

In Abbildung 5-3 ist die Funktionsweise der Webapplikation in beiden Varianten detailliert dargestellt. Die mobile Variante ist mit der Server-Variante identisch. Als Mastersystem wird allerdings das Client-Server System betrachtet, weshalb auch von diesem System die Daten zum mobilen System exportiert werden. Im Mastersystem werden somit alle Daten vorgehalten – in der mobilen Lösung häufig nur Projektbezogene Daten. Die Systemdateien müssen alle exportiert werden, wohingegen die Daten der Laserscanning-Projekte und Layer (Web Services, GML-Dateien, Images und Kartendienste wie Google Maps) nicht zwangsweise exportiert werden müssen. Über

welche Funktionen der Client verfügt und wie dessen User Interface implementiert ist, wird in den nächsten Kapiteln erläutert.

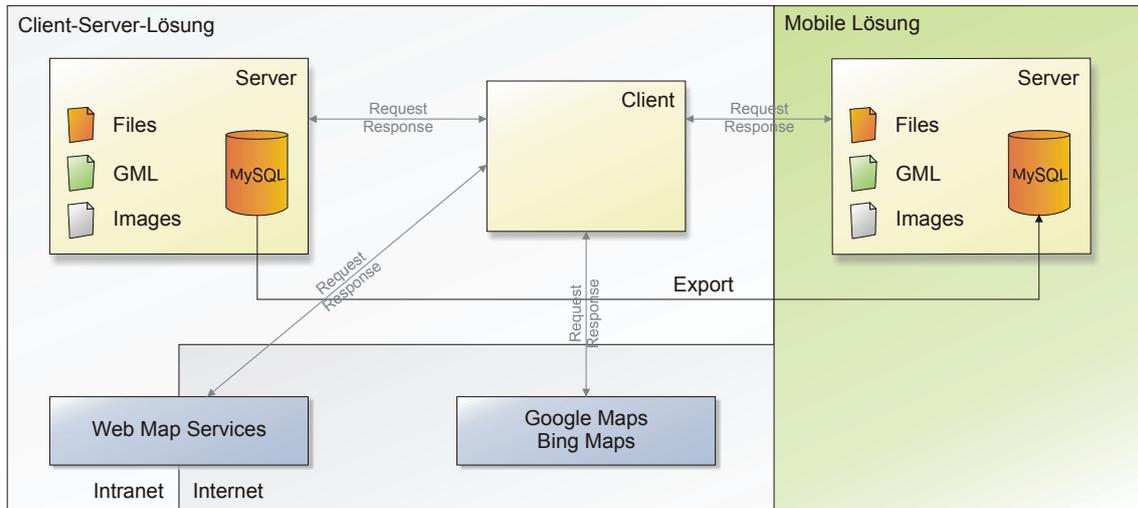


Abbildung 5-3: Funktionsweise der Webapplikation (Client-Server und mobile Lösung)

5.4 User Interface

Für die Gestaltung des User Interfaces bieten Ext JS und GeoExt optisch sehr ansprechende Widgets, so dass man sich im Wesentlichen nur noch mit der Anordnung der grafischen Komponenten der Webapplikation und dem Event-Handling befassen muss. Standardmäßig werden diese Widgets in blau dargestellt, was aber geändert werden kann. Bei der Entwicklung des Prototyps werden die Standard-Einstellungen genutzt. Das gesamte User Interface ist, wie auch die Daten an sich, in englischer Sprache verfasst. In den folgenden Abschnitten wird das verwendete Layout und dessen Komponenten sowie darüber hinaus deren Interaktion beschrieben.

5.4.1 Layout und Ext JS

Den Kern für die Gestaltung des User Interfaces stellt das Layout-Konzept von Ext JS dar, das an dieser Stelle kurz erläutert wird. Der Namespace *Ext.layout* beinhaltet zahlreiche Klassen zur Gestaltung von Layouts für Webseiten wie Spaltenlayout (*Ext.layout.ColumnLayout*), Tabellenlayout (*Ext.layout.TableLayout*) etc. Ein solches Layout setzt sich aus einer bestimmten Anzahl von Containern (*Ext.Container*) wie beispielsweise Fenstern, Menüs, Toolbars, Formularen und Baumstrukturen zusammen. Zudem können Container weitere Layout-Komponenten beinhalten. Ein besonderer Container ist der Viewport (Klasse *Ext.Viewport*). Dieser repräsentiert den gesamten

Bereich innerhalb eines Browserfensters in dem eine Webseite dargestellt wird, also den sogenannten Browser-Viewport. Der `Ext.Viewport` rendert sich automatisch in das `body`-Tag der Webseite, so dass der gesamte Browser-Viewport eingenommen wird.

Für das Layout der Webapplikation wird ein Border Layout (`Ext.layout.BorderLayout`) verwendet. Dieses Layout ist sehr populär und weit verbreitet (ZAMETTI 2009). Für die Gestaltung des Prototyps ist es aufgrund der Einteilung in Bereiche sehr gut geeignet. Es besteht aus insgesamt fünf Bereichen (*Regions*): Region oben, Region rechts, Region unten, Region links und Region Mitte. Entsprechend der Himmelsrichtungen werden diese Bereiche auch mit *North*, *East*, *South*, *West* und *Center* bezeichnet. Es besteht allerdings kein Zwang, alle fünf Regionen in einem Layout zu verwenden. Mit dem Border Layout wird automatisch die Trennung der Bereiche mittels Split Bars (`Ext.SplitBars`) unterstützt. Diese kann der Nutzer zudem in der Größe verändern. Auch das Ein- und Ausblenden von Regionen wird unterstützt. In der Webapplikation werden die vier Bereiche *North*, *East*, *West* und *Center* genutzt um die notwendigen Daten und Funktionalitäten zu integrieren. Das verwendete Border Layout und die verwendeten *Regions* sind in Abbildung 5-4 dargestellt.

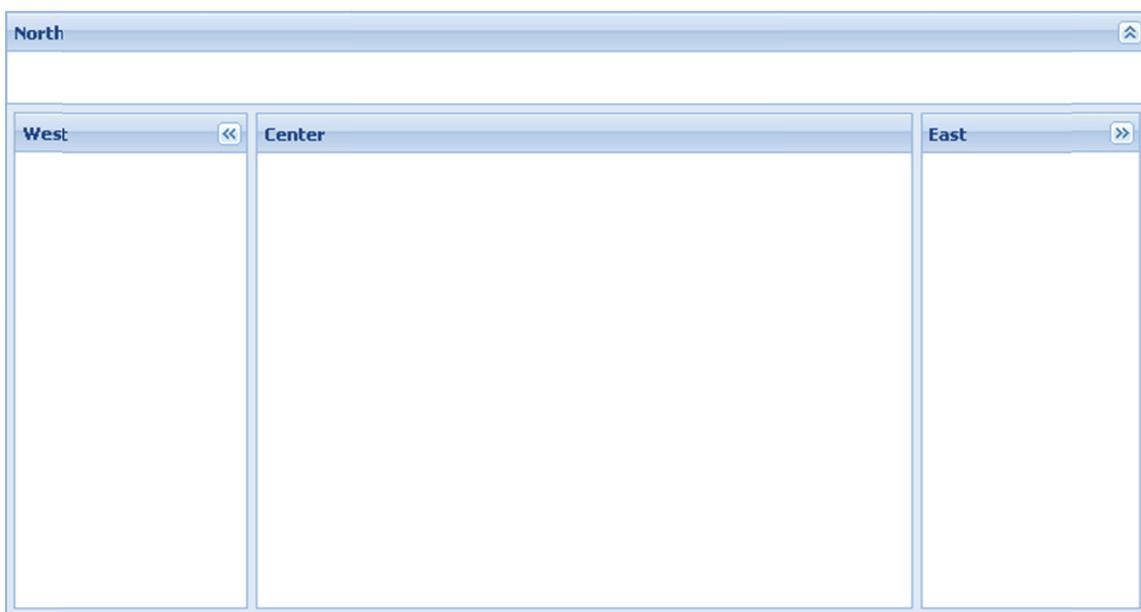


Abbildung 5-4: Grundgerüst der Webapplikation im Border Layout

Die in Abbildung 5-4 dargestellten Bereiche und deren Inhalt werden in den nachstehenden Kapiteln erläutert. Hierbei wird auch auf die Interaktion zwischen den einzelnen Bereichen eingegangen.

5.4.2 Region North

Die Region *North* stellt einen eigenständigen Container dar und beinhaltet den Titel der Webapplikation „Laser Scanning Information System“ mit dem Dow-Logo. Weiterhin ist eine Auswahlliste für die Suche von geographischen Namen aus der *GeoNames*-Datenbank implementiert. Die *GeoNames*-Datenbank beinhaltet über acht Millionen geographische Namen und Orte. Es werden dabei alle Länder abgedeckt (GEONAMES 2010). In diese Auswahlliste kann ein geographischer Name wie beispielsweise eine Stadt eingegeben werden. Im Hintergrund wird in der *GeoNames*-Datenbank nach den eingegebenen Zeichen gesucht und dabei werden automatisch Vorschläge unterbreitet. Der User kann jetzt einen der vorgeschlagenen Stadtnamen selektieren und wird zum entsprechenden Bereich auf der Karte (*Region Center*) navigiert.

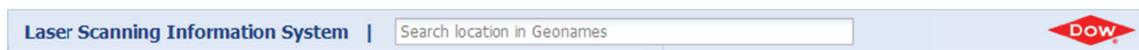


Abbildung 5-5: Nördlicher Bereich der Webapplikation mit Auswahlliste zur Suche von geographischen Namen

5.4.3 Region West

In diesen Bereich ist ein *TabPanel* (`Ext.TabPanel`) namens „Geodata“ eingebettet. Dieses besteht aus zwei Tabs (Reitern): der Tab *Projects* dient dem strukturierten Zugriff auf Laserscanning-Projekte und den entsprechenden Standpunkten des Laserscanners; der Tab *Maps* listet alle Layer (`OpenLayers.Layer`) wie beispielsweise die Straßenkarte von Google Maps oder georeferenzierte Karten (Bilder) auf (siehe Kapitel 4.2.9 und 5.1.2). In den folgenden beiden Kapiteln werden die beiden Tabs näher erläutert.

5.4.3.1 Projects-Tab

Der standardmäßig aktive Tab ist der Tab *Projects*. Er ermöglicht das Darstellen der Laserscanner-Standpunkte in der *Region Center* und die Anzeige von Sach- bzw. Metadaten in der *Region East* des Viewports. Realisiert wurde der Zugriff auf die Laserscanner-Standpunkte mithilfe einer Baumstruktur durch die Klasse `Ext.tree.TreePanel` (siehe Abbildung 5-6). Die Baumstruktur bestehend aus dem Wurzelement *Laser Scanning Projects*, dem Kontinent (Entität *Continent*), dem Werk (Entität *Site*), dem Laserscanning-Projekt (Entität *Project*) und dem Bereich, in dem mehrere Laserscanner-Standpunkte gruppiert sind (Entität *Area*). Unterhalb der *Area* befinden sich alle Laserscanner-Standpunkte (Entität *Laser Scanner Position*) mit einem

Link zum entsprechenden Viewer – beim Prototypen TruView. Die verschiedenen Hierarchiestufen des Baumes sind mit kleinen Symbolen bereichert um deren Bedeutung zu veranschaulichen (siehe auch Kapitel 5.4.3.3).

Das Wurzelement *Laser Scanning Projects* wird während der Initialisierung der Webapplikation aufgeklappt, so dass alle Kontinente sichtbar sind. Dieses Element wurde direkt in ExtJS mit der Klasse `Ext.tree.AsyncTreeNode` erstellt. Alle weiteren Elemente werden auf Anfrage des Clients mittels AJAX aus der MySQL-Datenbank abgefragt. Der Zugriff auf die Knoten erfolgt analog der Steuerung beispielsweise des Windows Explorers: ein einfacher Klick auf das kleine Dreieck, links eines Knotens bzw. ein Doppelklick auf den Knotennamen klappt den Knoten auf. Mit einem einfachen Klick auf den Knotennamen hingegen werden Sach- und Metadaten in der Region *East* angezeigt. Dies gilt allerdings nur für Knoten der Kategorien *Project*, *Area* und *Laser Scanner Position*. Bei den anderen Knoten werden keine Sachdaten angezeigt. Hier erscheint lediglich ein Hinweis, dass nur Sachdaten bei den drei oben genannten Kategorien erscheinen.

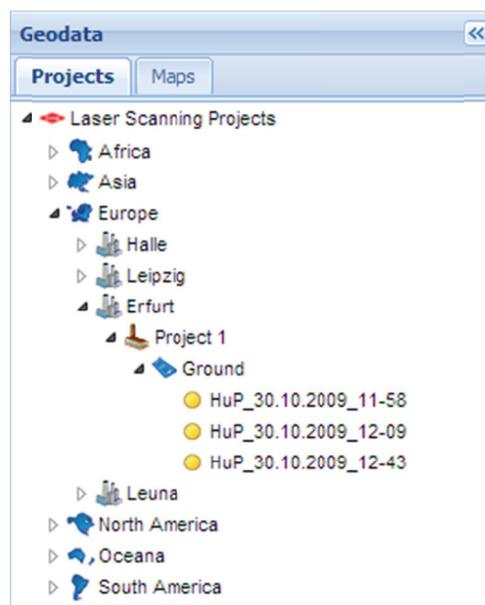


Abbildung 5-6: Baumstruktur für den Zugriff auf die Laserscanning-Daten

Die Auswahl eines Laserscanner-Standpunkts in der Baumstruktur bewirkt das Hervorheben des entsprechenden Laserscanner-Standpunkts in der Karte (*Region Center*) in blauer Farbe. An dieser Stelle existiert in GeoExt noch keine Klasse, die das Hervorheben automatisch ermöglicht. Lediglich für Daten, die in Form eines Tabellenformats (Grid) angezeigt werden, existiert ein solcher Automatismus, der durch die Klasse `GeoExt.grid.FeatureSelectionMode` realisiert ist. Um das Fehlen einer Klasse für

Daten aus Baumstrukturen auszugleichen, wurde unter Verwendung der Klassen `OpenLayers.Style` und `OpenLayers.Rule` ein eigenes Feature Selection-Model entwickelt. An dieser Stelle wird auf die Notwendigkeit eines Feature Selection-Models im Zusammenhang mit Bäumen hingewiesen; sofern dies vorhanden ist, wird viel Entwicklungsaufwand gespart.

5.4.3.2 Maps-Tab

Im Tab *Maps* werden alle Layer (`OpenLayers.Layer`), wie schon bei dem Tab *Project*, anhand einer Baumstruktur verwaltet (siehe Abbildung 5-7). Allerdings wird zum Erzeugen der Baumstruktur an dieser Stelle nicht die beim Tab *Project* verwendete Klasse `Ext.tree.TreePanel` verwendet. Stattdessen bietet GeoExt für die Verwaltung von Layern innerhalb einer Baumstruktur zwei Klassen: `GeoExt.BaseLayerContainer` und `GeoExt.OverlayLayerContainer`. Mit der Klasse `GeoExt.BaseLayerContainer` kann ein Container erstellt werden, mit dem alle Basislayer einer Karte (`OpenLayers.Map`) gesammelt werden. Basislayer sind die Layer der Kartendienste Google Maps und Bing Maps sowie der Layer *Blumarble* der NASA. Zwischen den verschiedenen Basislayern kann mittels Radiobutton hin und hergeschaltet werden. Somit kann immer nur einer dieser Layer ausgewählt und in der Region *Center* dargestellt werden. Standardmäßig ist der Layer *Blumarble* in der Karte sichtbar. Basislayer werden immer unterhalb der Overlays gerendert (siehe Kapitel 4.2.9).

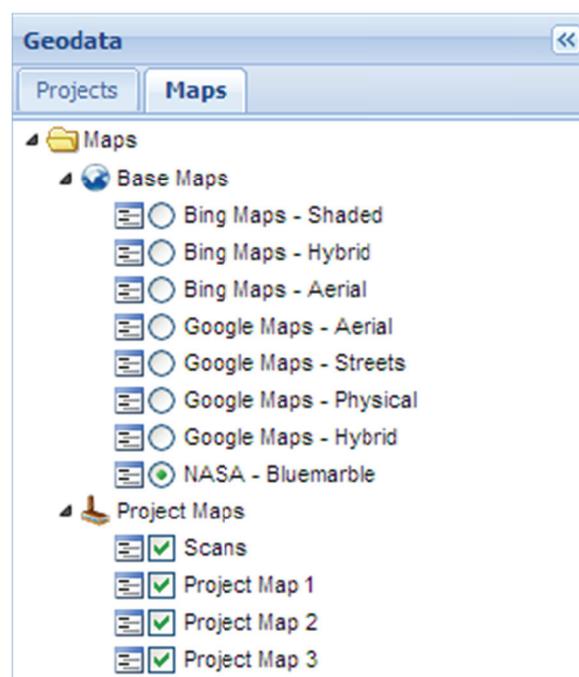


Abbildung 5-7: Baumstruktur für die Projekt-bezogenen Layer

Im Gegensatz zur Klasse `GeoExt.BaseLayerContainer` können bei der Klasse `GeoExt.OverlayLayerContainer` auch mehrere Layer parallel ausgewählt werden. Der `GeoExt.OverlayLayerContainer` ist außerdem ein Container mit dem alle Overlays einer Karte gesammelt werden. Ein besonderes Overlay ist der Vektor-Layer *Scans* (`OpenLayers.Layer.Vector`), der alle Laserscanner-Standpunkte der momentan vom User ausgewählten *Area* beinhaltet. Wenn die Webapplikation initialisiert wird, beinhaltet dieser Layer keine Daten. Erst wenn eine *Area* aufgeklappt wurde bzw. angeklickt wurde, werden die Laserscanner-Standpunkte in der Region *Center* visualisiert. Wenn ein Overlay für ein Projekt vorhanden ist, wird dieses automatisch geladen, wenn versucht wird die Laserscanner-Standpunkte darzustellen.

5.4.3.3 Symbolik

Standardmäßig beinhaltet ExtJS Symbole für die Darstellung von Knoten in einer Baumstruktur. Beispielsweise wird bei einem Knoten, der weitere Knoten beinhaltet das klassische Ordnersymbol verwendet. Diese Symbolik kann für die Webapplikation angewendet werden, birgt allerdings einige Nachteile in sich. So ist beispielsweise die Bedeutung der jeweiligen Knotenhierarchie anhand der Symbolik nicht erkennbar, da jedes Symbol gleich aussieht. Der User erkennt also nicht ohne weiteres ob er gerade ein *Project* oder eine *Site* angeklickt hat. Charakteristische Symbole können hier Abhilfe schaffen und erleichtern zudem den Umgang mit der Webapplikation. Aus diesem Grund wurden eigene Symbole für die beiden Baumstrukturen in den Tabs *Projects* und *Maps* verwendet (siehe Tabelle 5-3).

Tabelle 5-3: Verwendete Symbole in den Baumstrukturen

Kategorie	Symbol
Continent	
Site	
Project, Project Maps	
Area	
Base Maps	

Die Symbole erleichtern somit das Navigieren und ermöglichen ein intuitives Arbeiten mit der Webapplikation. Die Einbindung der Symbole in die Webapplikation erfolgt

mittels der zentral definierten Stylesheets-Datei (siehe Kapitel 5.2). Der Name für die entsprechende Stylesheet-Klasse wird aus der Datenbank abgefragt.

5.4.4 Region Center

Im zentralen Bereich des Viewports, der als *Map* bezeichnet wird, befindet sich der Container für die Karte: das sogenannte Map-Panel (`GeoExt.MapPanel`). Dieser Container dient der Darstellung des jeweiligen Basislayers und der Overlays mithilfe der beiden Baumstrukturen der Region *West*. Der Container beinhaltet im linken Teil einen *ZoomSlider* (`GeoExt.ZoomSlider`), mit dem es möglich ist die Zoomstufe der Karte zu variieren (siehe Abbildung 5-8, links). Die Anzahl der Zoomstufen ist abhängig von den vordefinierten Parametern eines jeden Layers. Im unteren rechten Teil werden die Koordinaten der Mausposition im Bezugssystem WGS 84 angezeigt (siehe Abbildung 5-8, unten rechts).

Sofern Laserscanner-Standpunkte im Kartenfenster dargestellt sind, können diese auch im Kartenfenster ausgewählt werden. Sodann werden in der Region *East* Informationen zum selektierten Standpunkt angezeigt. Es handelt sich hierbei um die gleichen Informationen wie bei der Auswahl eines Standpunktes mithilfe der Baumstruktur. Ein selektierter Laserscanner-Standpunkt (blau markiert) ist in Abbildung 5-8 zu sehen.



Abbildung 5-8: Kartenfenster mit Laserscanner-Standpunkten und einem Basislayer von Google Maps

5.4.5 Region East

Die Region *East* (*Scan Details / Miscellaneous*) dient vorrangig der Anzeige von Informationen bezüglich Laserscanning-Projekten (*Projects*), Bereichen (*Areas*) und Laserscanner-Standpunkten (*Laser Scanner Positions*). Sofern beispielsweise der User im Navigationsbereich ein Projekt auswählt, werden in der Region *East* alle zugehörigen Projekt-Informationen wie Bezeichnung des Projekts, Projektbeginn und Angaben zur Projektdatenbank angezeigt (siehe Abbildung 5-9). Sofern Email-Adressen aufgeführt werden, sind diese als Verlinkung dargestellt um die Anfrage zusätzlicher Informationen zu erleichtern. Diese Daten werden aus der MySQL-Datenbank abgefragt und müssen innerhalb der Webapplikation mittels Cascading Style Sheets formatiert werden. Hierfür wird die Klasse *Ext.Template* verwendet. Mit dieser Klasse können HTML-Fragmente definiert und zu einem späteren Zeitpunkt mit Daten beispielsweise aus einer Datenbank gefüllt werden (GARCIA 2010). Die Templates werden während der Initialisierung der Webapplikation kompiliert, um eine höhere Performance zur Laufzeit der Webapplikation zu erreichen. Sofern das Datenmodell verändert wird, müssen die entsprechenden Templates ebenfalls angepasst werden. Welche Daten in der Region *East* angezeigt werden, wird durch die Templates beeinflusst.

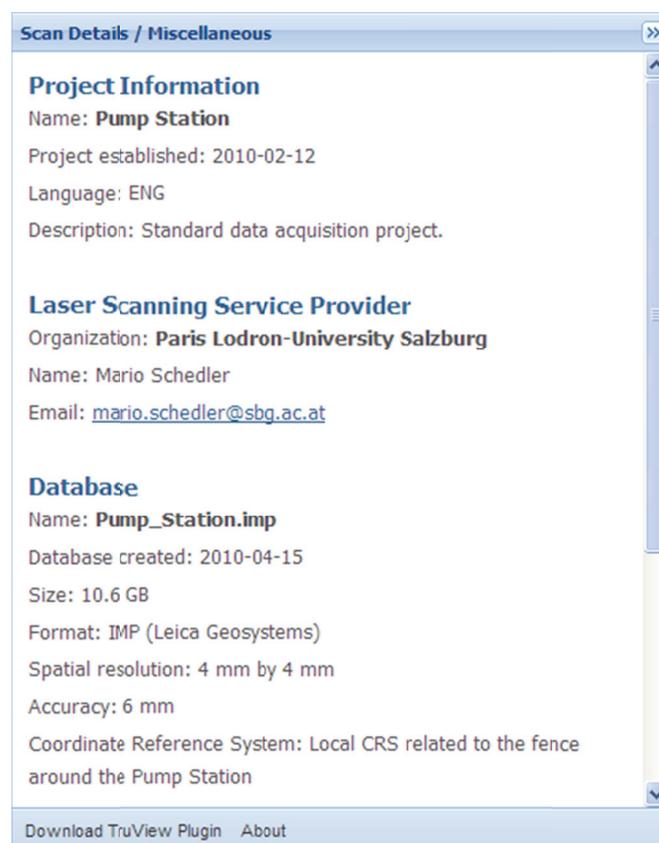


Abbildung 5-9: Daten eines Laserscanning-Projekts in Region East

In der Region *East* kann nach Auswahl eines Laserscanner-Standpunkts auch auf TruView-Daten zugegriffen werden. Sofern ein Laserscanner-Standpunkt ausgewählt wurde, erscheint in der Region East ein kleines Viewer-Icon (siehe Abbildung 5-11). Ein Klick auf dieses Icon und es öffnet sich das entsprechende TruView mit den Laserscanning-Daten des gewählten Standpunkts. Die Sachdaten der *Areas* und *Laser Scanning Positions* sind hinsichtlich ihres Layouts analog dem Laserscanning-Projekt aufgebaut (siehe Abbildung 5-11).

Die Region *East* beinhaltet zusätzlich noch eine Toolbar im unteren Bereich, in der sich zwei Buttons befinden (siehe Abbildung 5-9). Der Button *Download TruView Plugin* dient dem Herunterladen einer ZIP-Datei, die entpackt werden muss, um schließlich das TruView Plug-In zu installieren. Somit wird dem User die ggf. sehr mühselige Suche nach diesem Plug-In erspart, da es gleich in der Webapplikation integriert ist. Mit einem Klick auf den Button *About* öffnet sich ein Fenster, das den Nutzer über die Webapplikation informiert. Hier wird beispielsweise die Versionsnummer angegeben (siehe Abbildung 5-10).

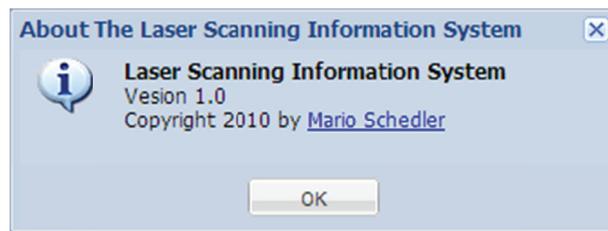


Abbildung 5-10: Hinweisfenster der Webapplikation

5.5 Der Prototyp

Wie bereits in Kapitel 5.4.1 erwähnt wurde, werden alle vier Regionen durch ein `Ext.layout.BorderLayout` in einem `Ext.Viewport` zusammengeführt. Dieses Viewport wird in das `body`-Tag der Webseite gerendert. Der `body`-Bereich in der Webseite beinhaltet neben einem `div`-Container für die Realisierung der Baumstruktur keine weiteren HTML-Tags. Im Anschluss an die Entwicklung der Webapplikation wurden die verwendeten JavaScript-Bibliotheken optimiert: die während der Entwicklung verwendeten JavaScript-Frameworks beinhalteten alle Klassen, auch diejenigen, die nicht verwendet wurden. Um nun die Performance der Webapplikation zu erhöhen und unnötigen Ballast zu entfernen, wurden individuelle JavaScript-Frameworks generiert. Hierfür stehen sogenannte Builder zur Verfügung. Beispielsweise bietet GeoExt den

GeoExt-Builder⁷, womit eine individuelle JavaScript-Datei erzeugt wurde. Diese Datei wurde anschließend im Header der Datei *tls.php* eingetragen. Analog wurde mit ExtJS und OpenLayers verfahren. Im Anschluss daran wurde die Webapplikation mit dem Internet Explorer in der Version 6, 7 und 8 sowie mit Firefox in der Version 3 und 4 Beta getestet. In allen Tests hat die Webapplikation funktioniert. Die finale Webapplikation *Laser Scanning Information System* ist in Abbildung 5-11 dargestellt.

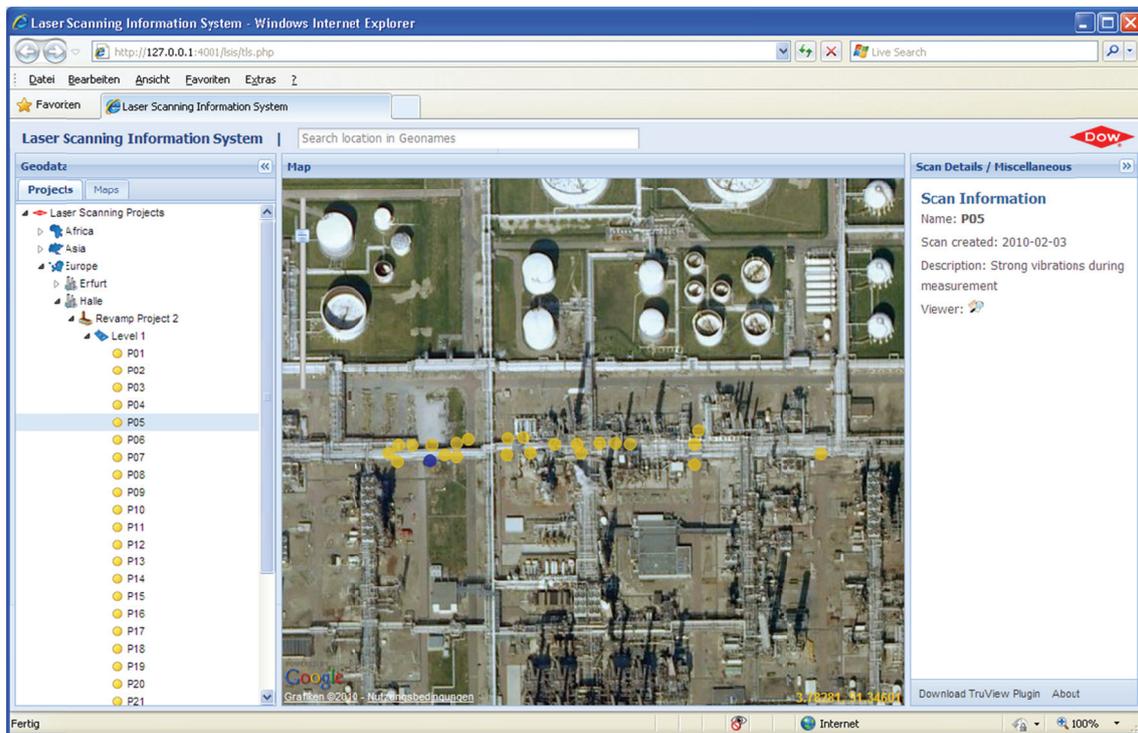


Abbildung 5-11: Prototyp des Laser Scanning Information Systems im Internet Explorer

Die Pflege der Daten erfolgt browserbasiert mithilfe der Software *phpMyAdmin*, die eine in PHP entwickelte Applikation zur Administration eben solcher Datenbanken ist. Die Administration in *phpMyAdmin* erfolgt dabei mittels einer grafischen Oberfläche, die auch ohne Kenntnisse der SQL möglich ist.

Der Prototyp wurde in die Software Server2Go eingebunden und mit dieser zusammen auf einen USB-Stick kopiert. Nachdem wenige Einstellungen vorgenommen wurden, konnte Server2Go gestartet werden und die Webapplikation wurde im Internet Explorer Explorer geöffnet. Diese mobile Variante der Webapplikation funktioniert analog zur „Klassischen“ sowohl im Internet Explorer als auch im Firefox. Die Laserscanning-Daten können somit auch mobil genutzt werden.

⁷ <http://geoext.org/builder/v0.7.html>

6 Abschließende Betrachtungen

6.1 Zusammenfassung

Im Rahmen dieser Masterthesis wurde eine Webapplikation zur Dokumentation von Laserscanning-Projekten und Visualisierung von terrestrischen Laserscanning-Daten entwickelt und prototypisch bei der Dow Chemical Company implementiert. Mit dieser Webapplikation wird dem Nutzer ein hilfreiches Werkzeug zur Verfügung gestellt, mit dem webbasiert nach Laserscanning-Daten gesucht, auf diese zugegriffen werden kann und mit dem diese auch visualisiert werden können.

Durch die Auseinandersetzung mit den Themen Web-Mapping und den verschiedenen Möglichkeiten Webapplikationen zu erstellen; terrestrischem Laserscanning und bestehenden webbasierten Applikationen sowie Metadaten und Metadaten-Standards konnte ein Überblick zum gegenwärtigen Stand der Technologie geschaffen werden. Neben diesen theoretischen und technischen Grundlagen wurde dargestellt, wer die zukünftigen Nutzer der Webapplikation sind und welche Anforderungen sie an das System stellen. Darüber hinaus wurden Anwendungsfälle (*Use Cases*) konstruiert, die aufzeigen, wie der Nutzer mit der Webapplikation interagieren wird.

Das Wissen um die theoretischen und technischen Grundlagen sowie der Anforderungen an die Webapplikation, lieferte die Basis um zielgerichtet mit der Entwicklung des Prototyps zu beginnen. Am Anfang der Entwicklungsphase stand die Datenmodellierung, welche im weiteren Verlauf zu einem konzeptionellen Datenmodell führte, das schließlich in einer MySQL-Datenbank realisiert wurde. Die Webapplikation an sich wurde mit den JavaScript-Frameworks Ext JS, OpenLayers und GeoExt entwickelt, wodurch eine hohe Usability erreicht werden konnte.

Mit dem entwickelten Prototyp können somit Laserscanning-Daten gesucht, Informationen über Laserscanner-Standpunkte bzw. -Projekte abgerufen und letztlich auch Laserscanning-Daten mithilfe des Viewers TruView visualisiert werden.

6.2 Ergebnisse

In diesem Abschnitt werden die Ziele der Masterthesis, die in Kapitel 1.2 festgelegt sind, noch einmal aufgegriffen und überprüft inwieweit sie erreicht wurden. Die beiden Hauptziele der Masterthesis waren erstens die Entwicklung einer benutzerfreundlichen Webapplikation zur Dokumentation von Laserscanning-Projekten und Visualisierung von terrestrischen Laserscanning-Daten sowie zweitens deren Umsetzung bei der Dow Chemical Company. Diese beiden Ziele wurden – wie bereits in der Zusammenfassung deutlich wird – erreicht. Des Weiteren sollten die Nachteile bestehender Lösungen in der Webapplikation beseitigt werden. Im Folgenden werden daher die aus dieser Zielstellung resultierenden Entwicklungsfortschritte näher dargestellt.

Ein gewichtiger Punkt während der Entwicklung der Webapplikation war die Steigerung der Usability gegenüber den bestehenden Systemen. Die Webapplikation soll intuitiv bedienbar sein und ohne spezielle Fertigkeiten im Bereich des Laserscannings genutzt werden können. Durch die Verwendung moderner JavaScript-Frameworks und Webservices wie Google Maps mit denen viele Anwender vertraut sind, konnte die Usability im Vergleich mit bestehenden Systemen deutlich verbessert werden.

Ein vorbestehendes Problem war die unübersichtliche Darstellung von großflächig verteilten Standpunkten. Mit der entwickelten Webapplikation können nun auch größere Bereiche mit heterogen verteilten Standpunkten abgebildet werden, da flächendeckend Basiskarten und eine Zoomfunktion vorhanden sind. Somit kann die Webapplikation auch für die großräumige bzw. globale Verwaltung von Laserscanning-Daten verwendet werden. Um darüber hinaus die Benutzung der Karte übersichtlich zu gestalten, wurde die Sichtweise auf die Kartendaten durch die Draufsicht beschränkt. Die Darstellung der Daten in einer isometrischen Ansicht ist nicht möglich, was die räumliche Einordnung von Laserscanner-Standpunkten vereinfacht. Die Unterteilung in *Areas* ist an dieser Stelle zielführender und für den Nutzer verständlicher.

Die Überlagerung von Laserscanner-Standpunkten ist in der Webapplikation nur noch bedingt möglich. So werden die Laserscanner-Standpunkte in *Areas* gruppiert. Bei der entsprechenden Wahl der Zoomstufe überlagern sich die Symbole nicht und eine Selektion der Standpunkte ist ohne weiteres möglich. Die Wahl von punkthaften Symbolen anstelle von Dreiecken, wie beispielsweise bei TruView, begünstigt dies ebenfalls, da ein Symbol weniger Fläche einnimmt. Sofern man aber hinauszoomt, kommt es auch hier zu Überschneidungen von Laserscanner-Standpunkten.

Die Verwendung von Webservices wie Google Maps liefert Basisdaten, auf denen die Overlays wie beispielsweise die Laserscanner-Standpunkte in vielen Bereichen sehr gut dargestellt werden können. Durch diese Trennung der Basislayer von den Overlays wird die Datenverwaltung erleichtert und zudem können mehrerer Laserscanning-Projekte in einer Applikation verwaltet werden. Wenn die Datenqualität der Webservices nicht ausreichend ist, können projektbezogene Layer in die Webapplikation hinzugefügt werden. Die Webapplikation ist somit skalierbar und im Gegensatz zu verweissensitiven Karten nicht statischer Natur.

Zusammenfassend kann gesagt werden, dass das Ziel dieser Masterthesis, nämlich die Entwicklung einer Webapplikation zur Dokumentation von Laserscanning-Projekten und Visualisierung von terrestrischen Laserscanning-Daten, erreicht wurde. Auch ein Prototyp wurde entwickelt und bei der Dow Chemical Company implementiert. Die angesprochenen Kritikpunkte bisheriger Lösungen wurden eliminiert bzw. werden im Ausblick dazu Lösungen aufgezeigt.

6.3 Ausblick

Der in dieser Masterthesis entwickelte Prototyp ist voll funktionsfähig und bietet eine grundlegende Funktionalität, die der Nutzer benötigt um sich über Laserscanning-Projekte und -Daten zu informieren. Natürlich könnten weitere und durchaus nützliche Funktionalitäten in der Webapplikation implementiert werden, dies würde jedoch hinsichtlich des Entwicklungsaufwandes über den zeitlichen Rahmen einer Masterthesis hinausgehen. Die entwickelte Webapplikation wird sich im Laufe der Zeit verändern. An dieser Stelle soll deshalb ein Ausblick gegeben werden, wohin die Weiterentwicklung der Webapplikation führen kann.

Momentan werden Daten mithilfe der Software phpMyAdmin in die Webapplikation eingepflegt, was von verschiedenen Nutzern aufgrund von komplexen Anforderungen wie der Eingabe von Primär- und Fremdschlüssel, der Indexierung etc. als unkomfortabel angesehen werden könnte. Um die Pflege und Aktualisierung der Daten zu vereinfachen und vor allem um technisch nicht versierten Personen die Datenpflege zu ermöglichen, könnte zukünftig eine eigene Grafische Benutzeroberfläche (GUI) implementiert werden. Hierbei muss überprüft werden, ob die GUI direkt in die Webapplikation integriert wird oder ob hierfür eine zusätzliche Webseite erstellt wird. In beiden Fällen

würde die Datenpflege gegenüber der Anwendung von phpMyAdmin vereinfacht werden.

Auch die Möglichkeit den Bereich des Kartenfensters zu drucken, könnte kurzfristig implementiert werden. Eine Druckfunktion ist zwar nicht zwangsweise notwendig und wurde bisher auch nicht angefragt, dennoch könnte diese hin und wieder als nützlich angesehen werden. Ähnlich verhält es sich mit einer Hilfeseite, mit der die Webapplikation kurz erläutert wird. Sofern sich im Laufe der Zeit herausstellt, dass eine Hilfeseite notwendig ist, könnte diese in der Region *East* (siehe Kapitel 5.4.5) implementiert werden. Weiterhin kann über kartographische Verfeinerungen wie beispielsweise dicht beieinanderliegender bzw. überlappender Symbole und deren Darstellung in der Karte nachgedacht werden. Die Klasse `OpenLayers.Strategy.Cluster` erlaubt es Laserscanner-Standpunkte zu einem Cluster zusammenzufassen. Gegenwärtig wird durch die Einordnung der Punkte in Bereiche (*Areas*), auf eine solche Funktion verzichtet. Wenn dies nicht mehr ausreichend ist oder vom Anwender gewünscht wird, so sollte eine Clustering der Laserscanner-Standpunkte implementiert werden.

Insgesamt konnte mit der Webapplikation ein funktionales Werkzeug entwickelt werden, welches den bisher bekannten Anforderungen der Nutzer voll entspricht. Die Applikation wurde jedoch skalierbar entwickelt, um darüber hinausgehende Anforderungen leicht implementieren zu können.

Literaturverzeichnis

- AALDERS, H. (2007): An Introduction to Metadata for Geographic Information. In MOELLERING H. (Ed.): World Spatial Metadata Standards. Amsterdam: Elsevier, 3-27.
- ASCHE, H. (2001): Kartographische Informationsverarbeitung in Datennetzen – Prinzipien, Produkte, Perspektiven. In HERRMANN, C. & ASCHE, H. (Hrsg.): Web.Mapping 1: Raumbezogene Informationen und Kommunikation im Internet. Heidelberg: Wichmann, 3-17.
- BEHR, F.-J. (2000): Strategisches GIS-Management. Grundlagen Systemeinführung und Betrieb. Heidelberg: Wichmann.
- BILL, R. (1999): Grundlagen der Geoinformationssysteme. Band 1: Hardware, Software und Daten. Heidelberg: Wichmann.
- BOEHLER, W., HEINZ, G., MARBS, A. & SIEBOLD, M. (2002): 3D Scanning Software: An Introduction. <http://www.i3mainz.fh-mainz.de/publicat/korfu/p11_Boehler.pdf> (Zugriff: 2009-12-18).
- BRINKHOFF, T. (2005): Geodatenbanksysteme in Theorie und Praxis. Einführung in objektrelationale Geodatenbanken unter besonderer Berücksichtigung von Oracle Spatial. Heidelberg: Wichmann.
- COPPLE, J. (Z+F UK LTD) (2010): LFM Server. Email (2010-04-07).
- DCMI (DUBLIN CORE METADATA INITIATIVE) (2009a): Dublin Core Metadata Initiative Home. <<http://dublincore.org/>> (Zugriff: 2009-12-22).
- DCMI (DUBLIN CORE METADATA INITIATIVE) (2009b): Dublin Core Metadata Element Set, Version 1.1. <<http://dublincore.org/documents/2008/01/14/dces/>> (Stand: 2008-01-12) (Zugriff: 2009-12-22).
- DCMI (DUBLIN CORE METADATA INITIATIVE) (2009c): Expressing Dublin Core in HTML/XHTML meta and link elements.

- <<http://dublincore.org/documents/2003/11/30/dcq-html/>> (Stand: 2003-11-30)
(Zugriff: 2009-12-22).
- DICKMANN, F. (2004): Einsatzmöglichkeiten neuer Informationstechnologien für die Aufbereitung und Vermittlung geographischer Informationen - das Beispiel kartengestützter Online-Systeme. Göttinger Geographische Abhandlungen, Heft 112. Göttingen: Verlag Erich Goltze GmbH & Co. KG.
- DOW (2009a): Leuna. Internes Bild. Schkopau: o.V.
- DOW (2009b): Kaverne 10. Internes Bild. Schkopau: o.V.
- DUHL, J. (2003): Rich Internet Applications. White Paper.
<http://www.gevasys.de/materialien/idc_impact_of_rias.pdf> (Stand: 2003-11-30)
(Zugriff: 2010-01-31).
- DUMKE, R., LOTHER, M., WILLE, C., & ZBROG, F. (2003). Web Engineering. München: Pearson Studium.
- EICHINGER, C. (2004): Architektur von Web-Anwendungen. In: KAPPEL, G., PRÖLL, B., REICH, S. & RETSCHIZEGGER, W. (Hrsg.): Web Engineering. Systematische Entwicklung von Web-Anwendungen. Heidelberg: dpunkt.verlag, 77-100.
- ELMASRI, R. & NAVATHE, S. (2009): Grundlagen von Datenbanksystemen. Bachelorausgabe. München [u.a.]: Pearson Studium.
- EXT JS (2010): Licensing Overview - Ext JS and Ext GWT.
<<http://www.extjs.com/products/license.php>> (Zugriff: 2010-01-31).
- FARO (2009): Laser Scanner Photon Overview. <http://laser-scanner.faro.com/tl_files/faro-laser-scanner-photon-120-20/extras/Extra%20-%20Laser%20Scanner%20Overview.pdf> (Zugriff: 2010-08-19).
- FAYAD, M., SCHMIDT, D. C. & JOHNSON, R. E. (1999). Building Application Frameworks. Object-oriented Foundations of Framework Design. New York: Wiley.
- FITZKE, J. (1999): GIS online: WebGIS-Typen. <<http://www.giub.uni-bonn.de/gistutor/internet/inetgis/423.htm>> (Stand: 1999-03-03) (Zugriff: 2010-01-10).
- FOSSGIS (2009): FreeGIS Datenbank. Software – WebGIS.
<<http://freegis.org/database/?cat=9>> (Stand: 2009-07-30) (Zugriff: 2010-01-23).
- GARCIA, J. (2010): Ext JS in Action. Greenwich: Manning Publications Co.
- GEOEXT (2009): GeoExt. Milestone 0.5. <<http://trac.geoext.org/milestone/0.5>> (Stand: 2009-06-15) (Zugriff: 2010-01-24).

- GEOEXT (2010): GeoExt. JavaScript Toolkit for Rich Web Mapping Applications. <<http://www.geoext.org/>> (Zugriff: 2010-01-24).
- GEONAMES (2010): GeoNames. <<http://www.geonames.org/>> (Zugriff: 2010-06-07).
- GRÜNBACHER, P. (2004): Requirements Engineering für Web-Anwendungen. In KAPPEL, G., PRÖLL, B., REICH, S. & RETSCHITZEGGER, W. (Hrsg.): Web Engineering. Systematische Entwicklung von Web-Anwendungen. Heidelberg: dpunkt.verlag, 29-47.
- HABERKERN, T. (2010): What is Server2Go. <<http://www.server2go-web.de/>> (Zugriff: 2010-07-23).
- HOCEVAR, A & MOULLET, C. (2009): Building a Web Mapping Application with GeoExt. <<http://dev.geoext.org/geoext.pdf>> (Stand: 2009-10-23) (Zugriff: 2009-11-18).
- INGENSAND, H. & SCHULZ, T. (2005): Terrestrisches Präzisions-Laserscanning. In: CHESI, G. & WEINOLD, T. (Hrsg.): Internationale Geodätische Woche Obergurgl 2005. Heidelberg: Wichmann, 189-192.
- ISO/TC 211 (INTERNATIONAL ORGANIZATION FOR STANDARDIZATION /TECHNICAL COMMITTEE 201) (2003): ISO 1915. Geographic Information – Metadata. Geneva: ISO.
- JSON (2010): Introducing JSON. <<http://www.json.org/>> (Zugriff: 2010-06-18).
- KORDUAN, P., & ZEHNER, M. L. (2008). Geoinformation im Internet. Technologien zur Nutzung raumbezogener Informationen im WWW. Heidelberg: Wichmann.
- KOTHURI, R., GODFRIND, A. & BEINAT, E. (2004). Pro Oracle Spatial. The essential Guide to Developing spatially enabled Business Applications. New York: Springer.
- KRAAK, M. (2001): Settings and Needs for Web Cartography. In KRAAK, M. & BROWN, A. (Eds.): Web Cartography. Development and Prospects. London & New York: Taylor & Francis, 1-7.
- KRAUS, K. (2007): Photogrammetry. Geometry from Images and Laser Scans. Berlin: De Gruyter.
- KRESSE, W., & FADAIE, K. (2004). ISO Standards for geographic Information. Berlin: Springer.
- KST. GDI-DE (KOORDINIERUNGSSTELLE GDI-DE) (2008): Deutsche Übersetzung der Metadatenfelder des ISO 19115 Geographic Information – Metadata. <

- de.org/de_neu/download/AK/ISO19115_GermanTranslation_GDIDE.pdf >
(Stand: 2008-10-08) (Zugriff: 2009-10-15).
- LEICA (2009a): TruView Sample Data. Oil and Gas Refinery. <<http://www.leica-geosystems.com/truview/Plant-2K/sitemap.htm>> (Zugriff: 2009-10-15).
- LEICA (2009b): Leica Cyclone PUBLISHER 7.0 und TruView 2.0: Kostenfrei und einfach. <http://www.leica-geosystems.com/downloads123/hds/hds/cyclone/brochures-datasheet/Leica_Cyclone_PublisherTruView_DS_de.pdf> (Zugriff: 2009-10-15).
- LONGLEY, P. A., GOODCHILD, M.F., MAGUIRE, D. J. & RHIND, D. W. (2005): Geographic Information Systems and Science. Chichester: John Wiley & Sons.
- MINTERT, S. & LEISEGANG, C. (2007): Ajax. Grundlagen, Frameworks und Praxislösungen. Heidelberg: dpunkt.
- MITCHELL, T. (2005): Web Mapping Illustrated. Sebastopol: O'Reilly.
- MYSQL (2010): MySQL 5.1 Reference Manual.
<<http://dev.mysql.com/doc/refman/5.1/en/create-table.html>> (Zugriff: 2010-06-12).
- NISO (NATIONAL INFORMATION STANDARDS ORGANIZATION) (2004): Understanding Metadata.
<<http://www.niso.org/publications/press/UnderstandingMetadata.pdf>> (Zugriff: 2010-01-05).
- NONN, U. & ZIPF, A. (2007): Metadaten für 3D-Stadtmodelle – Untersuchung der Eignung von ISO 19115 und Möglichkeiten der Erweiterung.
<<http://www.geographie.uni-bonn.de/karto/CORP07.3D-Metadaten.nonn.zipf.draft.pdf>> (Zugriff: 2009-12-30).
- OESTEREICH, B. (2001). Objektorientierte Softwareentwicklung. Analyse und Design mit der Unified Modeling Language. München: Oldenbourg.
- OGC (2010): Web Map Service – OpenGIS Web Map Service (WMS) Implementation Specification. <<http://www.opengeospatial.org/standards/wms>> (Zugriff: 2010-02-12).
- OPENLAYERS (2010a): OpenLayers: Free Maps for the Web.
<<http://www.openlayers.org/>> (Zugriff: 2010-01-17).
- OPENLAYERS (2010b): Spherical Mercator.
<http://docs.openlayers.org/library/spherical_mercator.html> (Zugriff: 2010-05-08).

- PENG, Z.-R., & TSOU, M.-H. (2003): Internet GIS. Distributed Geographic Information Services for the Internet and Wireless Networks. New York: Wiley.
- PHP (2010): What is PHP? <<http://php.net/index.php>> (Zugriff: 2010-06-18).
- SCARDINA, M. V., CHANG, B., & WANG, J. (2004): Oracle Database 10g. XML & SQL: Design, Build & Manage XML Applications in Java, C, C++ & PL/SQL. New York: McGraw-Hill.
- SCHÄFER, T. (2008): "Malen nach Zahlen" – Das Zusammenspiel von Punktwolke und CAD. In: DVW e. V. - GESELLSCHAFT FÜR GEODÄSIE, GEOINFORMATION UND LANDMANAGEMENT (Hrsg.): Terrestrisches Laserscanning (TLS 2008). Schriftenreihe des DVW, Band 54. Augsburg: Wißner-Verlag, 15-27.
- SCHÜTZE, E. (2007): Stand der Technik und Potenzial von Smart Map Browsing im Webbrowser am Beispiel der Freien WebMapping-Anwendung OpenLayers. Osnabrück, Deutschland. Diplomarbeit, Hochschule Bremen.
- SHEKHAR, S., & XIONG, H. (2008). Encyclopedia of GIS. New York: Springer.
- STEFANAKIS, E & PATROUMPAS, K. (2008): Google Earth and XML: Advanced Visualization and Publishing of Geographic Information. In PETERSON, M. (Ed.): International Perspectives on Maps and the Internet. Berlin, Heidelberg & New York: Springer, 143-152.
- STEYER, R. (2008): Ajax Frameworks. RIAs mit Dojo & Co. Open source library. München: Addison Wesley in Pearson Education Deutschland.
- STROBL, J. (1994): Grundzüge der Metadatenorganisation für GIS. <<http://www.sbg.ac.at/geo/agit/papers95/jstrobl.htm>> (Zugriff: 2010-01-03).
- UNISON ENGINEERING (2009): 3D Laserscanning Engineering Services - Dienstleistungen: Unison PlantView / TruView. <<http://www.unison-engineering.de/laserscanning.html>> (Zugriff: 2009-10-31).
- VAN TULDER, G. (2003): Storing Hierarchical Data in a Database. <<http://articles.sitepoint.com/article/hierarchical-data-database>> (Zugriff: 2010-03-24).
- W3C (WORLD WIDE WEB CONSORTIUM) (2009): Extensible Markup Language (XML). <<http://www.w3.org/XML/>> (Stand: 2009-04-16) (Zugriff: 2010-02-07).
- W3C (WORLD WIDE WEB CONSORTIUM) (2009): Resource Description Framework (RDF) / W3C Semantic Web Activity <<http://www.w3.org/rdf/>> (Zugriff: 2009-12-27).

- WENZ, C. (2007): JavaScript und AJAX. Das umfassende Handbuch; [Einführung, Praxis, Referenz; browserübergreifende Lösungen; Web 2.0: DOM, CSS, XML, Web Services]. Bonn: Galileo-Press.
- WILL, M. (Google Ireland Ltd.) (2010): Google Maps Premier API für Interne Anwendung. Email (2010-08-19).
- WIKIPEDIA (2009): Dublin Core. <http://de.wikipedia.org/wiki/Dublin_Core> (Zugriff: 2009-10-31).
- Z+F (2009): LFM Viewer/ViewerLite. <http://www.zf-uk.com/lfm/pdf/viewer_ppm.pdf> (Zugriff: 2009-11-02).
- Z+F (2010a): LFM NetView 1.1: Product Release. <http://www.zf-uk.com/newsletters/LFMNetView_1.1.html> (Zugriff: 2009-11-02).
- Z+F (2010b): LFM NetView V1.1.0.0. Quick Start Guide. <https://netview.zf-uk.com/EN/Content/Help/LFM_NetView_QSG_v1.1.0.0.pdf> (Zugriff: 2010-01-02).
- Z+F (2010c): LFM NetView. <http://www.zf-laser.com/LFM_netview_D_neu.pdf> (Zugriff: 2010-01-03).
- Z+F (2010d): LFM NetView Demo Projects. Forensic. <<https://netview.zf-uk.com/EN/Content/lfmNetView.html?projectToLoad=FOR>> (Zugriff: 2010-01-04).
- Z+F (2010e): LFM Server. <http://www.zf-laser.com/lfm_server_E_new.PDF> (Zugriff: 2010-01-04).
- ZAMMETTI, F. W. (2009): Practical Ext JS projects with Gears. Berkeley, CA: Apress.

Anhang

Im Anhang befindet sich eine DVD auf der alle relevanten Daten, die für den Betrieb der Webapplikation notwendig sind, gespeichert sind. Auch der Prototyp der Webapplikation – in Form der mobilen Variante – befindet sich auf dieser DVD.