



**Konzeption und Implementierung eines multimodalen
Campusroutenplaners am Beispiel der Universität Potsdam
– Möglichkeiten und Grenzen der Nutzung von Open Source
Software und freien Daten**

Zur Erlangung des Grades

Master of Science

im Studiengang Geoinformation und Visualisierung

an der Fakultät für Mathematik und Naturwissenschaften

der Universität Potsdam

vorgelegt von

Christian Kuntzsch

Betreuer:

Dr. Anne Tauch
Jörg Becker

Eingereicht am:

17.04.2015

Dieses Werk ist unter einem Creative Commons Lizenzvertrag lizenziert:
Namensnennung – Keine kommerzielle Nutzung – Weitergabe unter gleichen
Bedingungen 4.0 International
Um die Bedingungen der Lizenz einzusehen, folgen Sie bitte dem Hyperlink:
<http://creativecommons.org/licenses/by-nc-sa/4.0/>

Online veröffentlicht auf dem
Publikationsserver der Universität Potsdam:
URN [urn:nbn:de:kobv:517-opus4-77467](http://nbn-resolving.org/urn:nbn:de:kobv:517-opus4-77467)
<http://nbn-resolving.org/urn:nbn:de:kobv:517-opus4-77467>

EIGENSTÄDNIGKEITSERKLÄRUNG

Ich erkläre hiermit, dass ich die vorliegende Thesis selbständig und ohne unzulässige fremde Hilfe angefertigt habe. Die verwendeten Quellen sind vollständig zitiert.

Datum: 17.04.2015 **Unterschrift** _____

Christian Kuntzsch



Dieses Werk ist lizenziert unter einem Creative Commons Lizenzvertrag:

Namensnennung – Keine kommerzielle Nutzung – Weitergabe unter gleichen Bedingungen 3.0
Deutschland

Die Bedingungen der Lizenz sind unter folgendem Link einzusehen:

<http://creativecommons.org/licenses/by-nc-sa/3.0/de/>

DANKSAGUNG

Besonderer Dank gilt meinen Betreuern, Dr. Anne Tauch, Jörg Becker und Alexander Pilz, die mich während der Bearbeitungszeit dieser Arbeit unterstützt haben und mit Rat und Tat zur Seite standen – egal wie absurd die Fragen auch waren, die ich gestellt habe.

Für ihre fantastischen Ergebnisse im Studienprojekt, worauf diese Arbeit aufbaut, danke ich Marcel Hibbe und Mathias Robeck.

Danken möchte ich Dominik Brauner, Frank Krause, Sarah Paatsch und Wolfram Pönisch, die mir bei der Fertigstellung der Arbeit geholfen haben.

Schließlich danke ich meiner Familie, besonders meinen Eltern Dagmar und Andreas, für ihre schier unendliche Geduld und bedingungslose Unterstützung während meines gesamten Studiums.

KURZFASSUNG

Konzeption und Implementierung eines multimodalen Campusroutenplaners am Beispiel der Universität Potsdam – Möglichkeiten und Grenzen der Nutzung von Open Source Software und freien Daten

Von Christian Kuntzsch

Die regelmäßige Navigation durch den Raum gehört für Studenten der Universität Potsdam zum Alltag. Man möchte, unabhängig vom Fortbewegungsmittel, schnell und sicher von zu Hause zum Hörsaal oder Seminargebäude. Eine umfassende Navigationshilfe, die alle Transportmodi verbindet, wird dafür verlangt.

Das Ziel dieser Arbeit besteht darin, ein Konzept für einen multimodalen Routenplaner zu entwickeln, der es Studenten und Gästen der Universität Potsdam ermöglicht, sich zwischen den dezentral gelegenen Campusstandorten zu bewegen – egal ob mit Bus und Bahn, dem Auto, Fahrrad oder zu Fuß. Die Implementierung erfolgt ausschließlich auf Grundlage freier Daten und freier, quelloffener Software (FOSS), die für diesen Zweck aufbereitet werden. Ergebnis ist eine webbasierte Applikation, die über eine Entwicklerschnittstelle (API) in andere Projekte eingebunden werden kann.

ABSTRACT

Conceptual design and implementation of a multimodal trip planner using the example of the University of Potsdam - Possibilities and limitations using Open Source software and free data

By Christian Kuntzsch

For Potsdam University students, navigation through space on a regular basis belongs to everyday life. There is a need for getting from one place to another as fast and as safe as possible, regardless of the means of transportation. A comprehensive navigational aid that connects any mode of transport is required.

The aim of this thesis is to develop a concept for a multimodal trip planner, allowing students and guests of the University of Potsdam, to move between the decentralized campus locations – whether by bus and train, car, bike or on foot. The implementation is done solely based on free data and free open source software (FOSS) which are processed for this purpose. The result is a web-based application, integrated with a developer interface (API) to use in other projects

INHALTSVERZEICHNIS

DANKSAGUNG	V
KURZFASSUNG	VII
ABSTRACT	VIII
ABKÜRZUNGSVERZEICHNIS	XI
ABBILDUNGSVERZEICHNIS	XII
TABELLENVERZEICHNIS	XIII
QUELLCODEVERZEICHNIS	XIV
1 EINLEITUNG	1
1.1 Motivation.....	1
1.2 Problemstellung	3
1.3 Struktur.....	4
2 VERWANDTE ARBEITEN	6
2.1 Campusinformations- und Navigationssysteme	6
2.2 Multimodale Routenplaner	8
3 KONZEPTION	10
3.1 Funktionsumfang	10
3.1.1 Start und Ziel festlegen.....	11
3.1.2 Routingfähiges Netzwerk	12
3.1.3 Routingalgorithmus	13
3.1.4 Konfigurationsmöglichkeiten durch den Nutzer	15
3.1.5 Präsentation der Routingergebnisse.....	18
3.2 Zusammenfassung.....	21
4 DATENGRUNDLAGE	23
4.1 OpenStreetMap.....	23
4.2 General Transit Feed Specification	26
4.3 OpenTripPlanner	30
4.3.1 Routing und Algorithmen des OTP	33
4.3.2 Routing basierend auf OSM-Tags	34
5 UMSETZUNG	37
5.1 Verwendete Software	37
5.1.1 Struktur OpenTripPlanner	38
5.2 Einrichtung der Arbeitsumgebung	39
5.3 Aufbereitung der Daten	41
5.3.1 OpenStreetMap	41
5.3.2 Geocoder	43
5.3.3 Digitales Geländemodell.....	45
5.4 Routingfähiges Netzwerk erzeugen	47
5.5 Serverseitige Implementierung	48
6 LIZENZEN	53
6.1 OpenStreetMap.....	53
6.2 General Transit Feed Specification	55
6.3 OpenTripPlanner	56
6.4 Auswirkungen auf das Projekt.....	56

7	ERGEBNISSE	59
7.1	Bewertung	59
7.2	Ausblick	64
7.3	Zusammenfassung.....	68
8	LITERATURVERZEICHNIS	71
9	ANHANG	79
9.1	Anhang A – application-context.xml.....	79
9.2	Anhang B – graph-config.xml	85

ABKÜRZUNGSVERZEICHNIS

API	Application Programming Interface
CSS	Cascading Style Sheets
DGM	Digitales Geländemodell
ETRS89	Europäisches Terrestrische Referenzsystem 1989
(F)OSS	(Free) Open Source Software
HTML	Hypertext Markup Language
GDAL	Geospatial Data Abstraction Library
GTFS	General Transit Feed Specification
GTFS-RT	General Transit Feed Specification Real Time
IDE	Integrierte Entwicklungsumgebung
(L)GPL	(Lesser) General Public License
MIV	Motorisierter Individualverkehr
ÖPNV	Öffentlicher Personennahverkehr
OSM	OpenStreetMap
OTP	OpenTripPlanner
POI	Point Of Interest
REST	Representational State Transfer
URL	Uniform Resource Locator
VBB	Verkehrsverbund Berlin-Brandenburg
WGS 84	World Geodetic System 1984
XML	Extensible Markup Language

ABBILDUNGSVERZEICHNIS

Abbildung 1 - Statischer Lageplan des Campus Golm angeboten auf der Homepage der Universität Potsdam (BRANDT, 2013).....	2
Abbildung 2 - Campusinformations- und Navigationssysteme anderer Hochschulen (l.o.: MARTIN-LUTHER-UNIVERSITÄT HALLE-WITTENBERG 2010; r.o.:TECHNISCHE UNIVERSITÄT DRESDEN 2012; l.u.:UNIVERSITÄT HAMBURG 2013; r.u.:BISCHOFF 2011)	7
Abbildung 3 - Graph nach BLEEKE (2006)	13
Abbildung 4 - Zusammenhängender Graph nach BLEEKE (2006)	14
Abbildung 5 - Nicht zusammenhängender Graph nach BLEEKE (2006)	14
Abbildung 6 - Gewichteter Graph nach BLEEKE (2006)	15
Abbildung 7 - Gerichteter Graph nach BLEEKE (2006).....	15
Abbildung 8 - Schematische Übersicht eines webbasierten Routenplaners	19
Abbildung 9 - Konzept für den Funktionsablauf eines multimodalen Routenplaners	22
Abbildung 10 - Daten- und Informationsstruktur des Verkehrsverbunds Berlin-Brandenburg nach VBB VERKEHRSVERBUND BERLIN-BRANDENBURG GMBH (2014)	29
Abbildung 11 - Konfigurationsdreieck für Fahrradfahrer in verschiedenen Einstellungen.....	32
Abbildung 12 - Erzeugung einer Abkürzungskante (nach Geisberger 2008)	34
Abbildung 13 - Anteile der Programmiersprachen von OTP	38
Abbildung 14 - Übersicht aller OpenTripPlanner Module in Eclipse	40
Abbildung 15 - Unbearbeitete (oben) und nachbearbeitete (unten) OSM-Daten am Beispiel Haus 24, Institut für Geographie, Golm.....	43
Abbildung 16 - Ein Graph überlagert das DGM (BYRD, 2011).....	46
Abbildung 17 - High-Level-Architektur von OpenTripPlanner nach BERGSON-SHILCOCK (2011)...	49
Abbildung 18 - Attributierung der Lizenz CC BY-NC-SA 3.0 DE mittels eines Logos und Hyperlink	57
Abbildung 19 - Vergleich Streckenoptimierung für Fahrradfahrer	63
Abbildung 20 - Campus Routenplaner auf dem Smartphone. Links: OTP-Webserver im mobilen Browser. Mitte und rechts: Der eingerichtete Server eingebunden in die OTP-App unter Android.	66
Abbildung 21 - Analysemodul. Erreichbarkeitsanalyse mit verschiedenen Transportmitteln...	68
Abbildung 22 - Campus Routenplaner	69

TABELLENVERZEICHNIS

Tabelle 1 - Übersicht von Variablen die für webbasierte Karten genutzt werden. Nach LI U. A. (2004).....	20
Tabelle 2 - Inhalt einer GTFS-Datei, nach GOOGLE DEVELOPERS (2013) und HIBBE & ROBECK (2013)	27
Tabelle 3 - Auswahl an OSM-Tags und deren bestimmte Nutzung.....	35
Tabelle 4 - OSM-Tags und deren Gewichtungsmultiplikator	36
Tabelle 5 - Modul- und Funktionsübersicht OpenTripPlanner	39
Tabelle 6 - Beschreibung der API Parameter für eine beispielhafte Anfrage	51
Tabelle 7 - Vergleich der Lizenzmodelle CC BY und CC BY-SA.....	55

QUELLCODEVERZEICHNIS

Quellcode 1 - Befehl zum Croppen eines Ausschnitts aus OSM mit Osmosis anhand einer Bounding Box.....	42
Quellcode 2 - XML-Schema für die statische Liste universitärer Einrichtungen	44
Quellcode 3 - Einbinden zweier alternierender Geocoder.....	45
Quellcode 4 - Konfiguration des Geocoders in der Datei <i>config.js</i>	45
Quellcode 5 - Ausschnitt aus der Datei <i>graph-config.xml</i> . Konfiguration der GTFS-Daten	48
Quellcode 6 - XML-Response auf eine Anfrage an den Geocoder mit dem Begriff Bibliothek	52
Quellcode 7 - Komplette Liste aller universitärer Einrichtungen im alternativen Geocoder	84
Quellcode 8 - Konfigurationsdatei zur Erzeugung des Graph.....	86

1 EINLEITUNG

Routenplaner stellen im alltäglichen Leben vieler Menschen eine kaum mehr wegzudenkende Hilfe dar. Pendler informieren sich über eventuelle Verspätungen, Urlaubsreisen werden interaktiv im Web vorausgeplant, oder man möchte einfach die nächste Tramverbindung zum Bahnhof suchen. Der folgende Abschnitt soll eine kurze Einleitung in das Thema multimodale Routenplaner geben und erläutern, unter welchen Rahmenbedingungen ein solches System für eine beliebige Einrichtung, z.B. für die Universität Potsdam, sinnvoll um- und einsetzbar ist.

1.1 MOTIVATION

Die Universität Potsdam ist die größte Forschungseinrichtung des Landes Brandenburg. Knapp 20.000 Studenten sowie ca. 4.000 Mitarbeiter bewegen sich täglich zwischen den drei Universitätskomplexen *Am Neuen Palais*, *Golm* und *Babelsberg am Griebnitzsee*, sowie Außenstellen, Mensen und Wohnheimen (UNIVERSITÄT POTSDAM - DEZERNAT FÜR PLANUNG STATISTIK FORSCHUNGSANGELEGENHEITEN UND ORGANISATION, 2014). Die Navigation zwischen diesen Einrichtungen stellt nicht nur einen enormen logistischen Aufwand für den lokalen Verkehrsbetrieb Potsdam GmbH (ViP) und regionalen Verkehrsverbund Berlin-Brandenburg (VBB) dar. Auch für Studienanfänger und Gäste gestaltet sich die optimale Bewegung aufgrund der komplexen, dezentralen Struktur der Hochschule oft als schwierig. Radfahrer verlassen sich häufig auf ihre Ortskenntnis, Autofahrer greifen in den meisten Fällen auf



Abbildung 1 - Statischer Lageplan des Campus Golm angeboten auf der Homepage der Universität Potsdam (BRANDT, 2013)

Google Maps zurück und Personen die auf Bus und Bahn angewiesen sind, informieren sich über die Verkehrsauskunftssysteme oben genannter Nahverkehrsanbieter. Die Universität Potsdam selbst bietet zur Orientierung auf dem Campusgelände Lagepläne in Form statischer Karten der Universitätskomplexe auf ihrer Homepage an (Abbildung 1). Daraus resultiert, dass Nutzer gezwungen sind, zwischen verschiedenen Anwendungen hin und her zu wechseln.

Vor diesem Hintergrund wird im Rahmen dieser Arbeit ein multimodaler Routenplaner konzipiert und prototypisch implementiert. Die zentrale Forschungsfrage dabei lautet:

Welche Möglichkeiten und Grenzen bietet die Nutzung von Open Source Software und freien Daten bei der Umsetzung eines webbasierten multimodalen Routenplaners?

Daraus resultieren weitere Fragen:

Ist mit der Nutzung freier Daten und quelloffener Software eine lückenlose Haus-zu-Haus-Navigation (seamless navigation) möglich?

In welchem Rahmen ist eine Datenaufbereitung notwendig, bzw. ist die Datenqualität für die Nutzung als Navigationsgrundlage ausreichend?

Welche Auswirkungen haben freie Lizenzen auf die Veröffentlichung eines solchen Projekts?

1.2 PROBLEMSTELLUNG

Die Arbeit baut auf den Untersuchungen und Ergebnissen des Studienprojektes ‚Konzeption und Implementierung eines multimodalen und multitemporalen Campus-Routenplaners am Beispiel der Universität Potsdam‘ (HIBBE, ROBECK, 2013) auf. Hierbei rückt besonders der technische Aspekt bei der Umsetzung eines Routenplaners in den Fokus.

Die Umsetzung eines multimodalen Routenplaners allein auf der Grundlage quelloffener, freier Software (*Free Open Source Software - FOSS*) und freier Daten stellt auf der einen Seite eine finanzielle Entlastung dar. Die Nutzung von Daten privater Anbieter führt unter Umständen zu einer einmaligen, hohen Investition, aber vor allem die Wartung und Aktualisierung spiegelt sich in wiederkehrenden Kosten wider.

Multimodale und multitemporale Routenplaner unterscheiden sich in zwei Punkten von herkömmlichen Navigationssystemen: Multimodalität verweist auf die Nutzung verschiedener Verkehrsmittel, also ÖPNV, Fahrrad, Auto oder die Fortbewegung zu Fuß. An den Routenplaner wird auch der Anspruch gestellt, dass der Wechsel zwischen den Verkehrsmitteln jederzeit möglich ist. Multitemporalität bedeutet für den Nutzer die freie Wahl der Reisezeit. Der Fokus der theoretischen Grundlagen liegt auf der Analyse solcher Verkehrsnetzwerke. Es wird geklärt, wie Multimodalität und Multitemporalität in einem Navigationssystem umgesetzt und welche Anforderungen dabei an die Daten gestellt werden.

Weiterhin wird betrachtet, wie Campusnavigations- und Informationssysteme anderer Hochschulen aufgebaut sind und umgesetzt wurden. Es gilt zu klären, welche lizenzrechtlichen Aspekte bei der Nutzung von Open Source und freier Software und freien Daten beachtet werden müssen.

1.3 STRUKTUR

In Abschnitt 2 wird geklärt, was unter einem Campusinformationssystem verstanden wird und in welchem Sinne sich der Begriff Campusnavigationssystem davon abgrenzen lässt. Anhand von vier Beispielen werden unterschiedliche Systeme vorgestellt und deren Funktionsumfang beleuchtet. Weiterhin wird eine Auswahl aktueller Entwicklungen und Methoden vorgestellt, die versuchen, Zugriffszeiten für multimodale Routenplaner zu optimieren, und welche Ansätze zur Umsetzung solcher Systeme im Bereich der Open Source Software existieren.

In Abschnitt 3 wird aufgezeigt, welche Funktionen webbasierte Routenplaner bieten. Darauf basierend wird ein Konzept zur Umsetzung eines solchen Systems gestaltet. In einer Übersicht werden Routenplaner der einzelnen Transportmodi vorgestellt und miteinander verglichen. Es wird außerdem auf die dahinter liegenden Mechanismen eingegangen, sowie auf welchen mathematischen Grundlagen das Problem des kürzesten Wegs beruht.

Abschnitt 4 beschreibt die Datengrundlage für die prototypische Implementierung des multimodalen Routenplaners. Es werden die notwendigen Quellen für das Straßen- und Nahverkehrsnetz erläutert. In einer kurzen Einführung zum *OpenStreetMap*-Projekt wird aufgezeigt, wie sich die Verteilung der Objekte vom urbanen zum ländlichen Raum verhält und wie die Daten aufgebaut sind. Im Anschluss wird die für die prototypische Umsetzung zum Einsatz kommende Software *OpenTripPlanner* vorgestellt. Mit der Erläuterung der Funktionsweise und Struktur der Software wird geklärt, wie *OpenTripPlanner* auf die vorgestellten Daten zugreift und sie für ein Routing verwendet.

Eine ausführliche Erläuterung zur Umsetzung des Konzepts für einen multimodalen Routenplaner und der schrittweisen Implementierung liefert Abschnitt 5. Es wird darauf eingegangen, inwieweit die verwendeten Rohdaten aufbereitet werden müssen. Dazu zählen auch die Gestaltung einer Liste universitärer Einrichtungen sowie deren geographischen Koordinaten. Im Anschluss wird beschrieben, wie die prototypische Umsetzung auf einem Webserver veröffentlicht wird.

Abschnitt 6 befasst sich mit der Lizenzierung der einzelnen Komponenten, die in der Arbeit zum Einsatz kommen. Da eine Umsetzung mit freien Daten und Open Source Software angestrebt wird, ist zu klären, unter welchen Rahmenbedingungen die Nutzung, Bearbeitung und Weitergabe der Daten und Software möglich ist.

In Abschnitt 7 werden die während der Arbeit gewonnen Erkenntnisse zusammengefasst und bewertet. Im Ausblick werden weitere Einsatzmöglichkeiten und zukünftige Entwicklungen präsentiert.

2 VERWANDTE ARBEITEN

In diesem Abschnitt werden die Umsetzungen von Informationsdiensten und Orientierungshilfen anderer Universitäten oder Fachhochschulen betrachtet. Im Fokus stehen dabei Routinganwendungen.

2.1 CAMPUSINFORMATIONSSYSTEMS- UND NAVIGATIONSSYSTEME

Der Begriff *Campusinformationssystem* ist nicht eindeutig definiert. Teilweise wird darunter ein webbasierter Dienst zur Organisation des Studiums verstanden, mit dem z.B. Prüfungsergebnisse, Stundenpläne oder studienrelevante Dokumente verwaltet werden. In einigen Fällen bietet ein solches System auch Auskünfte über freie Räume auf dem Campus, die zur Arbeit genutzt werden können (HOCHSCHULE BOCHUM, 2014; RWTH AACHEN, 2014; UNIVERSITÄT HEIDELBERG, 2014). Übersichtspläne der Campusgelände, Orientierungshilfen oder Routinganwendungen sind in diesen Fällen entweder nicht vorhanden oder ausgelagert.

Campusinformationssysteme anderer Hochschulen bieten im Gegensatz dazu (interaktive) Langepläne und Navigationshilfen (siehe Abbildung 2) und lassen sich als Campusnavigationssysteme bezeichnen. Die Martin-Luther-Universität Halle-Wittenberg stellt neben analogen Stadt- und Lageplänen auch *Campus Maps*¹ zur Verfügung (THUERKOW, 2010). Diese Webanwendung erlaubt die

¹ <http://maps.uni-halle.de/>

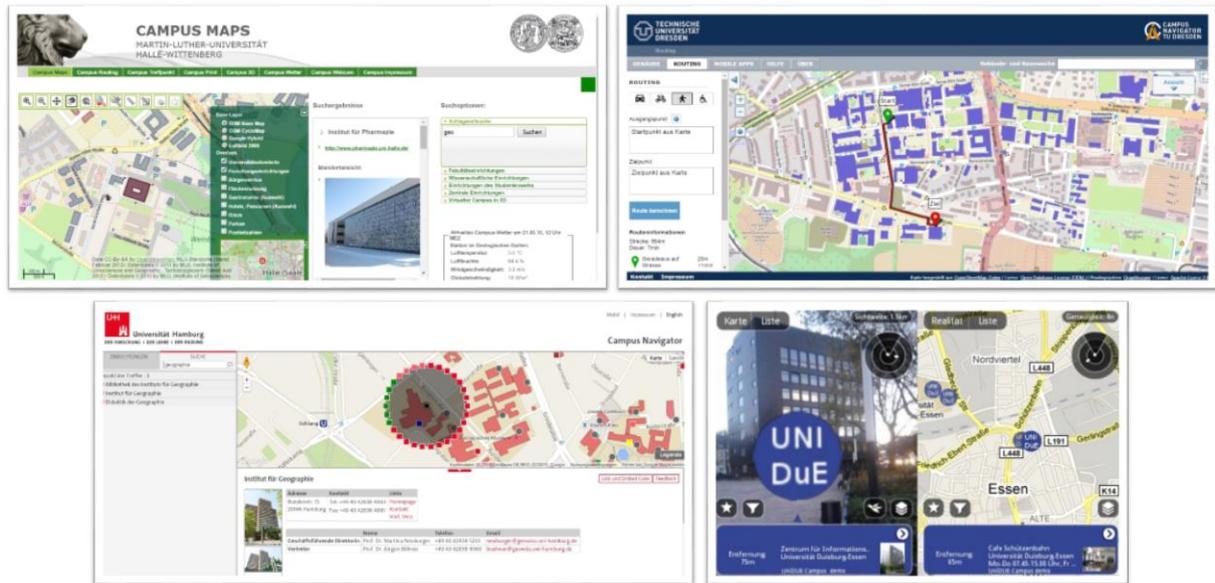


Abbildung 2 - Campusinformations- und Navigationssysteme anderer Hochschulen (l.o.: MARTIN-LUTHER-UNIVERSITÄT HALLE-WITTENBERG 2010; r.o.: TECHNISCHE UNIVERSITÄT DRESDEN 2012; l.u.: UNIVERSITÄT HAMBURG 2013; r.u.: BISCHOFF 2011)

Suche von universitären Einrichtungen auf interaktiven Webkarten, Routing für Fußgänger und Radfahrer auf Basis von *OpenStreetMap* und dreidimensionale Gebäudeansichten in Form von *Google Earth*. Die drei Komponenten sind dabei nicht in einer Anwendung miteinander verbunden, sondern modular getrennt in verschiedenen Webanwendungen. Eine Suchanfrage einer universitären Einrichtung, der Route dorthin und der Visualisierung in der dreidimensionalen Ansicht ist nicht möglich. Der *Campus Navigator*² der Technischen Universität Dresden bietet interaktive Lagepläne mit Suchfunktion, die bis auf Raumebene aufgegliedert ist. Dazu findet sich eine Routinganwendung für motorisierten Individualverkehr, Radfahrer und Fußgänger, sowie barrierefreies Routing (TECHNISCHE UNIVERSITÄT DRESDEN, 2012). Der ebenfalls *Campus Navigator*³ genannte Informationsdienst der Universität Hamburg stellt interaktive Übersichtspläne basierend auf *Google Maps* und eine Einrichtungssuche bereit (UNIVERSITÄT HAMBURG, 2013). Eine Routinganwendung ist nicht vorhanden. Ein Forschungsprojekt der Universität Duisburg-Essen

² <https://navigator.tu-dresden.de/>

³ <https://www.uni-hamburg.de/onTEAM/campus/>

beschäftigt sich mit einem mobilen Smartphone-basierten Informationssystem. Es kombiniert zweidimensionale Lagepläne mit Augmented-Reality-Funktionen, die als Quelle für Auskünfte und Orientierungshilfe dienen sollen (BISCHOFF, 2011).

2.2 MULTIMODALE ROUTENPLANER

Multimodales Routing beschreibt Verfahren zur Wegfindung in einem Netzwerk verschiedener Fortbewegungsmittel. Ein besonderes Problem stellt dabei die Verknüpfung zeitunabhängiger Netze (motorisierter Individualverkehr, Fahrrad, Fußgänger) mit dem zeitabhängigen Netz des öffentlichen Personennahverkehrs dar. Die Optimierung von Algorithmen, Netzwerken und Zugriffszeiten steht im Fokus vieler Untersuchungen. DIBBELT U. A. (2012) untersucht Möglichkeiten der Restriktion von Kombinationsmöglichkeiten verschiedener Transportmodi. Die Extraktion des qualitativ besten Weges aus einer großen Ergebnismenge wird in DELLING U. A. (2012) erforscht. BRAUN (2012) untersucht einen Ansatz, aktuelle, schnelle Routenplanungsmethoden für öffentliche Verkehrsnetze mit zeitunabhängigen Netzen zu kombinieren.

Es existiert eine Reihe von Ansätzen, multimodale Routenplaner in Form von Open Source Software umzusetzen. MARTIN-ANDERSON (2004) entwickelt *GraphServer*⁴, ein multimodaler Routenplaner geschrieben in der Programmiersprache Python, und stellt den Quellcode frei zur Verfügung. GraphServer ist eines der ersten Projekte, welches GTFS-Daten verwendet, ein von Google entwickeltes Format zur Speicherung von Nahverkehrsdaten (siehe dazu Abschnitt 4.2). An der Universität València wird GraphServer zur Umsetzung eines multimodalen Routenplaners mit Daten der OpenStreetMap verwendet (JORDÁN ALDASORRO U. A., 2009). Große Teile der Kernfunktionen von GraphServer werden später in der Umsetzung von *OpenTripPlanner* verwendet, einer Open Source Software, die nicht nur Start-Ziel-Routing unterstützt, sondern auch einfache Netzwerkanalysen in Form von Erreichbarkeitsgraphen,

⁴ <http://graphserver.github.io/graphserver/>

und die Ergebnisse in einer webbasierten Nutzeroberfläche präsentiert (OPENSTREETMAP FOUNDATION, 2014b). Eine ausführliche Analyse des Funktionsumfangs von OpenTripPlanner liefern HILLSMAN & BARBEAU (2011).

3 KONZEPTION

Die zur Umsetzung eines multimodalen Routenplaners notwendigen Daten werden durch seinen gewünschten Funktionsumfang definiert. Zunächst werden grundlegende Funktionen beschrieben. Darauf folgt eine Zusammenfassung des Funktionsumfangs in einem Konzept zur Umsetzung eines Prototyps.

3.1 FUNKTIONSUMFANG

Aufgrund der weiten Verbreitung, die webbasierte Routenplaner erfahren haben und dass sie sich im Aufbau sehr ähneln hat sich eine Erwartungshaltung der Nutzer etabliert. Diese stützt sich im Wesentlichen auf die Nutzung von Routenplanern für den motorisierten Individualverkehr (MIV), wie z.B. bei Falk⁵, ADAC Maps⁶, später Google Maps⁷ und Yahoo⁸. Systeme, die Auskunft über Verbindungen des öffentlichen Personennahverkehrs geben, sind gleichartig aufgebaut, bieten dem Nutzer aber angepasste Konfigurationsmöglichkeiten zur Suche einer passenden Route. Ähnliches gilt für Radfahrer und Fußgänger.

⁵ <http://www.falk.de/>

⁶ <http://maps.adac.de/>

⁷ <https://maps.google.com/>

⁸ <https://maps.yahoo.com/>

Im Wesentlichen findet ein Routenplaner den Weg zwischen einem Ausgangspunkt (Start) und einem Endpunkt (Ziel). Die dafür notwendigen Funktionen sind folgende:

- a) eine Möglichkeit, Start und Ziel auszuwählen
- b) ein Netzwerk beschreibende Daten, um Start und Ziel miteinander zu verbinden
- c) ein Algorithmus, der den Pfad zwischen Start und Ziel entlang des Netzwerks findet
- d) eine Möglichkeit zur Anpassung des Routingalgorithmus durch den Nutzer
- e) die Präsentation der Ergebnisse der Routingabfrage

3.1.1 START UND ZIEL FESTLEGEN

Im Rahmen der Nutzung webbasierter Routenplaner hat sich nach HILLSMAN & BARBEAU (2011) eine Reihe von Varianten etabliert, die es dem Nutzer ermöglichen, den Ausgangs- und Endpunkt der Reise zu definieren.

- ein Klick auf die Karte zum Setzen von Start- und Zielpunkt (*Point and Click*)
- die Eingabe von Start- und Zieladresse
- die Eingabe des Namens (z.B. Institut, Haltestelle, POI) des Start- und Zielpunkts
- Bestimmung der Position durch den GPS-Empfänger eines Smartphones
- Positionsabfrage über Geotargeting der IP-Adresse des Nutzers

Variante Eins benötigt zur Umsetzung eine graphische Präsentation des durch den Routenplaner bedienten Gebietes in Form einer dynamischen Webmap, die genügend Informationen enthält, damit sich der Nutzer auf ihr orientieren und Start und Ziel festlegen kann. Die Klickposition des Cursors wird anschließend in geographische Koordinaten übertragen.

Die Varianten Zwei und Drei greifen auf den Vorgang des Geokodieren (*Geocoding*) zurück, bei dem Straßenadressen und Namen in geographische Koordinaten überführt werden (BARTELME, 2005). Es existieren freie Geocoder,

die für den Einsatz in einem multimodalen Routenplaner genutzt werden können. Nominatim (siehe Abschnitt 5.3.2) ist ein solcher freier Dienst, der entweder von dritten Anbietern zur Verfügung gestellt wird, aber auch auf einem eigenen Webserver eingerichtet werden kann.

Die letzten beiden Varianten sind insbesondere dann von Vorteil, wenn die Position des Nutzers mit dem Ausgangspunkt der Navigationsanfrage übereinstimmt und auf die Eingabe einer Adresse verzichtet werden kann.

3.1.2 ROUTINGFÄHIGES NETZWERK

Für die Bestimmung einer Route zwischen zwei Punkten für Autofahrer wird ein routingfähiges Straßennetzwerk benötigt. Pfade entlang dieses Netzes werden diversifiziert durch Beschränkungen:

- An welchen Stellen ist das Abbiegen erlaubt, eventuell nur aus einer bestimmten Richtung?
- Ist die Einfahrt in Straßenabschnitte nur in eine Richtung erlaubt (Einbahnstraßen) oder nur zu bestimmten Tageszeiten?
- Welche Geschwindigkeitsbeschränkung gilt entlang einer Straße?
- usw.

Andere Transportationsmodi berufen sich ebenfalls auf das Straßennetzwerk, nutzen aber zum Teil Abschnitte, die für Autoverkehr nicht vorgesehen sind. Fußgänger müssen z.B. nicht ausschließlich entlang von befahrenen Straßen laufen, sondern können auch Wege durch Parks nehmen. Radwege sind ebenfalls im Straßennetz enthalten, aber mit eigenen Attributen versehen. Außerdem müssen im Datensatz Informationen darüber enthalten sein, an welchen Stellen ein Wechsel zwischen den Transportmodi möglich ist, z.B. an Haltestellen oder in Fällen von Fußwegen, die per Zebrastreifen oder Ampel die Straßenseite wechseln.

Das ÖPNV-Netz präsentiert sich als komplexes Gefüge aus räumlichen (Haltestellenpositionen, Routen) und zeitlichen (Abfahrtzeiten) Informationen. Datensätze im GTFS-Format haben sich als „de facto Standard zur Beschreibung solcher Netze für Routenplaner etabliert“ (HILLSMAN, BARBEAU, 2011).

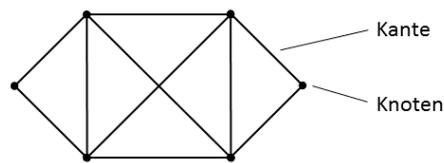


Abbildung 3 - Graph nach BLEEKE (2006)

3.1.3 ROUTINGALGORITHMUS

Zentraler Bestandteil eines Routenplaners ist der zugrunde liegende Algorithmus zum Finden eines Weges zwischen einem vorgegebenem Start- und Zielpunkt. Wichtig ist dabei, dass der Routingalgorithmus auf der einen Seite Knoten und Kanten im Netz auf Grundlage von Nutzerangaben priorisieren kann. Auf der anderen Seite müssen für die Wegfindung aber auch sinnvolle Standards vorgegeben sein, falls der Nutzer keine näheren Angaben macht.

Die theoretischen Grundlagen für das Routing liefert die Graphentheorie. Nach DIESTEL (2000) beschreibt sie die Eigenschaften von Graphen, sowie deren Beziehungen untereinander und dient als Hilfsmittel zur Visualisierung, Analyse und Lösung topologischer Sachverhalte. Ein Graph G setzt sich aus den Elementen V (Knoten, *vertices*) und E (Kanten, *edges*) zusammen. Knoten bilden Anfangs- und Endpunkt einer Kante (Abbildung 3).

Graph: $G = (V, E)$

Knotenmenge: $W = V(G)$, Knoten $v_i \in V(G)$

Kantenmenge: $F = E(G)$, Kante $e_{ij} \in E(G)$

Nach BARTELME (2005) stellt das Straßen- und Transportnetzwerk einen Graphen für den Algorithmus dar. Darin existieren zum einen Knoten, z.B. Straßenkreuzungen oder Stellen, an denen man die Art der Fortbewegung ändern kann, wie Haltestellen. Zum anderen enthält der Graph Kanten, also Verknüpfungen zwischen den einzelnen Knoten. Straßenabschnitte, Radwege oder Schienen stellen Kanten dar. Die Suche nach dem kürzesten Weg zwischen zwei mit einander verbundenen Knoten wird als *Shortest-Path-Problem* bezeichnet. Der Routingalgorithmus sucht nach dem kürzesten Weg zwischen

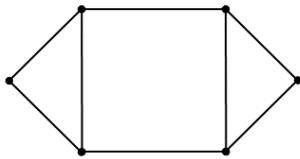


Abbildung 4 - Zusammenhängender Graph nach BLEEKE (2006)

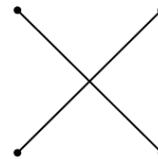


Abbildung 5 - Nicht zusammenhängender Graph nach BLEEKE (2006)

zwei Knoten entlang der Kanten. Wichtig ist dabei, dass der Graph zusammenhängend ist, d.h. dass jeder Knoten, und damit jede potentielle Haltestelle, Straßenkreuzung, usw. über mindestens eine Kante mit dem restlichen Graphen verknüpft ist (vgl. dazu Abbildung 4 und Abbildung 5).

Damit der Routingalgorithmus basierend auf den Vorgaben des Nutzers den besten Weg finden kann, erhalten Knoten und Kanten entsprechend ihrer Attribute eine Gewichtung, z.B. die Entfernung zwischen zwei Knoten (siehe Abbildung 6). Man spricht dabei von einer Metrik. Allen Kanten $e \in E(G)$ des Graphs werden hier über eine Kostenfunktion $f_{kosten}(e)$ positive Werte zugeordnet (MERRIFIELD, 2010). Üblicherweise spielen bei der Wegfindung im Netz des Individualverkehrs lediglich räumliche Faktoren eine Rolle, wohingegen ÖPNV-Netze zusätzlich temporalen Restriktionen durch Abfahrtszeiten unterliegen (BARTELME, 2005). Ein zeit-abhängiger Graph nutzt eine Kostenfunktion $f_{kosten}(e, t)$ mit t für Zeit. Die Kosten, eine Kante zu benutzen, sind hier zusätzlich von der aktuellen Zeit abhängig. Eine andere Bezeichnung für solch ein Netzwerk lautet *dynamischer Graph* (MERRIFIELD, 2010).

Gerichtete Graphen spielen dann eine Rolle, wenn z.B. Abbiegevorschriften und Einbahnstraßen für ein Straßennetz von Bedeutung sind. Kanten erhalten dabei Anfangs- und Endknoten, welche die Richtung bestimmen (siehe Abbildung 7).

Der wohl bekannteste Algorithmus zur Bestimmung des kürzesten Weges in einem Netzwerk ist der von Dijkstra: Es wird ein Startknoten festgelegt, von dem aus die Gewichtung zu benachbarten Knoten geprüft wird. Derjenige, mit der geringsten Distanz wird als besucht markiert und von ihm erneut nach dem

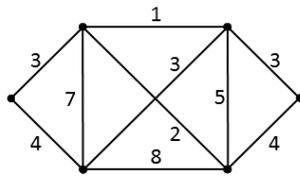


Abbildung 6 - Gewichteter Graph nach BLEEKE (2006)

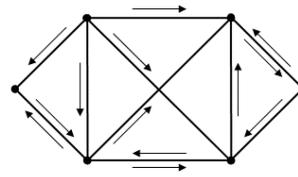


Abbildung 7 - Gerichteter Graph nach BLEEKE (2006)

Knoten mit der geringsten Distanz gesucht (DIJKSTRA, 1959). Der A*-(„A Stern“) Algorithmus als Derivat von Dijkstra kann über einen heuristischen Ansatz die Suchzeit verringern. Dabei wird beispielsweise die gerade geometrische Verbindung (Luftlinie) zwischen zwei Knoten als Schätzwert angenommen. Solange dieser Schätzwert kleiner als die Kosten der tatsächlichen Verbindung zwischen Start- und Zielknoten ist, wird der A*-Algorithmus den kürzesten Weg finden (HART U. A., 1968).

3.1.4 KONFIGURATIONSMÖGLICHKEITEN DURCH DEN NUTZER

Routenplaner stellen dem Nutzer eine Reihe von Anpassungsmöglichkeiten zur Verfügung, um entweder bestimmte Streckenabschnitte zu meiden oder größeren Wert auf besondere Aspekte einer Route zu legen.

MIV-Routenplaner

Betrachtet man den motorisierten Individualverkehr (MIV), werden dem Nutzer meist verschiedene Varianten einer Route angeboten. Für Überlandfahrten spielt die Wahl zwischen Autobahn oder Landstraße eine Rolle. Bei Fahrten im Innenstadtbereich ist oft entscheidend, ob man die schnellste, oder die kürzeste Strecke wählt. Zum Teil wird dem Nutzer außerdem die Möglichkeit gegeben, entlang einer Route verschiedene Stopps zu definieren oder bestimmte Abschnitte basierend auf lokaler Expertise zu umfahren.

ÖPNV-Routenplaner

Routenplaner für öffentliche Verkehrsmittel weisen zum Teil markante Unterschiede bei den möglichen Nutzereinstellungen auf. Nach HILLSMAN & BARBEAU (2011) ist für den Autofahrer meist die folgende Frage entscheidend:

Welcher ist der schnellste Weg?

Für ÖPNV-Nutzer dagegen ist zunächst wichtig:

Kann ich mit diesen Verkehrsmitteln überhaupt mein Ziel erreichen?

Wie anspruchsvoll (z.B. durch häufiges Umsteigen) ist die Fahrt? (HILLSMAN, BARBEAU, 2011)

Der Routenplaner muss dem Nutzer erlauben, entweder eine bestimmte Abfahrts- oder Ankunftszeit festzulegen und die Suche auf ein bestimmtes Verkehrsmittel zu beschränken. In der Regel werden verschiedene Routenvarianten ermittelt, aus denen der Nutzer wählen kann. Zusätzlich sind Auswahlmöglichkeiten nützlich, die Strecke mit der kürzesten Fahrtzeit und der kürzesten Laufwege zur Haltestelle oder den wenigsten Umstiegen zu ermitteln. Wechselt der Fahrgast die Tarifzone oder werden Transportmittel durch unterschiedliche Ticketpreise belegt, kann die günstigste Strecke ausgewählt werden – eine entsprechende Datengrundlage vorausgesetzt.

Fahrradroutenplaner

Radroutenplaner sind noch nicht sehr weit verbreitet. Ein Grund dafür mag die Tatsache sein, dass Radfahrer äußert differenzierte Ansprüche an eine Strecke stellen, die über objektive Kriterien, wie kürzester Weg oder fester Straßenbelag, hinausgehen. Google bietet über Maps zwar Routen für Radfahrer an, weist aber stets darauf hin, dass die Strecke Fehler enthalten und nicht für Radfahrer geeignet sein könne. Einzige Konfigurationsmöglichkeit für den Nutzer ist hier die Option, Fahren zu vermeiden (GOOGLE INC., 2014a). Das Projekt

BBBike⁹ ist ein Routenplaner speziell für Radfahrer im Raum Berlin Brandenburg. Neben der Wahl der durchschnittlichen Fahrgeschwindigkeit steht es dem Nutzer hier außerdem frei, den bevorzugten Straßentyp und -Belag festzulegen, sowie Ampeln oder unbeleuchtete Wege zu meiden (REZIĆ, 1999, 2014). Koomot¹⁰ ist ein Planer für Start-Ziel- und Rundtour-Routen, der diese Art der Konfiguration anhand der körperlichen Fitness und der technischen Ausrüstung des Nutzers regelt. Ob eine Route entlang von POIs über asphaltierte Straßen oder durch unwegsames Gelände führt, wird durch die Wahl des Fahrrads (Trekking-, Mountain-, Rennrad) bestimmt (KOMOOT GMBH, 2014).

Fußgängerrouutenplaner

Routenplaner für Fußgänger sind ähnlich wie Radroutenplaner wenig verbreitet. Bekannte Anbieter von Kartendiensten weisen bei ihren Fußgängerstrecken stets darauf hin, dass Vorsicht geboten sei und nicht überall Fußwege vorhanden sein können (GOOGLE INC., 2014b; MAIRDUMONT GMBH & Co. KG, 2014; MICROSOFT INC., 2014). Für Fußgänger spielt neben der kürzesten Strecke wohl die Sicherheit eine entscheidende Rolle. Daten über Beleuchtung entlang der Strecke, Ampeln an Kreuzungen und Auslastung von Straßen sind wichtig, um entsprechende Anpassungsmöglichkeiten zur Verfügung zu stellen. Betrachtet man Barrierefreiheit, muss der Nutzer außerdem in der Lage sein, Treppen und steile Hänge zu umgehen bzw. abgesenkte Bordsteine zu bevorzugen.

Multimodale Routenplaner

Multimodale Routenplaner stellen alle vorgestellten Methoden zur Verfügung. Markantestes Unterscheidungsmerkmal ist dabei, dass verschiedene Transportmodi untereinander kombiniert werden können, z.B. im Sinne von *Park and Ride*. Es soll die Möglichkeit bestehen, die Nutzung eines bestimmten Transportmittels zu bevorzugen, sodass z.B. Fußwege minimiert werden. Bei

⁹ <http://www.bbbike.de/cgi-bin/bbbike.cgi>

¹⁰ <https://www.komoot.de/>

der Kombination von ÖPNV und Fahrrad wird dem Nutzer teilweise die Wahl gestellt, sein Rad an der Haltestelle abzustellen, oder mitzunehmen, falls die Beförderungsbestimmungen dies zulassen. Nachteil der großen Konfigurationsfreiheit ist, dass die *usability*, also die Einfachheit und Verständlichkeit der Benutzeroberfläche, kompromittiert wird. Es müssen also sinnvolle Standardeinstellungen für die verschiedenen Transportmodi gefunden werden. Andernfalls besteht die Möglichkeit, die erweiterten Einstellungen, die ein Nutzer vornimmt, für spätere Suchanfragen zu speichern.

3.1.5 PRÄSENTATION DER ROUTINGERGEBNISSE

Laut HILLSMAN & BARBEAU (2011) geht der Mensch zwei verschiedenen Ansätzen nach, einer gegebenen Route zu folgen. Der erste Ansatz wird als räumlich-kognitive Navigation bezeichnet und beschreibt den Vorgang des Orientierens im Gelände mittels einer graphischen Repräsentation der Umgebung in Form einer Karte. Anhand der gegebenen Informationen, wie Straßen- und Wegverlauf, Gebäude, POIs usw. ist der Nutzer in der Lage, seine Position zu bestimmen und einer Route zu folgen. Die zweite Variante wird narrative oder Reiz-/ Reaktions-Navigation genannt und basiert auf der Angabe von Richtungsanweisungen, um den Nutzer an sein Ziel zu führen. Dem Nutzer eine Kombination aus beiden Varianten anzubieten, hat sich bei den meisten webbasierten Routenplaner durchgesetzt. Der Fensterinhalt wird üblicherweise vertikal im Verhältnis 1:3,5 geteilt, wobei links die Navigationsangaben zu finden sind und rechts die Kartenansicht mit der hervorgehobenen Route (siehe Abbildung 8). Die schrittweisen Navigationsangaben setzen sich aus folgenden Elementen zusammen:

- Straßennamen
- Direktionale Anweisungen durch Himmelsrichtungen
- Streckenangaben oder geschätzte Wegzeit
- Zustand von Wegen oder Straßen
- POIs

Die zusätzlichen Informationen dienen der Orientierung, auch wenn Straßennamen nicht verfügbar sind, z.B. in Parks oder auf einem Campus.

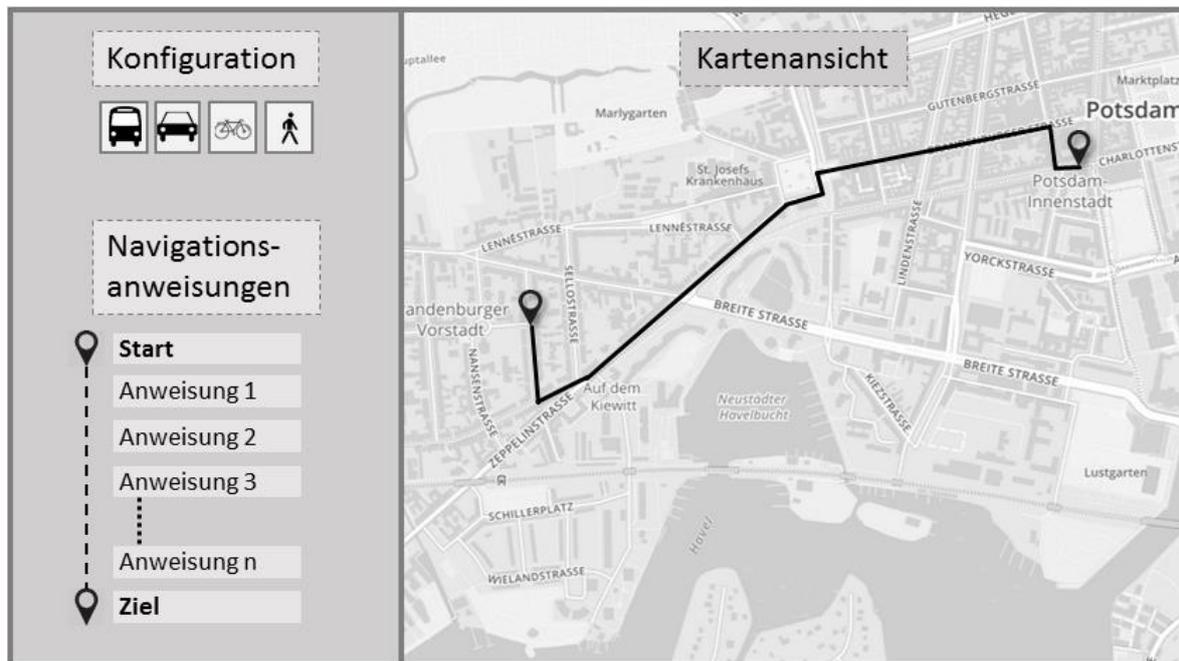


Abbildung 8 - Schematische Übersicht eines webbasierten Routenplaners

Die Kartenansicht muss den üblichen kartographischen Ansprüchen genügen und den grundlegenden graphischen Variablen folgen. Mit den Entwicklungen, welche die digitale Kartographie, besonders das Webmapping, mit sich bringen, müssen diese traditionellen Grundsätze erweitert werden. Tabelle 1 zeigt eine Übersicht möglicher Variablen, die über die Visualisierungsprinzipien von analogen Karten hinausgehen.

Die genannten zusätzlichen Variablen geben zum einen mehr Spielraum zur Visualisierung der Routingergebnisse. Andererseits ermöglichen vor allem das Vergrößern und Verkleinern, Schwenken und Hervorheben von Objekten dem Nutzer die Möglichkeit, sich den Streckenverlauf im Detail anzusehen und sich mit den örtlichen Gegebenheiten vertraut zu machen. Besonders zu Fuß sind Informationen über Landmarken und POIs wichtig, da sie das Fehlen von Straßennamen kompensieren können. Auch das Einblenden von optionalen Hinweisen über Streckenabschnitte, z.B. Baustellen, hohes Verkehrsaufkommen oder das Vorhandensein von Fahrradwegen, helfen dem Nutzer zu verstehen, warum eine bestimmte Route gewählt wird. Dadurch „entwickelt der Nutzer ein größeres Vertrauen in die Zuverlässigkeit des Routenplaners“ (HILLSMAN, BARBEAU, 2011).

Graphische Variablen	Größe, Form, Helligkeit, Richtung, Farbe, Muster (BERTIN, 1974)
Dynamische Variablen	Darstellungsdauer, Änderungsrate, Ebenenreihenfolge (DIBIASE U. A., 1992)
Untersuchungsmöglichkeiten	Verschieben (<i>drag</i>), Schwenken (<i>pan</i>), Vergrößern und Verkleinern (<i>zoom</i>), Hervorheben (<i>highlight</i>), Aufleuchten (<i>blink</i>), Auswählen (<i>click</i>) (JIANG, 1996)
Displayvariablen	Trüben (<i>blur</i>), Fokussieren (<i>focus</i>), Transparenz (VAN DEN WORM, 2001)
Webbasierte Variablen	Wechseln von Informationen (<i>browse</i>), Nutzung von Plug-Ins (LI U. A., 2004), Hyperlinks

Tabelle 1 - Übersicht von Variablen die für webbasierte Karten genutzt werden. Nach LI U. A. (2004)

Die Kartenansicht ermöglicht es dem Nutzer, ohne viel Interpretationsaufwand, die vom Routenplaner vorgeschlagenen Ergebnisse zu verstehen. Der Fokus liegt auf dem Routenverlauf. Zusätzlich werden Start- und Zielpunkt angegeben, an welchen Stellen der Wechsel von Verkehrsmitteln notwendig ist und wo mit eventuellen Hindernissen oder Unterbrechungen zu rechnen sein wird. Die Grundkarte steht dabei nicht im Vordergrund, sondern dient lediglich als Orientierungshilfe. Sie liefert einen topographischen Überblick über die Umgebung. Die Symbolisierung ist dabei einfach und verständlich gehalten, damit das Hauptaugenmerk auf der Route als Navigationshilfe ruht.

Abschließend sind zwei weitere Funktionen für den Nutzer wichtig. Der Routenplaner muss eine Möglichkeit bieten, die Routingergebnisse auszudrucken und unterwegs zur Hand zu haben. Dafür wird der Inhalt des Browserfensters dahingehend formatiert, dass Routinganweisungen und deren Position auf der Karte auf eine oder mehrere DIN A4 Seiten passen. Eine andere Möglichkeit besteht darin, das Ergebnis auf dem Smartphone oder Tablet offline zu speichern. Routinganweisungen können dann anhand der per GPS ermittelten Position des Nutzers ausgegeben werden. Außerdem muss der Nutzer in der Lage sein, dem Betreiber des Routenplaners Rückmeldungen zu

geben, ob es eventuelle Probleme während der Navigation gibt, die aus der gewählten Route resultieren.

3.2 ZUSAMMENFASSUNG

Beruhend auf den in Abschnitt 3.1 dargelegten Funktionen, die ein multimodaler Routenplaner bereitstellen muss, lässt sich ein grober Ablauf der Verwendung durch einen Nutzer abstrahieren und im Weiteren spezieller differenzieren. Abbildung 9 fasst diesen Ablauf zusammen:

1. Start und Ziel festlegen
2. Routingalgorithmus bestimmt Strecke
3. Visualisierung der Ergebnisse

In einem ersten Schritt werden Start und Ziel festgelegt. Dies geschieht entweder über

- (a) die Eingabe einer Adresse oder eines POIs in einem Suchfeld, oder
- (b) durch direktes Wählen von Start- und Endpunkt der Route in einer interaktiven Karte.

Die Eingabe des Nutzers wird im Falle von (a) mittels direkter Geokodierung in geographische Koordinaten umgewandelt. Im Fall (b) werden die Koordinaten direkt übergeben. In beiden Varianten werden die ermittelten Koordinaten den nächstgelegenen Knoten im hinterlegten Graphen zugeordnet und bestimmen damit Ausgangs- und Endpunkt für den Routingalgorithmus. Weiterhin legt der Nutzer den Transportmodus fest. Wird dabei eine Kombination gewählt, welche den Modus *Fußgänger* enthält, kann außerdem definiert werden, welche maximale Strecke zu Fuß zurückgelegt werden soll. Damit lässt sich z.B. ein längerer Fußweg zu einer Haltestelle in Kauf nehmen, von welcher im Gegenzug eine schnellere Nahverkehrsverbindung möglich ist. Schließlich besteht die Möglichkeit, die gewünschte Abfahrts- oder Ankunftszeit festzulegen.

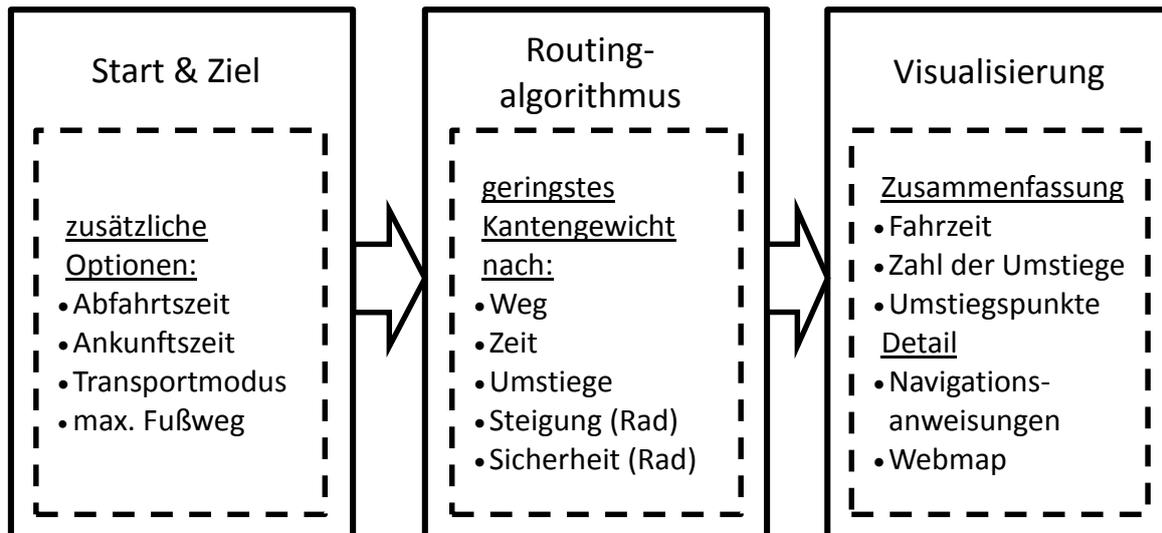


Abbildung 9 - Konzept für den Funktionsablauf eines multimodalen Routenplaners

Durch den Routingalgorithmus wird die Strecke mit der für den Graphen geringsten Kantengewichtung berechnet. Dabei handelt es sich entweder um eine Strecke mit der geringsten Länge, der kürzesten benötigten Fahrzeit oder mit den wenigsten Umstiegen. Für Radfahrer besteht die Möglichkeit der Routenführung anhand der Kriterien Schnelligkeit, Steigung oder Sicherheit. Anhand der nutzergesteuerten Präferenzen werden die günstigste Strecke, sowie eine Reihe von Alternativen, zurückgegeben.

Die berechnete Strecke wird in einer zusammengefassten Übersicht mit Abfahrts- und Ankunftszeit, der Gesamtfahrzeit, sowie den verwendeten Transportmodi dem Nutzer präsentiert. Die Detailansicht enthält Navigationsanweisungen mit Richtungsangaben und Entfernungen der einzelnen Abschnitte. In der interaktiven Webmap wird der Streckenverlauf visualisiert. Die Navigationsangaben der Detailansicht sind mit dem Streckenverlauf verknüpft. Der Nutzer erhält über einen visuellen Indikator einen Hinweis, wo entlang der Route sich der entsprechende Streckenabschnitt befindet.

4 DATENGRUNDLAGE

Für die Umsetzung prototypische Implementierung des multimodalen Routenplaners kommen freie Daten zum Einsatz. Der folgende Abschnitt liefert eine Übersicht der dafür zur Verfügung stehenden Datenquellen. Außerdem wird die zur Umsetzung verwendete Software vorgestellt.

4.1 OPENSTREETMAP

Seit seiner Gründung im Jahr 2004 in Großbritannien unter der Leitung von Steve Coast entwickelt sich das OpenStreetMap¹¹ (OSM)-Projekt zu einer beachtlichen Quelle für detaillierte, umfangreiche raumbezogene Daten. Aufgrund der Art und Weise, wie diese Daten zusammengetragen werden, wird oft von der „Wikipedia der Kartographie“ (RAMM, TOPF, 2010) gesprochen. Denn genau wie bei der freien, online verfügbaren Enzyklopädie, werden die Inhalte von OpenStreetMap von Nutzern auf freiwilliger Basis selbst generiert. Dabei ist zu beachten, dass das entstehende Kartenmaterial sowohl *kostenlos*, als auch zur *freien* Verwendung und Veränderung zur Verfügung steht. Dieses Prinzip wird durch die Open Data Commons Open Database Lizenz gewährleistet (siehe 6.1) (OPENSTREETMAP FOUNDATION, 2014a). Der in diesem Zusammenhang häufig verwendete Begriff *Crowdsourcing*, in Anlehnung an *Outsourcing*, geht auf den Journalisten JEFF HOWE (2006) zurück und beschreibt die Auslagerung von Aufgaben an Freiwillige.

¹¹ <https://www.openstreetmap.org/>

Mit Stand Januar 2015 zählt OSM gut 1,8 Millionen registrierte Nutzer, wobei gut ein Viertel davon aktiv ist und regelmäßig Änderungen und/oder Neuerungen zum Datensatz beiträgt (OPENSTREETMAP FOUNDATION, 2014c). Untersuchungen zum Vergleich der Datenqualität zwischen OSM und Anbietern offizieller bzw. kommerzieller Geodaten in Großbritannien und Deutschland von HAKLAY (2010) und ZIELSTRA & ZIPF (2010) zeigen, dass die Lagegenauigkeit größtenteils übereinstimmt. Weiterhin wird aber auch festgestellt, dass die Vollständigkeit von OSM in dünn besiedelten Gebieten z.B. den Daten von TeleAtlas unterlegen sei. Allerdings „[...] in Groß- und Mittelstädten übertrifft die [Anzahl der Objekte] von OpenStreetMap die von TeleAtlas sogar“ (ROICK U. A., 2011).

Aus diesem Grund hat sich OpenStreetMap als akzeptierte Alternative zu anderen Webmap-Diensten wie Google Maps und Bing Maps¹² etabliert. Wikipedia¹³ nutzt z.B. Daten basierend auf OSM, um Karten für spezielle Artikel zu rendern (WIKIMEDIA LABS, 2014). Auch Social Media Plattformen wie Flickr¹⁴ und Foursquare¹⁵ setzen auf die freie Weltkarte (FOURSQUARE, 2014; YAHOO INC., 2008). Aber nicht nur die alleinige Nutzung von OSM als Grundkarte ist weit verbreitet: Aufgrund der Datenstruktur ist es möglich, Routinganfragen an OSM zu stellen. Dafür benötigt werden u.a. Angaben über Einbahnstraßen, Hausnummern und Straßennamen, sowie die sogenannten *turn restrictions*, also ob an Straßenkreuzungen das Abbiegen erlaubt ist. Der von der Universität Bonn entwickelte und im Moment von der Universität Heidelberg betriebene webbasierte Dienst *OpenRoutingService*¹⁶ bietet neben Navigation für Autofahrer auch Streckenfindung für Radfahrer und Fußgänger (RAMM, TOPE, 2010).

In ihrer Struktur ähneln die Daten von OSM bereits denen eines routingfähigen Graphen. Bei den zwei elementarsten Objekten handelt es sich um Knoten,

¹² <https://www.bing.com/maps/>

¹³ <https://de.wikipedia.org/wiki/Wikipedia:Hauptseite>

¹⁴ <https://www.flickr.com/>

¹⁵ <https://de.foursquare.com/>

¹⁶ <http://www.openrouteservice.org/>

sogenannten *nodes*, und Kanten, *ways*, den Verbindungen zwischen den Knoten. Ein Polygon, also ein Objekt welches eine Fläche beschreibt, existiert gewissermaßen nicht. Erst durch Relationen wird festgelegt, ob mindestens drei Kanten die miteinander verbunden sind und deren Anfangs- und Endknoten an der gleichen Position liegen ein geschlossenes Polygon bilden. Die Grenzen eines Gebäudes beispielsweise erzeugen noch kein geschlossenes Objekt. Erst durch die Attributierung der Kanten und deren Verbindung zueinander entsteht ein Gebäudegrundriss. Attributdaten werden in *Tags* gespeichert. Sie können an jedes geometrische Grundelement und an Relationen angehängt werden. Ein Tag besteht im Allgemeinen aus einem Schlüssel, *key*, und einem Schlüsselwert, *value*. Zu den am häufigsten verwendeten key/value-Kombinationen zählen:

- Gebäude (*building=yes*)
- jegliche Arten von Straßen (*highway=**)
- Bäume (*natural=tree*) (OPENSTREETMAP FOUNDATION, 2014d).

Im Prinzip sind dem Nutzer bei der Vergabe von Schlüsseln und Schlüsselwerten keine Einschränkungen gesetzt. Allerdings haben sich über die Jahre, basierend auf der Häufigkeit der genutzten Tags, einige Konventionen entwickelt, wie TOPF & TOPF (2014) zeigen. Die Nutzer sind angehalten, diese zu befolgen.

Als Datengrundlage stellt OSM u.a. jenes Straßennetzwerk zur Verfügung, welches für Routinganfragen für den MIV notwendig ist. Es enthält Informationen über die bereits erwähnten *turn restrictions*, Geschwindigkeitsbeschränkungen, Landmarken oder POIs und Adressdaten. Weiterhin sind Wegabschnitte enthalten, die speziell für Fußgänger oder Radfahrer ausgezeichnet sind, also z.B. Abkürzungen, Waldwege usw. Ebenso weist OSM Daten zu Haltestellenpositionen, Verkehrsmitteln und Linienverbindungen des ÖPNV auf. Abfahrt- und Fahrzeiten fehlen in der Datenstruktur allerdings. Diese werden durch eine andere Datenquelle, der General Transit Feed Specification (siehe 4.2), bereit gestellt.

OpenStreetMap-Daten sind frei verwendbar und stehen auf Regionen, z.B. Länderebene, zugeschnitten u.a. bei der Geofabrik zum Download zur Verfügung (GEOFABRIK GMBH, 2013).

4.2 GENERAL TRANSIT FEED SPECIFICATION

Das Datenformat General Transit Feed Specification (GTFS) stellt Google ursprünglich im Jahr 2005 vor, damals noch unter dem Namen *Google Transit Feed Specification*. Zu diesem Zeitpunkt wird es in Zusammenarbeit mit dem in Portland ansässigen Verkehrsunternehmen TriMet entwickelt (GOOGLE DEVELOPERS, 2012a). Seitdem ist vor allem in den USA die Nutzung von GTFS als Modell zu Speicherung von ÖPNV-Daten weit verbreitet (HILLSMAN, BARBEAU, 2011). Das Format kommt im ÖPNV-Modul von Google Maps zum Einsatz. Das bedeutet, dass im Prinzip überall dort, wo Suchanfragen auf das regionale oder lokale Tram-, Bus-, sowie U- und S-Bahnnetz über Google Maps möglich sind, ein GTFS-Datensatz existiert. Dieser muss aber nicht zwingend für jedermann zur freien Verfügung stehen (MATZAT, 2012). Nichtsdestotrotz gilt das Format als Standard für ÖPNV-Daten. Online¹⁷ werden frei verfügbare GTFS-Datensätze gesammelt und zur Nutzung bereitgestellt. Downloads von insgesamt 829 Verkehrsunternehmen aus aller Welt stehen dort zur Verfügung (CZEBOTAR, 2014).

Ein GTFS-Datensatz besteht aus einer Reihe kommaseparierter Textdateien, die in einem ZIP-Archiv zusammengefasst werden. Tabelle 2 gibt eine Übersicht der enthaltenen Dateien und deren Bedeutung für den Datensatz.

¹⁷ <http://www.gtfs-data-exchange.com/>

Dateiname	Notwendigkeit	Inhalt
agency.txt	obligatorisch	Welche(s) Verkehrsunternehmen stellen/stellt die Daten zur Verfügung
stops.txt	obligatorisch	Haltestellenpositionen
routes.txt	obligatorisch	Linienverbindungen der im Verkehrsnetz verfügbaren Verkehrsmitteln
trips.txt	obligatorisch	Fahrten entlang einer Route, bestehend aus mindestens zwei miteinander verbunden Haltestellen
stop_times.txt	obligatorisch	Ankunfts- und Abfahrtzeiten von Verkehrsmitteln an Haltestellen
calendar.txt	obligatorisch	Termine, wann eine bestimmte Route bedient wird, basierend auf einem wöchentlichen Rhythmus
calendar_dates.txt	optional	Ausnahmen zu den in calendar.txt beschriebenen Daten
fare_attributes.txt	optional	Fahrpreisinformationen
fare_rules.txt	optional	Regeln zur Berechnung des Fahrpreises
shapes.txt	optional	Exakte Geometrien zur Visualisierung der Routen
frequencies.txt	optional	Zeit zwischen Fahrten, für Routen mit variablen Bedienfrequenzen
transfers.txt	optional	Regeln zur Erstellung Umsteigebedingungen an Knotenpunkten
feed_info.txt	optional	Zusätzliche Metadaten wie Herausgeber, Versionsnummer, Verfallsdatum

Tabelle 2 - Inhalt einer GTFS-Datei, nach GOOGLE DEVELOPERS (2013) und HIBBE & ROBECK (2013)

Die einfache Datenstruktur eines GTFS-Pakets in Form von simplen Textdateien macht die Be- und Verarbeitung unproblematisch. Im Falle einer Änderung entlang einer Linie, von z.B. Abfahrtzeiten, müssen nicht alle Dateien angepasst werden, sondern lediglich einzelne Einträge in der Datei `stop_times.txt`. Es existieren eine Reihe von proprietären und Open Source Tools zur Generierung und Manipulation von GTFS-Daten, z.B. der *GTFS Manager* von Trilium Solutions oder der Java-basierte *GTFS Editor* (GOOGLE CODE, 2014).

Für die einheitliche Verkehrsstruktur aller ÖPNV-Aufgabenträger zeichnet sich der Verkehrsverbund Berlin-Brandenburg (VBB) verantwortlich. Zu seinen Aufgaben zählen unter anderem die „[Abstimmung von] Bahn- und Busfahrpläne[n], [...] Sicherung der Anschlüsse und [Vermittlung von] zunehmend auf Echtzeitdaten zurückgreifende Fahrgastinformationen“ (VBB VERKEHRSVERBUND BERLIN-BRANDENBURG GMBH, 2015). Seit Ende 2012 stellt der VBB seine Fahrplandaten im GTFS-Format auf dem Portal für offene Daten der Stadt Berlin unter der Creative-Commons-Lizenz CC-BY (siehe 6.2) zur Verfügung. Anstoß dazu gab das Projekt OpenplanB von KREIL (2012). Er und andere Mitarbeiter veröffentlichten „im September 2012 [...] unter anderem die Daten des VBB auf eigene Faust [...]“ (THOMA, 2013). Mit der Freigabe der Daten im Rahmen des Entwicklertages *Apps & the City* wolle man „kreativen und engagierten Programmierern, Webentwicklern und Open-Data-Enthusiasten“ die Möglichkeit geben „kundenorientierte Fragestellungen zum öffentlichen Personennahverkehr“ aufzugreifen und zu beantworten (VBB VERKEHRSVERBUND BERLIN-BRANDENBURG GMBH, 2012). Damit ist der VBB das einzige überregionale Verkehrsunternehmen in Deutschland, welches seine Fahrplandaten veröffentlicht und zur Weiterverwendung zur Verfügung stellt.

Das Fahrplanauskunftssystem des VBB verbindet die Plandaten aller in Berlin und Brandenburg beheimateten Anbieter von Nahverkehr. Über eine gemeinsame Schnittstelle wird ein unternehmensübergreifendes Routing ermöglicht. Technisch kommt dabei das HaCon Fahrplan-Auskunfts-System (HAFAS) der Hannover Consulting (HaCon) zum Einsatz, d.h. die Daten liegen im softwareeigenen Format *Hafas Raw Data Format* (HRDF) vor. Die Rohdaten können in das GTFS-Format überführt werden. Die komplette Daten- und Informationsstruktur des VBB ist in Abbildung 10 zu sehen. Die notwendigen Schritte zur Erzeugung des GTFS-Datensatzes sind unten links im grünen Kasten der Abbildung markiert.

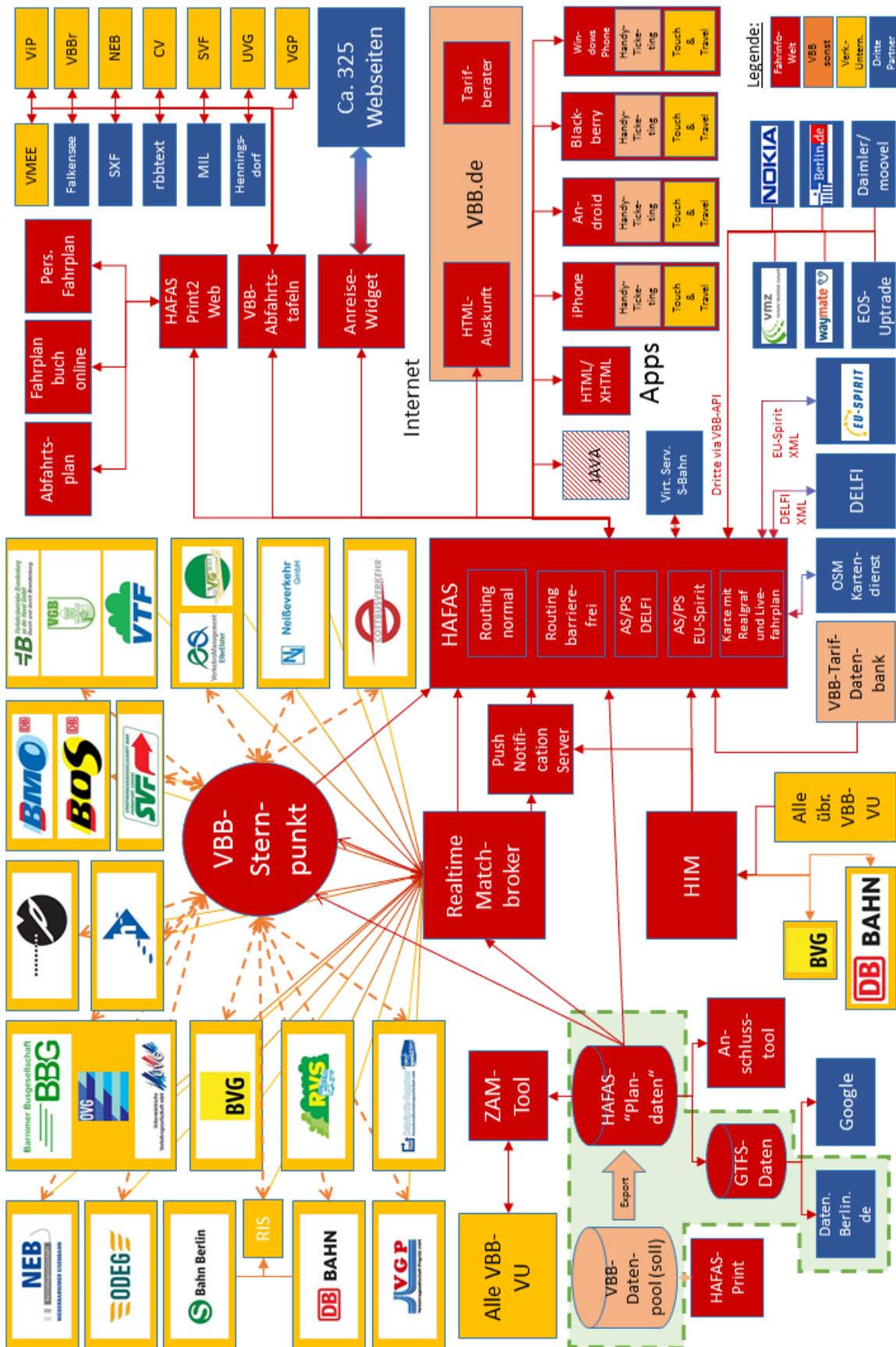


Abbildung 10 - Daten- und Informationsstruktur des Verkehrsverbunds Berlin-Brandenburg nach VBB VERKEHRSVERBUND BERLIN-BRANDENBURG GMBH (2014)

Der GTFS-Datensatz, den der VBB frei zur Verfügung stellt, umfasst alle in Brandenburg und Berlin ansässigen regionalen und lokalen Nahverkehrsanbieter. Für die Fragestellung dieser Arbeit ist allein der Raum Potsdam von Bedeutung. Aus diesem wurde ein auf das Stadtgebiet zugeschnittener Datensatz zur Verfügung gestellt, bestehend aus folgenden Anbietern:

- S-Bahn Berlin GmbH
- Havelbus Verkehrsgesellschaft mbH
- DB Regio AG
- Verkehrsbetrieb Potsdam GmbH

4.3 OPENTRIPPLANNER

OpenTripPlanner (OTP) ist eine Software zur Bereitstellung eines webbasierten multimodalen Routenplaners. OTP ist Open Source, d.h. es steht jedem frei, den Quellcode des Programms auf den eigenen Rechner herunterzuladen, auszuführen und Veränderungen daran vorzunehmen. Es handelt sich um ein gemeinschaftliches Projekt verschiedener Akteure:

- der Nahverkehrsverband *TriMet*¹⁸ der Region Portland, Oregon,
- *OpenPlans*¹⁹, einem Projekt für kollaborative Webkarten,
- den Entwicklern von *Five Points*²⁰, einem Entwicklerteam für Webseiten,
- *OneBusAway*²¹, einem Projekt, welches sich auf die Entwicklung von freien Echtzeitanwendungen für Verkehrsunternehmen spezialisiert hat und
- *Graphserver*, einem frühen multimodalen Routenplaner (OPENSTREETMAP FOUNDATION, 2014b)

¹⁸ <http://trimet.org/>

¹⁹ <http://openplans.org/>

²⁰ <http://www.fpdev.org/>

²¹ <http://onebusaway.org/>

Mit Stand Januar 2015 befindet es sich in der Entwicklung. In unregelmäßigen Abständen werden neue Versionen mit verbessertem Quellcode und neuen Funktionen veröffentlicht. Das Projekt ist auf *GitHub*²², einer Plattform zum Management und zur Versionierung von Quellcode, verfügbar.

OTP bietet zwei Hauptanwendungen: zum einen den multimodalen Routenplaner und zum anderen ein Werkzeug für Netzwerkanalysen basierend auf Fahrplandaten. Der Routenplaner steht im Rahmen dieser Arbeit im Fokus. OTP bietet einerseits eine webbasierte Nutzeroberfläche für multimodale Routinganfragen, wie sie in Abschnitt 3.1.5 beschrieben ist. Andererseits wird eine Programmierschnittstelle (*Application Programming Interface* – API) bereitgestellt, die es Programmen dritter Anbieter ermöglicht, auf die Routing Engine zuzugreifen.

Zur Berechnung der Routen greift die Software auf die in den Abschnitten 4.1 und 4.2 vorgestellten Daten zurück. Ein auf das Untersuchungsgebiet zugeschnittener Ausschnitt der OSM-Daten stellt die Grundlage zur Routenberechnung der Transportmodi Fußgänger, Radfahrer und MIV dar. GTFS-Datensätze bilden die Grundlage für das Routing im ÖPNV-Netz. Für diese Arbeit wird auf ein einzelnes GTFS-Paket zurückgegriffen. Für ein überregionales Routing, basierend auf den Daten verschiedener Nahverkehrsanbieter, ist auch die Verwendung mehrerer GTFS-Datensätze möglich.

Im Kern besteht der OTP aus drei grundlegenden Softwarekomponenten: dem *Graph-Builder*, der *Routing Engine* und der graphischen Benutzeroberfläche. Der Graph-Builder erzeugt auf Grundlage der OSM- und GTFS-Daten ein routingfähiges Netzwerk. Die beiden Datenquellen werden über die Haltestelleninformationen miteinander verknüpft, da deren Position und Name in beiden Datensätzen enthalten sind. Die Routing Engine greift auf zwei Algorithmen zur Berechnung von Strecken zurück: den A*-Algorithmus und einer relativ neuen Methode, Knotenhierarchie (*Contraction Hierarchies*)

²² <https://github.com/>

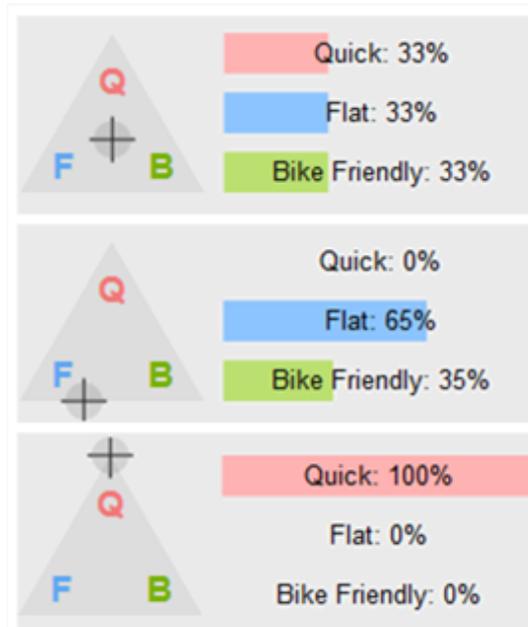


Abbildung 11 - Konfigurationsdreieck für Fahrradfahrer in verschiedenen Einstellungen.

genannt. Sowohl der Graph-Builder, als auch die Routing Engine und andere Komponenten des OTP sind in Java geschrieben. Die webbasierte graphische Nutzeroberfläche basiert auf HTML, CSS und JavaScript (OPENSTREETMAP FOUNDATION, 2014b).

Weiterhin bietet OTP folgende Funktionalitäten:

- Angabe von Ankunfts- bzw. Abfahrtzeiten
- Optionale Anpassungsmöglichkeiten zur Routenberechnung wie
 - die schnellste Route
 - die wenigsten Umstiege
 - für Fahrradrouten über ein „Konfigurationsdreieck“ die schnellste, flachste oder fahrradfreundlichste Route (siehe Abbildung 11)
 - Fahrradverleihstationen
 - Barrierefreiheit
 - maximaler Laufweg zu oder zwischen Haltestellen
- Bestimmung von Start- und Ziel entweder per Point and Click oder Adresseingabe
- Teilen von Routingergebnissen per statischer URL (Permalink)
- Routing über nichtlineare Streckenabschnitte wie Freiflächen und Plätze

4.3.1 ROUTING UND ALGORITHMEN DES OTP

Die Routing Engine des OTP nutzt den A*-Algorithmus und Knotenkontraktionen zur Berechnung von Strecken. Früheren Iterationen der Software verwenden über alle Transportmodi hinweg der A*-Algorithmus mit einer monotonen Heuristik basierend auf der euklidischen Distanz. Dieser Algorithmus führt allerdings mit wachsendem Graph zu einer Leistungsabnahme. Die Methode der Knotenhierarchie nach GEISBERGER (2008) erweist sich nach ihrer Implementierung als effektive und schnelle Alternative zur Routenplanung auf großen Graphen.

Betrachtet werden alle eingehenden Kanten (v, u) und alle ausgehenden Kanten (u, w) . Existiert ein Pfad $\langle v, u, w \rangle$ als einziger kürzester Weg zwischen v und w , wird eine neue Kante (v, w) eingefügt. Diese stellt eine Abkürzung dar. Im Ergebnis entsteht eine Kontraktionshierarchie (*contraction hierarchy*), bestehend aus Originalgraph und allen Abkürzungskanten (siehe Abbildung 12).

Darauf wird ein bidirektionaler Dijkstra-Algorithmus angewendet, d.h. es wird jeweils von Start- und Endknoten simultan eine Suche nach dem kürzesten Weg gestartet. Da der gefundene Weg Abkürzungskanten enthalten kann, müssen diese durch Verbindungen aus dem Originalgraph ersetzt werden. Insgesamt kann damit „im Vergleich zum bisher besten [...] hierarchischen Dijkstra-basierten Verfahren [...] die Suchzeit nochmals um den Faktor fünf verkleinert werden“ (GEISBERGER, 2008).

Mit der Implementierung stellt sich heraus, dass Kontraktionshierarchien sich nicht auf Graphen anwenden lassen, deren Knoten einer temporalen Abhängigkeit unterliegen. Demzufolge erhalten lediglich Suchanfragen auf

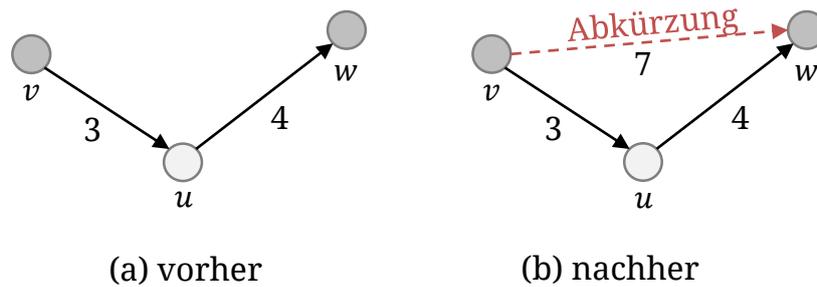


Abbildung 12 - Erzeugung einer Abkürzungskante (nach Geisberger 2008)

statischen Netzen den Geschwindigkeitsvorteil. Basierend auf den Ergebnissen von MERRIFIELD (2010) wird für ÖPNV-Netze ein A*-Algorithmus mit verbesserten, gitternetzbasierenden Heuristiken in das OTP-Projekt integriert.

4.3.2 ROUTING BASIEREND AUF OSM-TAGS

Um zu bestimmen, auf welchen Teilstrecken des Verkehrsnetzes welcher Transportmodus erlaubt ist, greift der OTP auf die tags aus OSM zurück. Grundlegend haben Fußgänger und Radfahrer keine Berechtigung Abschnitte zu nutzen, die unter folgende Kategorien fallen:

- Autobahn
- Schnellstraße
- Auffahrten zur Autobahnen und Schnellstraßen

Hingegen ist für Fußgängerzonen, Radwege und Fußwege der MIV gesperrt. Andere Sonderfälle ergeben sich aus zusätzlichen Attributierungen. Tabelle 3 zeigt eine Auswahl von OSM-Tags, deren allgemein verbreitete Bedeutung und erlaubte Nutzung.

Zugangs- bzw. Durchfahrtbeschränkungen müssen vom Routingalgorithmus ebenfalls beachtet werden. Diese sind gekennzeichnet als

`access=no`

oder

`access=private.`

Erlaubte Verkehrsmittel	OSM-Tag	Entsprechung
MIV	<code>highway=trunk</code>	Schnellstraße
	<code>highway=trunk_link</code>	Schnellstraßenauffahrt
	<code>highway=motorway</code>	Autobahn
	<code>highway=motorway_link</code>	Autobahnauffahrt
Fußgänger (Radfahrer)	<code>highway=pedestrian</code>	Fußgängerzone
Radfahrer	<code>highway=cycleway</code>	Radweg
Fußgänger	<code>highway=footway</code>	Speziell ausgezeichnete Fußweg
i.A. keine Einschränkung	<code>highway={primary, secondary, tertiary, ...} + footway=sidewalk</code>	Innerstädtische Straßen mit Bürgersteig

Tabelle 3 - Auswahl an OSM-Tags und deren bestimmte Nutzung

Wie in Abschnitt 3.1.3 erwähnt, werden den Kanten eines Graphen Gewichtungen zugeordnet. Für Fußgänger beispielsweise werden Gewichtungen abhängig von der Gehgeschwindigkeit und der Länge der Kantenabschnitte vergeben. Damit spielt die Entfernung eine entscheidende Rolle bei der Berechnung des kürzesten Weges. Im Falle des MIV stellt die maximal zulässige Höchstgeschwindigkeit der Straßenabschnitte ein entscheidendes Kriterium dar. Ein Weg durch eine verwinkelte Innenstadt mag zwar der kürzeste sein, ein Umweg über die Autobahn kann aber weniger Zeit in Anspruch nehmen, weil er schneller befahren werden kann. Geschwindigkeitsbeschränkungen, falls sie von den vorgegebenen Richtwerten abweichen, werden mit dem tag

`maxspeed=x km/h`
gekennzeichnet.

Beispielhaft wird das Prinzip der Gewichtung am Transportmodus *Fahrrad* erläutert. Ist im Konfigurationsdreieck die Option *fahrradfreundlich/sicher* gewählt, gilt für unterschiedlich gekennzeichnete Streckenabschnitte ein Multiplikator w mit $w > 0$, wobei $w < 1$ für den Router kürzer und $w > 1$ länger

OSM-Tag	Entsprechung	Multiplikator
highway=cycleway	Fahrradweg	0.60
highway=footway	Speziell ausgezeichnete Fußweg	1.10
highway=track	Feld- oder Waldweg	1.30
highway=motorway	Autobahn	8.00
highway=* + cycleway=lane	Straßen mit separater Fahrradspur	0.87
highway=track + surface=gravel	Feld- oder Waldweg mit Schotteroberfläche	1.50
highway=track + surface=sand	Feld- oder Waldweg mit Sandoberfläche	100

Tabelle 4 - OSM-Tags und deren Gewichtungsmultiplikator

erscheint. Abhängig davon, wie detailliert das Straßennetz durch die Nutzer von OSM attribuiert wird, steigert sich die Qualität des Routingergebnisses. Dabei kann auch das Material der befahrenen Oberfläche eine Rolle spielen. Tabelle 4 listet eine Auswahl von OSM-Tags und deren Gewichtungsmultiplikator auf.

5 UMSETZUNG

Für die Umsetzung der prototypischen Implementierung eines multimodalen Routenplaners werden alle Daten und Softwarepakete zusammengeführt. Die verwendeten Daten müssen dabei unter Umständen aufbereitet und der Arbeitsumgebung entsprechend angepasst werden. Der Fokus liegt auf der Umsetzung aller in Abschnitt 3.1 beschriebenen Funktionen.

5.1 VERWENDETE SOFTWARE

Das sich in Entwicklung befindliche Projekt *OpenTripPlanner* wird für die Umsetzung des multimodalen Routenplaners verwendet. OTP wird über die Quellcode-Versionierungsplattform GitHub verwaltet. Hier sind alle seit Beginn der Entwicklung veröffentlichten Programmversionen in sogenannten *branches* verfügbar. Im Januar 2015 stellt der *master-branch* in der Version 0.13.0 die aktuellste Iteration des OTP dar. Für die Umsetzung wird auf die ältere Version 0.9.2 zurückgegriffen. Im Abschnitt Ausblick 7.2 werden beide Versionen kurz verglichen, hauptsächlich neuere Funktionalitäten betreffend. OTP ist zu gut 85% in Java, 13% JavaScript und ca. 1% andere Markup- und Skriptsprachen (u.a. HTML, CSS, Python, XML) geschrieben (Abbildung 13).

Die Verwaltung des umfangreichen Quellcodes von OTP erfolgt über die integrierte Entwicklungsumgebung (integrated development environment, IDE) *Eclipse*²³. Diese ist ein quelloffenes Programmierwerkzeug. Ein Vorteil der

²³ <https://eclipse.org/juno/>

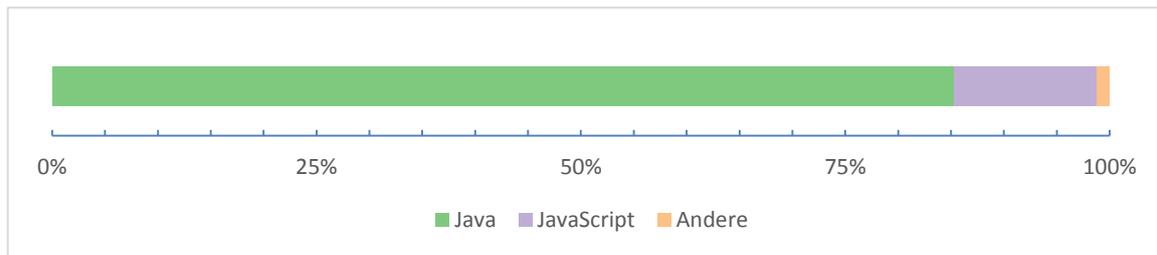


Abbildung 13 - Anteile der Programmiersprachen von OTP

Nutzung einer quelloffenen Entwicklungsumgebung wie Eclipse liegt unter anderem in der Erweiterbarkeit. Sie erlaubt zum Beispiel das Einrichten verschiedener Laufzeitumgebungen oder virtueller Server, ohne auf externe Software zurückgreifen zu müssen. Ursprünglich ist Eclipse auf die Entwicklung von Java-Programmen ausgerichtet. Durch zusätzliche Programmpakete und Plug-Ins ist aber auch die Arbeit u.a. an hybriden Webprojekten wie dem OTP möglich.

Für die Veröffentlichung des Projekts im Internet wird eine Version des *Apache Tomcat*²⁴ Webservers benötigt. Dabei handelt es sich ebenfalls um ein quelloffenes Tool, welches es erlaubt, auf Java basierende Webanwendungen (*servlets*) auszuführen.

Für die Aufbereitung der Daten kommen eine Reihe weiterer freier Standardprogramme und -tools zum Einsatz, darunter *GDAL*²⁵, *JOSM*²⁶, *LibreOffice*²⁷ und *Notepad++*²⁸.

5.1.1 STRUKTUR OPENTRIPPLANNER

In der verwendeten Version 0.9.2 setzt sich OTP aus verschiedenen Modulen zusammen, die verschiedene Funktionen erfüllen. Tabelle 5 enthält eine Übersicht aller Module und deren Funktion.

²⁴ <http://tomcat.apache.org/>

²⁵ <http://www.gdal.org/>

²⁶ <https://josm.openstreetmap.de/>

²⁷ <https://de.libreoffice.org/>

²⁸ <http://notepad-plus-plus.org/>

Modul	Position	Funktion
Admin Client	Backend	Administration und Konfiguration
Analyst Client	Frontend	Netzwerkanalysefunktionen (Fahrzeiten, Erreichbarkeit, ...)
Core	Backend	Hauptfunktionen (Routing-Engine, Graph-Builder, ...)
Geocoder	Backend	Modul für Geokodierung
Leaflet Client	Frontend	Graphische Benutzeroberfläche mit Webmap-Bibliothek <i>leaflet.js</i>
OpenLayers Client	Frontend	Graphische Benutzeroberfläche mit Webmap-Bibliothek <i>OpenLayers.js</i> (veraltet)
REST API Servlet	Frontend	Modul für REST API zur Nutzung von OTP für dritte Programme

Tabelle 5 - Modul- und Funktionsübersicht OpenTripPlanner

OTP ist in der verwendeten Version 0.9.2 mit dem Build-Management-Tool *Apache Maven* ²⁹ programmiert. Das bedeutet, dass ein Großteil der Konfiguration der Software über standardisiert formatierte XML-Dateien mit den Dateinamen `pom.xml` (**Project Object Model**) erfolgen kann (siehe dazu auch Abbildung 14). Dies vereinfacht überwiegend die Arbeit mit OTP. In der aktuellsten Version wird komplett auf die Maven-Struktur verzichtet.

5.2 EINRICHTUNG DER ARBEITSUMGEBUNG

Um sinnvoll Veränderungen an OpenTripPlanner und dessen einzelnen Komponenten vornehmen zu können, ist eine Einrichtung des Quellcodes in einer integrierten Entwicklungsumgebung notwendig. Aus den in Abschnitt 5.1 genannten Gründen, eignet sich dafür *Eclipse*. Hier lässt sich auch mittels virtueller Server das Verhalten von OTP testen. Eine Reihe von Plug-Ins müssen installiert werden, damit die Bearbeitung ohne Probleme verläuft:

²⁹ <https://maven.apache.org/>

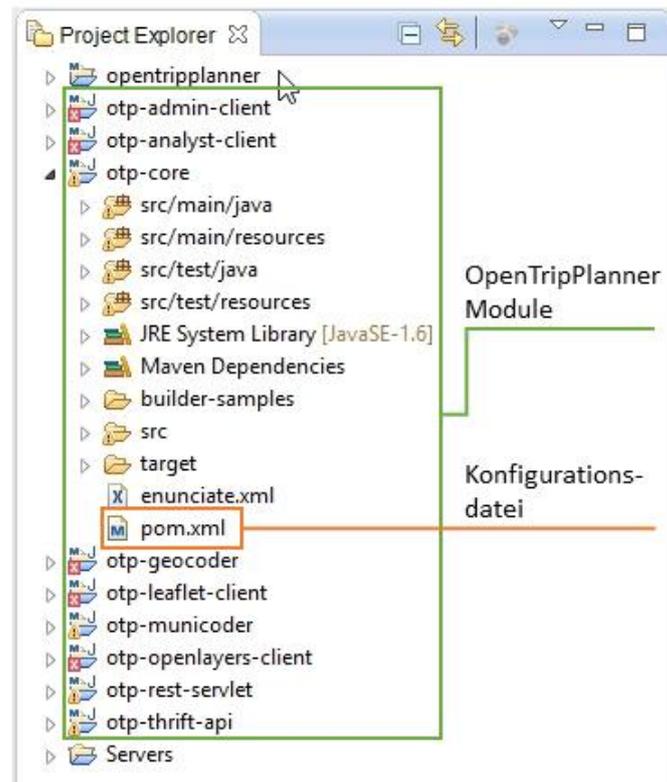


Abbildung 14 - Übersicht aller OpenTripPlanner Module in Eclipse

- EGit³⁰, erlaubt die Verwendung von Git als Versionierungssystem
- M2eclipse³¹, integriert das Build-Management-Tool *Apache Maven* in Eclipse
- Eclipse Web Tools Platform³², erweitert Eclipse um Werkzeuge zur Entwicklung von Webanwendungen

Der Quellcode von OTP wird über die Git-Shell heruntergeladen. Der Befehl `$ git clone -b 0.9.2 https://github.com/openplans/OpenTripPlanner.git` lädt eine Kopie des *branches* mit der Versionsnummer 0.9.2 herunter. Das Projektverzeichnis kann anschließend als Mavenprojekt nach Eclipse importiert werden.

³⁰ <https://eclipse.org/egit/>

³¹ <https://eclipse.org/m2e/>

³² <https://eclipse.org/webtools/>

5.3 AUFBEREITUNG DER DATEN

Einige der Eingangsdaten müssen als Vorbereitung auf die Umsetzung aufbereitet werden. Dazu zählen die OpenStreetMap-Daten, die speziell im Bereich der drei Campuskomplexe *Neues Palais*, *Griebnitzsee* und *Golm* zum Zwecke der Fußgängernavigation aufbereitet werden müssen. Außerdem ist eine Liste mit universitären Einrichtungen zu erstellen, die Gebäudenamen, -Funktion und Position enthält, welche später für das Geocoding zum Einsatz kommen wird.

5.3.1 OPENSTREETMAP

OpenTripPlanner nutzt Kartenmaterial von OSM als Navigationsgrundlage für Fußgänger, Radfahrer und den motorisierten Individualverkehr. Es greift dabei, wie andere auf OSM basierende Routingapplikationen, sowohl auf die Geometrie (Kantenlänge), als auch auf die Attributierung zurück. Da die Datenqualität vom Arbeitsaufwand der Nutzer abhängt, müssen die Daten unter Umständen dem eigentlichen Nutzungszweck entsprechend angepasst werden. Dieses Qualitätsmerkmal lässt unter der Bezeichnung *fitness for use*, also Gebrauchstauglichkeit, zusammenfassen (CAPRIOLI, TARANTINO, 2003). In diesem Falle ist der Verwendungszweck die Navigation. Beispielhaft werden für die drei Universitätsstandorte Golm, Neues Palais und Griebnitzsee auf qualitative Art und Weise Mängel auf geometrischer und attributiver Ebene behoben (siehe Abbildung 15).

Die Firma Geofabrik³³ bietet aktuelle, auf Regionen zugeschnittene Datensätze aus dem globalen *OSM World File* an. Potsdam, im Speziellen, ist nicht vorhanden, aber das Bundesland Brandenburg. Daraus lässt sich mit Hilfe des Werkzeugs *Osmosis*³⁴ ein Ausschnitt für Potsdam auf Basis einer Bounding Box generieren. Dieser Schritt der Reduktion ist notwendig, um die Datenmenge und damit verbundene Rechenzeiten zur Generierung der Routen gering zu halten.

³³ <http://download.geofabrik.de/>

³⁴ <http://wiki.openstreetmap.org/wiki/Osmosis>

Osmosis ist ein Kommandozeilentool. Der Befehl zum Ausschneiden anhand der Bounding Box lautet wie folgt:

```
1 | osmosis --read-xml enableDateParsing=no file="C:\workspace\osmosis\data\brandenburg-  
   | latest.osm" --bounding-box top=52.4313 left=12.9558 bottom=52.34763 right=13.16839 --  
   | write-xml "C:\workspace\osmosis\data\potsdam.osm"
```

Quellcode 1 - Befehl zum Croppen eines Ausschnitts aus OSM mit Osmosis anhand einer Bounding Box

OTP bietet die Möglichkeit, Daten von OSM basierend auf der Ausdehnung des eingegebenen GTFIS-Datensatzes automatisch herunterzuladen. Allerdings ist diese Funktion, ähnlich wie der direkte Download aus dem Browser heraus von www.openstreetmap.org, auf eine maximale Anzahl von Nodes beschränkt. Aus diesem Grund wird auf Osmosis zurückgegriffen.

Das Werkzeug *Java OpenStreetMap Editor* (JOSM) bietet die Möglichkeit, lokal vorhandene OSM-Dateien zu bearbeiten, ohne an den globalen OSM-Daten Veränderungen vorzunehmen. Dies ist insofern nützlich, als dass bestimmte Änderungen am Kartenmaterial erst nach Überprüfung durch andere Nutzer in das System von OSM übertragen werden. Für die unmittelbare Bearbeitung und weitere Nutzung eignet sich daher die Nutzung von JOSM. Diese Datenaufbereitung stellt keinen Anspruch auf Vollständigkeit, sondern dient lediglich zum Aufzeigen der Möglichkeit einer Nachbearbeitung der frei zugänglichen OSM-Daten. Im Sinne der Qualitätswahrung wird dieser Vorgang in regelmäßigen Abständen vom Betreiber des Routenplaners durchgeführt, um durchgehend hochwertige Routinganweisungen gewährleisten zu können.

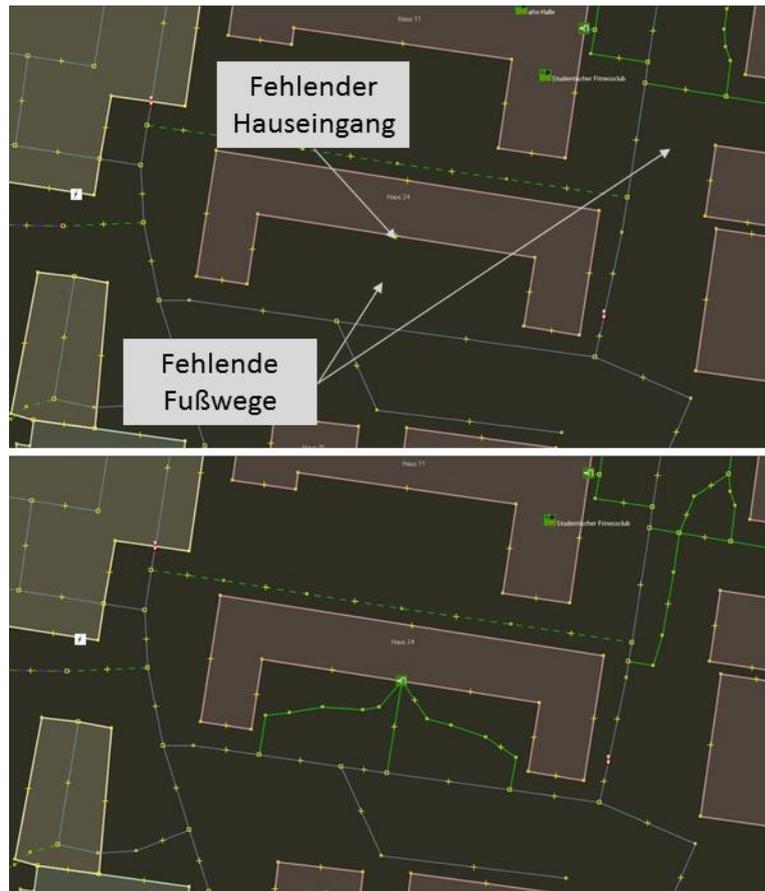


Abbildung 15 - Unbearbeitete (oben) und nachbearbeitete (unten) OSM-Daten am Beispiel Haus 24, Institut für Geographie, Golm

5.3.2 GEOCODER

Ein Geocoder ist ein Werkzeug, mit dessen Hilfe bezugsfreie Adress- oder Namensangaben in geographische Koordinaten umgewandelt werden können. Für den OpenTripPlanner werden zwei Geocoder eingerichtet:

- Nominatim³⁵, ein freier Geocoder, basierend auf den Daten von OSM
- Ein alternativer Geocoder mit einer festen Liste der Universitätseinrichtungen

Nominatim greift auf die Adressdaten zu, welche in OSM hinterlegt sind. Da OSM als Datengrundlage für OTP genutzt wird, eignet sich dieser Geocoder im

³⁵ <https://nominatim.openstreetmap.org/>

Besonderen zur Verwendung. Damit OTP den Geocoder ansprechen kann, muss das entsprechende Java-Servlet dem Server zur Verfügung gestellt werden.

Nominatim ist standardmäßig als Geocoder konfiguriert. OSM enthält aber nicht alle Informationen über Universitätsstandorte, die notwendig sind, um eine lückenlose Haus-zu-Haus-Navigation zu gewährleisten. Besonders Gebäude mit doppelter Belegung, z.B. durch zwei Institute, lassen sich in den *tags*, welche OSM verwendet, nur mäßig hinterlegen. Aus diesem Grund kommt ein zweiter, alternativer Geocoder zum Einsatz, der auf eine feste Liste universitärer Einrichtungen zurückgreift. Diese Erweiterung stammt aus einem *Fork* von BARBEAU (2012). Ein *Fork* ist eine Abspaltung eines auf GitHub bereits existierenden Projektes, in diesem Fall OTP, das anschließend verändert werden kann, ohne Einfluss auf das Original zu nehmen. Die statische Liste wird in die Datei `application_context.xml` im Geocoder-Modul nach dem Schema von Quellcode 2 integriert. Position, Namen und Funktion aller Gebäude basieren auf den Lageplänen der Universitätshomepage. Die vollständige Liste aller insgesamt 144 Einrichtungen findet sich in Anhang 9.1.

```

1 | <bean id="geocoderManual"
  |   class="org.opentripplanner.geocoder.
  |   manual.ManualGeocoder">
2 |   <property name="pois">
3 |     <map>
4 |       <entry key="[Einrichtung A]" value="[lat] [long]" />
5 |       <entry key="[Einrichtung B]" value="[lat] [long]" />
6 |       <entry key="[Einrichtung C]" value="[lat] [long]" />
7 |       <entry key="[Einrichtung D]" value="[lat] [long]" />
8 |       <!-- cont... -->
9 |     </map>
10 |   </property>
11 | </bean>

```

Quellcode 2 - XML-Schema für die statische Liste universitärer Einrichtungen

Anschließend lassen sich beide Geocoder ebenfalls über die Datei `application_context.xml` einbinden. OTP nennt das einen *alternierenden Service* (Quellcode 3). Gibt der Benutzer im Suchfeld für Start und Ziel einen Suchbegriff ein, wird zunächst die statische Liste durchsucht. Sollte darin kein Ergebnis gefunden werden, wird auf Nominatim und damit auf die Daten von OSM zurückgegriffen.

```

1 | <bean id="geocoderAlternatingService"
   |   class="org.opentripplanner.geocoder.
   |   AlternatingGeocoderModified">
2 |     <!-- 1. Geocoder -->
3 |     <constructor-arg ref="geocoderManual" />
4 |     <!-- 2. Geocoder -->
5 |     <constructor-arg ref="geocoderNominatim" />
6 |     <constructor-arg value="true" />
7 | </bean>

```

Quellcode 3 - Einbinden zweier alternierender Geocoder

Durch Konfiguration der Benutzeroberfläche mittels der Datei `config.js` wird der Geocoding-Service eingebunden (siehe dazu Quellcode 4).

```

1 |   geocoders : [
2 |     {
3 |       'name': 'geocoder',
4 |       'className': 'otp.core.Geocoder',
5 |       'url': 'http://141.89.192.240/otp-geocoder/geocode',
6 |       'addressParam': 'address'
7 |     }
8 |   ]

```

Quellcode 4 - Konfiguration des Geocoders in der Datei `config.js`

Da es sich bei dem Geocoder um ein eigenes Modul handelt, das unabhängig von der eigentlichen graphischen Benutzeroberfläche und der Routing-Engine arbeitet, lässt es sich auch als Web-Service mittels REST-Anfragen durch dritte Programme nutzen.

5.3.3 DIGITALES GELÄNDEMOMODELL

Für Radfahrer spielt die Berechnung höhenabhängiger Routen, um eine möglichst flache Strecken zu erhalten, eine Rolle. Damit stellen Höheninformationen eine weitere wichtige Komponente für den Graphen dar. Aus diesem Grund wird auf ein digitales Geländemodell (DGM) der Stadt Potsdam zurückgegriffen. Es handelt sich dabei um ein auf LiDAR-Messungen beruhendes Geländemodell mit einer vertikalen geometrischen Auflösung von 0,15m und 1m im horizontalen Bereich. Das DGM ist nicht frei verfügbar, sondern wird auf Anfrage zu Testzwecken für die Verwendung in dieser Arbeit durch die Stadt Potsdam zur Verfügung gestellt. Als alternative Quelle für Höheninformationen lassen sich die Fernerkundungsdaten der *Shuttle Radar Topography Mission* (SRTM) aus dem Jahr 2000 heranziehen. Deren geometrische Auflösung von 90m für den europäischen Raum wird für den vorliegenden Maßstab aber als zu gering eingeschätzt.

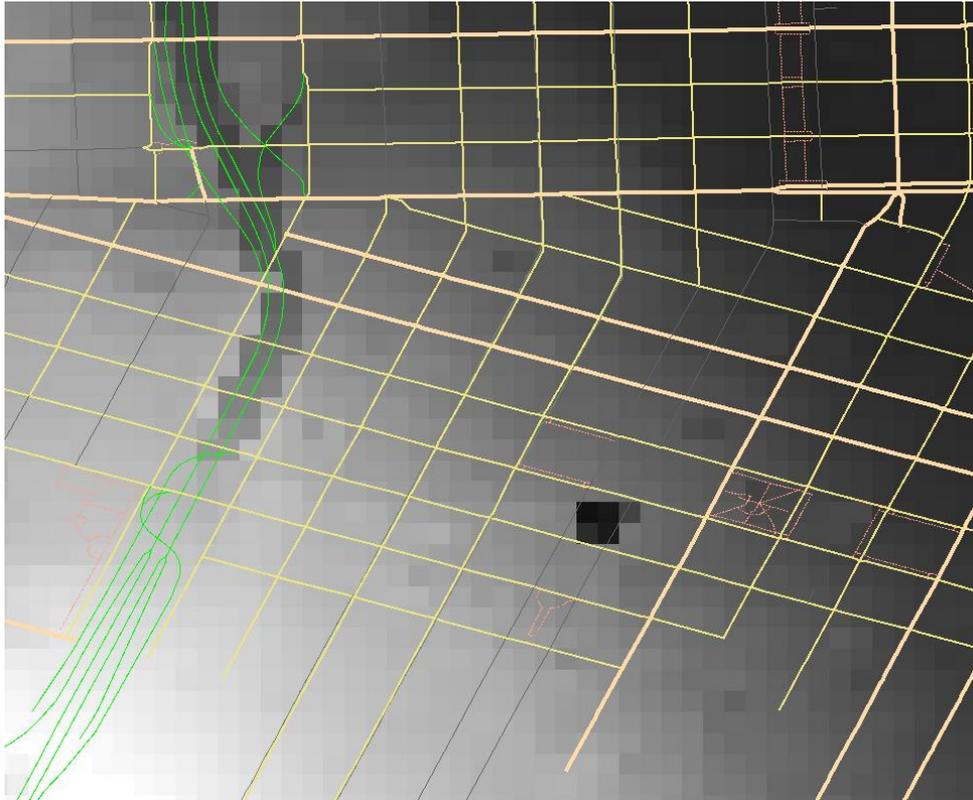


Abbildung 16 - Ein Graph überlagert das DGM (BYRD, 2011)

Die Verbindung aus Graph und DGM basiert darauf, dass die Knoten auf die Rasterzellen des Geländemodells gelegt werden. Jedem einzelnen Knoten wird daraufhin der zu Grunde liegende Wert aus der entsprechenden Rasterzelle zugeordnet. Zum einen lassen sich so die Höheninformationen in den Knoten speichern. Zum anderen wird aus den Werten zweier benachbarter Knoten der Anstieg entlang einer Kante berechnet.

Das DGM liegt im UTM-Koordinatensystem mit dem Referenzsystem ETRS89 vor. Damit OpenTripPlanner den Datensatz verwenden kann, muss das Raster in das Koordinatensystem WGS 84 transformiert werden. Die freie Programmbibliothek *Geospatial Data Abstraction Library* (GDAL) bietet dafür die Komponente *gdalwarp*. Der Befehl für die Transformation lautet:

```
gdalwarp -s_srs EPSG:25833 -t_srs EPSG:4326 dgm1roh.tif dgmPotsdamWGS84.tif
```

Damit kann das DGM für die weitere Verarbeitung verwendet werden.

5.4 ROUTINGFÄHIGES NETZWERK ERZEUGEN

Der Graph, also das routingfähige Netzwerk, stellt die Grundlage für Start-Ziel-Analysen dar. Es setzt sich zusammen, zum einen aus dem Verkehrsnetz, welches aus OpenStreetMap hervorgeht, und zum anderen aus den Nahverkehrsdaten in Form von GTFS. Höheninformationen erhalten die Knotenpunkte aus einem digitalen Geländemodell.

OpenTripPlanner bietet die Möglichkeit, die Datensätze zusammenzuführen. Die Java-Anwendung *Graph-Builder* liegt im OTP-Modul *Core*. Durch Konfiguration einer entsprechenden XML-Datei `graph-config.xml` lässt sich neben den Datenquellen auch die Routingeigenschaften verschiedener Straßen- und Wegetypen anpassen. Die Standardeinstellungen für die Nutzungsrechte von Kanten durch die verschiedenen Transportmodi liegen unter

`org.opentripplanner.graph_builder.impl.osm.DefaultWayPropertySetSource`.

Hier wird u.a. festgelegt, welche Kanten, mit welchen OSM-Tags durch welche Verkehrsteilnehmer genutzt werden dürfen. Dies ist wichtig für den Routingalgorithmus, um vernünftige Ergebnisse zu liefern. Es werden weiterhin Gewichtungen für spezielle Kanten/Modus-Kombinationen vergeben (siehe 4.3.2) und Richtgeschwindigkeiten festgelegt. Man hat die Wahl, diese Standardeinstellungen nach seinen Vorstellungen, bzw. den geltenden Gesetzen der Straßenverkehrsordnung, anzupassen, oder einzelne Veränderungen in die Datei `graph-config.xml` einzupflegen.

Wichtig ist hier vor allem für den Transportmodus MIV die korrekten Richtgeschwindigkeiten anzugeben, damit exakte Reisezeiten berechnet werden können. Standardmäßig beruhen diese auf den Vorgaben aus den USA.

Zusammenfassend werden die drei Datenquellen für den *Graph-Builder* in der Datei `graph-config.xml` festgelegt:

- GTFS
- OSM
- DGM

Quellcode 5 zeigt beispielhaft einen Ausschnitt der Konfiguration, in diesem Fall, wie die GTFS-Daten definiert werden. Die komplette Datei findet sich im Anhang 9.2.

```

1      <bean id="gtfsBuilder"
      class="org.opentripplanner.graph_builder.
      impl.GtfsGraphBuilderImpl">
2          <property name="gtfsBundles">
3              <bean id="gtfsBundles"
      class="org.opentripplanner.graph_builder.
      model.GtfsBundles">
4                  <property name="bundles">
5                      <list>
6                          <bean
      class="org.opentripplanner.graph_builder.
      model.GtfsBundle">
7                              <property name="path"
      value="C:\Master\otp-data\gtfs\
      20641.zip" />
8                              <property name="defaultBikesAllowed"
      value="true" />
9                          </bean>
10                     </list>
11                 </property>
12             </bean>
13         </property>
14     </bean>

```

Quellcode 5 - Ausschnitt aus der Datei *graph-config.xml*. Konfiguration der GTFS-Daten

Graph-Builder ist eine Klasse im Modul *Core* von OTP:

```
org.opentripplanner.graph_builder.GraphBuilderMain
```

Sie kann als Java-Applikation ausgeführt werden, greift auf die Konfigurationsdatei zurück, und erzeugt einen routingfähigen Graph, welcher von der Routingengine während der Suche von Strecken abgerufen wird. Der Graph sollte in regelmäßigen Abständen aktualisiert werden, zumindest aber wenn neue Fahrplandaten verfügbar sind. Dazu sollte ebenfalls eine aktuelle Version des OSM-Ausschnitts verwendet werden.

5.5 SERVERSEITIGE IMPLEMENTIERUNG

Mit der Konfiguration aller notwendigen Dateien und der Erzeugung des *Graph*-Objekts ist die serverseitige Implementierung vorbereitet. Der Quellcode wird durch Maven mittels des Befehls

```
clean package -DskipTests
```

kompiliert. Die einzelnen Module (Admin Client, Analyst Client, Geocoder, usw.) werden damit als Servlets bereitgestellt, die über Apache Tomcat geladen werden.

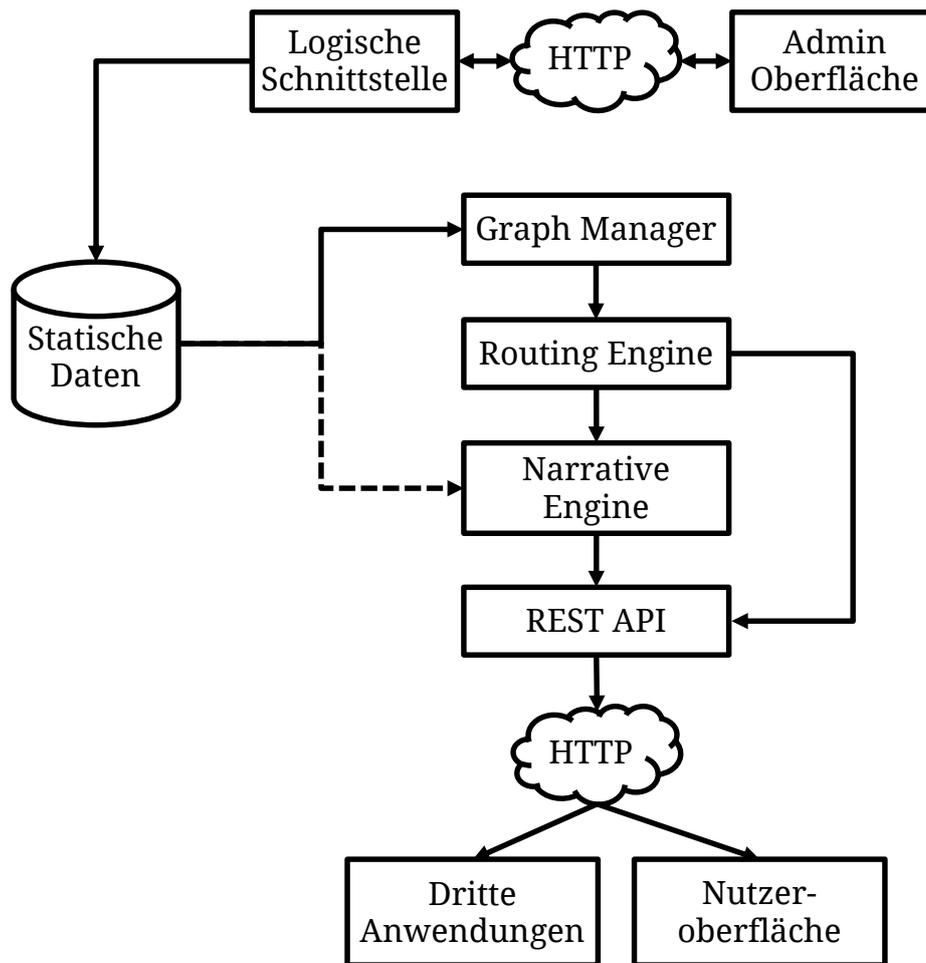


Abbildung 17 - High-Level-Architektur von OpenTripPlanner nach BERGSON-SHILCOCK (2011)

In Form einer Webanwendung spiegelt die Softwarearchitektur von OpenTripPlanner (OTP) den modularen Aufbau, beschrieben in Abschnitt 5.1.1, wider. Abbildung 17 zeigt die High-Level-Architektur.

Ein oder mehrere GTFS-Datensätze, OSM und das DGM stellen die statischen Daten dar. Eine Auslagerung der Daten auf den Server liegt im momentan nicht vor, ist aber denkbar. Über Skripte ließe sich das Abrufen der Daten und die anschließende Generierung des *Graph*-Objekts über den *Graph Manager* automatisieren.

Die narrative Engine ist verantwortlich für die an den Nutzer ausgegebenen Richtungsanweisungen. Informationen erhält sie indirekt aus den statischen Daten über Straßennamen, Nahverkehrsverbindungen, usw.

Die Nutzeroberfläche dient als Visualisierungskomponente der Routingergebnisse und greift auf die Entwicklerschnittstelle zu. Dritte Anwendungen, z.B. mobile Applikationen für Smartphone oder Systeme für grafikfreie Navigationsangaben können ebenfalls auf diese Schnittstelle zugreifen.

Testweise werden die Servlets auf einem virtuellen Server mit folgender Konfiguration verteilt:

- Hardware:
 - 4 physische, bzw. 8 virtuelle Prozessorkerne
 - 16 GB RAM
 - 50 GB HDD
- Software:
 - Betriebssystem Linux 64 Bit (Ubuntu 14.04 LTS)
 - Tomcat 7 + Apache
 - Java RE 7

Der *Tomcat Web Application Manager* steuert die verfügbaren Servlets. Über die Administrationsoberfläche lassen sich Module laden, starten oder beenden. Anhand der verschiedenen URL-Pfade werden die einzelnen Servlets angesprochen:

- <http://hostname...>
 - `/otp-geocoder` – Geocoder-Modul
 - `/otp-analyst-client` – Netzwerkanalyse-Modul
 - `/otp-rest-servlet` – Nutzerspezifische graphische Oberfläche
 - usw.

Nach der Bearbeitung aller benötigten Daten, ist eine Veröffentlichung auf einem Webserver möglich. Damit wird dem Nutzer der Zugriff auf den multimodalen Routenplaner ermöglicht. Der Zugriff sollte dabei auf die graphische Benutzeroberfläche beschränkt werden. Die Administrationsoberfläche und weitere Module der Software bleiben zu Verwaltungszwecken dem Host vorbehalten.

GET Parameter	Beschreibung	Standard
fromPlace	Startpunkt, entweder als Paar aus geographischer Breite und Länge in Grad oder ein Name aus der Liste des Geocoders	
toPlace	Zielpunkt, vgl. fromPlace	
date	Datum für die Routenanfrage im Format MM-DD-YYYY	
time	Die Abfahrtszeit für die Route im Format HH:mm	
mode	Ein oder mehrere Transportmodi, die genutzt werden sollen	TRANSIT,WALK
optimize	Charakteristiken, nach denen die Route verbessert werden soll	QUICK

Tabelle 6 - Beschreibung der API Parameter für eine beispielhafte Anfrage

OpenTripPlanner bietet die Möglichkeit, über eine REST-API Anfragen an die Routing-Engine, die Netzwerkanalyse und den Geocoder zu stellen. Die Anfrage für eine **flache Fahrradstrecke** von der **Golmer Universitätsbibliothek IKMZ** zum **Sportplatz am neuen Palais**, am **14.02.2015** um **10 Uhr** lautet z.B.:

```
http://hostname/otp-rest-servlet/ws/plan?
fromPlace=52.40871,12.97292&
toPlace=52.40157,13.00755&
date=03-25-2015&
time=10:00&
mode=BICYCLE&
optimize=FLAT
```

Tabelle 6 beschreibt die einzelnen verwendeten Parameter, gegebenenfalls deren Format und Standardwerte.

Die Antwort auf diese Anfrage erfolgt als XML. Ähnlich funktionieren Ansprachen an den Geocoder. Die URL

```
http://hostname/otp-geocoder/geocode?address=BIBLIOTHEK
```

liefert beispielweise die geographischen Koordinaten aller universitären Einrichtungen, die *Bibliothek* im Namen haben (siehe Quellcode 6). Eine ausführliche Beschreibung aller verfügbaren Parameter, sowie das verwendete XML-Antwortschema findet sich bei GENDRE (2014).

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <geocoderResults>
3   <count>4</count>
4   <results>
5     <result>
6       <description>GOLM, HAUS 18, UNIVERSITÄTSBIBLIOTHEK
7         (IKMZ)</description>
8       <lat>52.408712</lat>
9       <lng>12.972918</lng>
10    </result>
11   <result>
12     <description>NEUES PALAIS, HAUS 11,
13       UNIVERSITÄTSBIBLIOTHEK</description>
14     <lat>52.401877</lat>
15     <lng>13.012558</lng>
16   </result>
17   <result>
18     <description>NEUES PALAIS, HAUS 10,
19       UNIVERSITÄTSBIBLIOTHEK</description>
20     <lat>52.40026</lat>
21     <lng>13.012709</lng>
22   </result>
23   <result>
24     <description>GRIEBNITZSEE, HAUS 5,
25       UNIVERSITÄTSBIBLIOTHEK</description>
26     <lat>52.39395</lat>
27     <lng>13.130785</lng>
28   </result>
29 </results>
30 </geocoderResults>
```

Quellcode 6 - XML-Response auf eine Anfrage an den Geocoder mit dem Begriff Bibliothek

6 LIZENZEN

Die Verwendung freier Daten, freier Software und Software mit offenem Quellcode, geht zwangsläufig einher mit den damit verbundenen Regelungen zur Benutzung, Veränderung und Weitergabe. Diese sind in verschiedenen Lizenzmodellen festgehalten. Im Folgenden wird dargelegt, unter welchen Lizenzen die verwendeten Daten und Software verfügbar sind und mit welcher Lizenz das Projekt Campus Routenplaner veröffentlicht werden kann.

6.1 OPENSTREETMAP

Das Projekt OpenStreetMap (OSM) entwickelt sich mit dem Gedanken, eine global verfügbare, frei zugängliche Quelle von Geodaten zu sammeln, die jeder verwenden, verändern und weitergeben darf.

Mit der Gründung im Jahr 2006 werden die Daten unter der Lizenz *Creative Commons – Weitergabe unter gleichen Bedingungen* (CC BY-SA) in der Version 2.0 veröffentlicht. Das heißt:

- Die Daten dürfen kopiert werden.
- Die Daten dürfen verändert werden.
- Bei einer Verwendung der Daten in einem eigenen Projekt, muss dieses unter der gleichen Lizenz veröffentlicht werden.
- Bei einer Verwendung der Daten in einem eigenen Projekt, müssen die Urheber gekennzeichnet werden (CREATIVE COMMONS CORPORATION, 2014b).

Zu diesem Zeitpunkt stellt diese Lizenz die einzige Möglichkeit dar, die gesammelten Daten zu schützen. Es handelt sich dabei allerdings um eine Methode, kreatives Eigentum, wie Musik, Kunst oder Literatur, urheberrechtlich zu schützen. CC BY-SA ist für den Fall von faktischen, realen Geodaten, wie sie OSM sammelt, ungeeignet. Folglich stellt das bisherige Lizenzmodell von OSM einen entscheidenden Kritikpunkt dar, und die Daten sind für eine eventuelle Übernahme ohne Kennzeichnung ungeschützt (OPENSTREETMAP FOUNDATION, 2008).

Mit der Verbreitung von freien Daten haben sich neue Lizenzmodelle entwickelt, die für die Zwecke des OpenStreetMap-Projekts besser geeignet sind und größtenteils global anerkannt werden. Mit der Umstellung im Jahr 2012 gilt OSM inzwischen als *Open Data* gemäß der *Open Data Commons Open Database Lizenz* (ODbL) (OPENSTREETMAP FOUNDATION, 2012). Diese spezielle Lizenz ist direkt auf Daten zugeschnitten. Nach der Open Knowledge Foundation (2015) erlaubt sie:

- Das Kopieren, die Weitergabe und Verwendung der Daten.
- Das Erzeugen neuer Inhalte auf Grundlage der Daten.
- Die Veränderung der Daten.

Dazu müssen folgende Bedingungen erfüllt werden:

- Die ursprünglichen Urheber (OSM und Mitwirkende) müssen als Quelle kenntlich gemacht werden.
- Die Verwendung von modifizierten Daten verlangt die freie Weitergabe der Daten unter der gleichen Lizenz.
- Die modifizierten Daten können durch technische Maßnahmen, z.B. Kopierschutz, gesichert werden, solange eine offen zugängliche Variante zur Verfügung steht.

Diese Lizenzumstellung betrifft lediglich die Rohdaten. Die Verwendung von kartographisch aufbereiteten Kacheln, z.B. in Form einer Webkarte, erfolgt weiterhin unter *Creative Commons – Weitergabe unter gleichen Bedingungen*.

	CC BY	CC BY-SA
Die Daten dürfen kopiert werden.	✓	✓
Die Daten dürfen verändert werden.	✓	✓
Bei einer Verwendung der Daten in einem eigenen Projekt, müssen die Urheber gekennzeichnet werden	✓	✓
Bei einer Verwendung der Daten in einem eigenen Projekt, muss dieses unter der gleichen Lizenz veröffentlicht werden	x	✓

Tabelle 7 - Vergleich der Lizenzmodelle CC BY und CC BY-SA

6.2 GENERAL TRANSIT FEED SPECIFICATION

Die Transitdaten in Form von General Transit Feed Specifications (GTFS) werden über das Datenportal der Stadt Berlin frei zum Download angeboten. Das Projekt bietet den Zugriff als „Einstiegspunkt auf Daten und Informationen der Berliner Verwaltung [...] so dass durch neue Ideen sowie Kombination und Analyse neue Erkenntnisse aus den vorhandenen Daten gewonnen werden können“ (BERLINONLINE STADTPORTAL GMBH & Co. KG, 2015). Neben den Verkehrsdaten werden hier unter anderem Datenquellen und Zugriffspunkte zu verschiedenen sozialen und politischen Themen, wie Bildung, öffentliche Verwaltung oder Tourismus, angeboten. Die meisten der hier zur Verfügung gestellten Informationen liegen unter der Lizenz *Creative Commons – Weitergabe unter gleichen Bedingungen* (CC BY-SA) vor, die im Abschnitt 6.1 erläutert wird.

Die GTFS-Daten, bereitgestellt durch den Verkehrsverbund Berlin-Brandenburg (VBB) liegen unter der Lizenz *Creative Commons – Namensnennung* (CC BY) vor, ein leicht abgeschwächten Version der CC BY-SA. Beide Bestimmungen gehen weitestgehend konform, lediglich auf die Bedingung der Weitergabe der verwendeten Daten unter gleicher Lizenz verzichtet der Urheber. Damit eignen sich die GTFS-Daten zur Weiterverwendung z.B. in kommerziellen Projekten. Tabelle 7 stellt beide Lizenzmodelle gegenüber.

6.3 OPENTRIPPLANNER

OpenTripPlanner (OTP) als freie Software wird unter der *GNU Lesser General Public License* (LGPL) in der Version 3 zur Verfügung gestellt (BYRD, 2013). Diese von der *Free Software Foundation* konzipierte Lizenz erlaubt das Verwenden von LGPL-Software in eigener Software, stellt dabei aber frei, den Quellcode der eigenen Software nicht freizugeben. Einzig die Änderungen an den Programmteilen unter LGPL muss dem Nutzer nachvollziehbar offen gelegt werden. Weiterhin muss der Nutzer in der Lage sein, Änderungen an diesem Programmcode vorzunehmen, z.B. in Form von ausgelagerten Programmbibliotheken, die von der eigenen Software losgelöst vorliegen (FREE SOFTWARE FOUNDATION INC., 2007; GERWINSKI, 2007). Prinzipiell wäre damit z.B. die Nutzung der in OTP implementierten Routingalgorithmen auch in Form einer proprietären Software möglich, solange die entsprechenden Programmabschnitte weiterhin frei verfügbar bleiben.

Es ist wichtig zu bemerken, dass, obwohl OTP und dessen Quellcode, für jeden Nutzer frei zum Download, zur Bearbeitung und Weiterverwendung zur Verfügung stehen, es sich nicht um Software im Sinne von *Public Domain* handelt. Auf das Urheberrecht wird nicht gänzlich verzichtet, eine Attributierung der originalen Bezugsquelle muss immer gegeben sein.

6.4 AUSWIRKUNGEN AUF DAS PROJEKT

Die Veröffentlichung des Projekts in Form einer Webseite bestimmt die Lizenzierung. Es handelt sich nicht um wiederverwendbare Software, das heißt das Projekt fällt nicht unter die *GNU Lesser General Public License* (LGPL). Mit der Verwendung von *OpenTripPlanner* wird ein Verweis auf die Lizenz dieser Software nötig.

Werden die Daten der *OpenStreetMap* verwendet um zusammen mit einer anderen Datenquelle eine neue Form von Daten zu erzeugen, handelt es sich nach der OpenStreetMap Foundation (2015) um eine „abgeleitete Datenbank“ oder ein „Datenbankderivat“. Diese muss erneut unter die *Open Data Commons Open Database Lizenz* gestellt werden. Wegen seiner relativ unveränderten



Abbildung 18 - Attributierung der Lizenz CC BY-NC-SA 3.0 DE mittels eines Logos und Hyperlink

Struktur in Form von Knoten und Kanten wird der durch den *Graph Builder* generierte *Graph* als ein solches Derivat betrachtet. Allerdings existiert darüber kein klarer Konsens unter den Entwicklern von OTP (SUNTER, BYRD, 2014). Es wird darauf verwiesen, sollte eine abgeleitete Version der Daten von OSM öffentlich zugänglich gemacht werden, genüge die Veröffentlichung der entsprechenden Konfigurationsdatei, die zur Erzeugung des Derivats genutzt wurde. Dabei handelt es sich um die Datei `graph-config.xml`, beschrieben in Abschnitt 5.4. Da der sonstige Quellcode öffentlich verfügbar ist, ist eine Reproduktion des *Graph*-Objekts möglich.

Somit ist die Vergabe einer Lizenz für den in Rahmen dieser Arbeit konzipierten und prototypisch implementierten Routenplaner frei gestellt. Denkbar ist z.B. eine Attributierung unter *Creative Commons Namensnennung – Nicht-kommerziell – Weitergabe unter gleichen Bedingungen 3.0* für Deutschland (CREATIVE COMMONS CORPORATION, 2014a). Damit ist sichergestellt, dass Dritte, die das Projekt weiterverwenden, dieses Lizenzmodell ebenfalls verwenden und den originalen Urheber nennen.

Die Kennzeichnung des Projekts kann z.B. über ein Pop-Up oder ein zusätzliches Fenster auf der Webseite erfolgen, welches Informationen zur Lizenzierung, bzw. einen Hyperlink zum vollständigen Lizenztext enthält. Creative Commons bietet außerdem kompakte Logos, von denen die Lizenz unmittelbar abgelesen werden kann. Die Kennzeichnung ist beispielhaft in Abbildung 18 zu sehen.

7 ERGEBNISSE

Dieser Abschnitt fasst alle im Laufe der Arbeit gewonnenen Erkenntnisse zusammen. Konnten die im Vorfeld gestellten Ziele erfüllt und die Fragen beantwortet werden? Es wird eine Bewertung abgegeben, um den Aufwand der Nutzung freier Daten und der Einrichtung von Open Source Software einzuschätzen. Außerdem werden mögliche Weiterentwicklungen präsentiert.

7.1 BEWERTUNG

Für die Umsetzung eines multimodalen Routenplaners auf der Basis von freien Daten eignet sich die Software OpenTripPlanner (OTP) sehr gut. Sie bietet den benötigten Funktionsumfang für die Bereitstellung eines webbasierten Dienstes zur Navigation und Routenplanung mit verschiedenen Transportmitteln, darunter dem öffentlichen Personennahverkehr (ÖPNV). Da es sich bei OTP um in Entwicklung befindlicher Open Source Software handelt, müssen eine Reihe von Einschränkungen in Kauf genommen werden, aus denen sich aber auch potentiell positive Nebenerscheinungen ziehen lassen.

Hohe Einstiegshürde

OTP ist eine komplexe Software mit gut 460.000 Zeilen Quellcode und mindestens 60 Abhängigkeiten von externen Modulen, die serverseitige Backend-Programmierung in Java mit Frontend-Gestaltung und Programmierung in HTML, CSS und JavaScript verbindet. Die grundlegende Nutzung mit den Standardeinstellungen vereinfacht den Einstieg zwar, bietet aber wenig Konfigurationspotential. Tiefergehende Anpassungen, z.B. beim

Routingalgorithmus oder der Visualisierung erfordern ein hohes Maß an Einarbeitungszeit. Die online verfügbare Dokumentation wird in unregelmäßigen Abständen aktualisiert und beschreibt teilweise veraltete Projektstrukturen. Direkte Hilfe lässt sich meist nur über die spezifischen Foren und Mailing-Listen erhalten

Auf der anderen Seite bietet die Nutzung von Open Source Software die Möglichkeit, an jeder beliebigen Stelle Modifikationen an der Funktionsweise des Programms vorzunehmen. Beispielsweise ließe sich die Laufzeit des Routingalgorithmus anhand neuer Forschungsergebnisse optimieren, oder ein neuer Transportmodus kann implementiert werden.

Verwaltung und Wartung

Die Nutzung freier Daten und Open Source Software bedeutet auf der einen Seite die Vermeidung von Kosten durch die Inanspruchnahme externer Dienstleister. Die Verwaltung und die Wartung der Software und der Daten obliegen allein dem Betreiber des Routenplaners. Das bedeutet, um möglichst aktuelle und zuverlässige Routingangaben machen zu können, muss der Betreiber selbst in regelmäßigen Abständen eine Aktualisierung des Datenbestandes oder der Softwareversion durchführen.

Zum anderen besteht trotzdem eine Abhängigkeit von den externen Datenanbietern. OpenStreetMap (OSM), General Transit Feed Specification (GTFS) und OTP sind zwar frei verfügbar, aber nicht im Unendlichen gültig. Die Datenqualität von OSM ist im starken Maße abhängig von der Aktivität der Mapper im Untersuchungsgebiet. Regionale Raumausschnitten außerhalb von urbanen Ballungsgebieten können über Monate oder Jahre hinweg wenig bis keine Aktualisierungen widerfahren. Unbemerkte Fehler in der Datenaufnahme können zu Fehlern bei der Routenplanung führen.

Der GTFS-Datensatz wird von einer öffentlichen Stelle in regelmäßigen Abständen herausgegeben, allerdings immer mit einem Verfallsdatum. Da die Fahrpläne des öffentlichen Nahverkehrs immer nur bis zu einem bestimmten Datum gültig sind, nach dem erneut Fahrplanänderungen vorgenommen

werden können, muss an dieser Stelle ebenfalls eine regelmäßige Aktualisierung vorgenommen werden. Es existieren zwar Tools zur Bearbeitung von GTFS-Daten, das Nachvollziehen aller Änderungen und manuelle Nachtragen ist aber aufwendig und zeitintensiv.

Die Entwicklung von OTP finanziert sich allein durch Spenden. Die meisten der Entwickler arbeiten nebenbei an dem Projekt. Im Falle der Veröffentlichung einer Version mit schwerwiegenden Fehlern, kann die weitere Bereitstellung eines Routenplaners unmöglich gemacht werden. Damit ist man gezwungen, entweder auf eine ältere Version mit geringerem Funktionsumfang auszuweichen, oder darauf zu warten, dass ein Update veröffentlicht wird, das den Fehler behebt.

Datenaufbereitung

Eine Datenaufbereitung für das Stadtgebiet Potsdam ist nicht zwingend notwendig, aber in gewissem Maße sinnvoll. Obwohl man bei einem urbanen Raum inzwischen davon ausgehen kann, dass die Objektdichte in OSM ausreichend hoch ist, lassen sich dennoch Lücken feststellen. Da für die Untersuchung besonders die Campusstandorte von Beachtung sind, fallen hier zum Teil fehlende Geometrien auf, z.B. Gehwege und Hauseingänge.

Auf attributiver Seite ist festzuhalten, dass im Sinne des Geocoding die Übertragung der Gebädefunktionen nach OSM möglich ist, aber nicht mit den Tagging-Konventionen einhergeht. Dies wäre zur Gewährleistung einer Namenssuche ausschließlich basierend auf Nominatim denkbar. Tatsächlich beherbergen aber einige Gebäude (z.B. Haus 9 am Neuen Palais) bis zu 11 universitäre Einrichtungen. All diese Informationen in der Tag-Struktur von OSM einzupflegen ist zwar möglich. Allerdings lassen sich bei der Nutzung von Nominatim Probleme feststellen, z.B. das `tag description=*` auszulesen. Die Auslagerung in einen externen Geocoder, wie in 5.3.2 beschrieben, ist daher eine geeignete Alternative.

Um die angestrebte Haus-zu-Haus-Navigation umzusetzen, ist eine weitere Erwähnung des Geocoders nötig. Zwar sind in OSM die meisten Adressbereiche

hausnummerngenau eingetragen, an den Universitätsstandorten ist dies aber zum Teil nicht möglich. Die Standorte Golm, Neues Palais und Griebnitzsee sind alle unter einer Adresse und Hausnummer erreichbar (z.B. Golm - Karl-Liebknecht-Straße 24/25). Die eigentliche Unterscheidung der einzelnen Fakultäten und Institute erfolgt über universitätsinterne Hausnummern. In OSM sind diese unvollständig eingetragen und die Hauseingänge fehlen. Daher eignet sich in diesem Fall die Auslagerung in einen externen Geocoder, wie in der Arbeit umgesetzt.

Nutzung des DGM

Die Implementierung des digitalen Geländemodells funktioniert problemlos. Die Datenmenge für den Graph wird nur unwesentlich vergrößert. Ohne DGM beträgt die Dateigröße knapp 98 Megabyte, mit DGM 108 MB. Routing mit digitalem Geländemodell wirkt sich auf den Transportmodus *Fahrrad* aus. Über das Konfigurationsdreieck ist der Nutzer in der Lage die Routenführung anhand der Optionen *schnell (QUICK)*, *flach (FLAT)* oder *radfreundlich/sicher (BIKE FRIENDLY)* zu optimieren (siehe Abbildung 19).

Im Profil *Schnell* wird zwar die kürzeste Fahrzeit erreicht, dafür aber ein kontinuierlicher Anstieg über 20m in Kauf genommen. Der maximale Höhenunterschied im Profil *Flach* beträgt dagegen 14m, bei einer Fahrzeit von 12 Minuten. Im Profil *Radfreundlich/Sicher* werden Radwege besonders bevorzugt. Der Algorithmus wählt in diesem Fall eine Strecke entlang ausgewiesener Fahrradwege, die Fahrzeit beträgt knapp 14 Minuten. Auf den ersten Blick erscheinen diese Unterschiede minimal. Das ist auf die Morphologie des Untersuchungsgebietes zurückzuführen. Bei einem Einsatz von OTP in einer Region mit größeren Höhenunterschieden im Relief ist diese Funktion jedoch sehr nützlich.

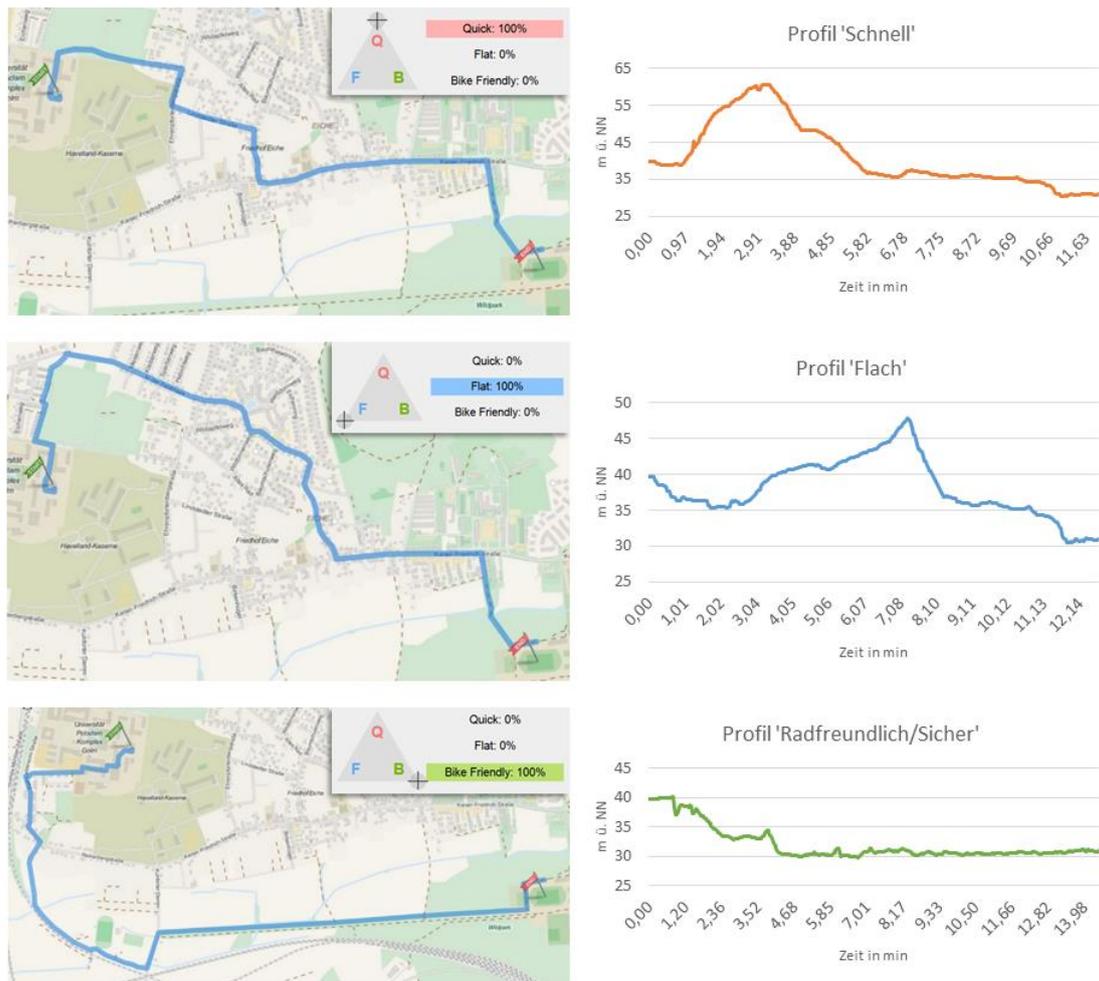


Abbildung 19 - Vergleich Streckenoptimierung für Fahrradfahrer

Veröffentlichung

Die Veröffentlichung eines Service für multimodales Routing an der Universität Potsdam unter einer freien Lizenz nach *Creative Commons* ist möglich. OTP wird unter einer Lizenz speziell für freie Software veröffentlicht und es genügt der Verweis auf den Ursprung des Projekts. Durch die Verknüpfung der Daten von OSM mit den Nahverkehrsdaten aus dem GTFS-Paket entsteht Datenbankderivat, welches im Sinne des Gemeinschaftsgedanken des OpenStreetMap-Projekts an die Community zurückgegeben werden sollte. Falls auf Anfrage Dritter die generierten Daten verlangt werden sollten, reicht es aus, die Konfigurationsdatei bereitzustellen, welche zur Erzeugung des Graphen dient.

7.2 AUSBLICK

Aktualisierung der Software

In dieser Arbeit wird nicht auf die aktuellste Version von OTP zurückgegriffen. Dies liegt zum einen daran, dass während der Bearbeitungszeit eine neuere Version veröffentlicht wurde, in der die komplette modulare Struktur der Software verworfen wurde und eine erneute Einarbeitungsphase verlangt hätte. Zum anderen handelt es sich um die einzige Iteration, welche erfolgreich in einer Testumgebung implementiert werden konnte. Eine Überführung der bisherigen Arbeitsschritte auf die aktuelle Version von OTP ist aber denkbar. Es werden Optimierungen an der Methode zur Zusammenführung des Graphen angeführt, die unter anderem eine Reduktion der Datenmenge zur Folge haben. Ein Nachteil der alten Version ist, dass die graphische Nutzeroberfläche lediglich in englischer Sprache verfügbar ist. Neuere Versionen bieten verschiedene Übersetzungen an, darunter Deutsch. Außerdem werden weitere Transportmodi in Form von Kombinationen von Verkehrsmitteln unterstützt. Dazu zählen:

- Parken und Reisen (*Park and Ride*), also die Kombination aus ÖPNV und PKW
- *Kiss and Ride*, Kurzzeithalteplätze oder Wendeschleifen in der Nähe größerer Haltestellen des ÖPNV (z.B. an Bahnhöfen)
- *Bike and Ride*, Fahrradstellplätze an Haltestellen des ÖPNV zur Vergrößerung zur erweiterten Erreichbarkeit von Haltestellen

Kiss and Ride wird z.B. häufig von Studenten genutzt, die sich ohne lange Wartezeit von Bus- oder Zugbahnhöfen abholen lassen wollen. Daher ist eine Ergänzung des Routenplaners um diese Funktionen sinnvoll.

Echtzeitdaten

Ein Nachteil der Nutzung von GTFS für die Verbindungsauskünfte über den Nahverkehr ist, dass es sich um einen statischen Datensatz mit den idealen Abfahrtszeiten handelt. Verspätungen oder Ausfälle werden nicht beachtet. Anhand von zwei Vorschlägen, ließe sich dieser Mangel beheben:

- a) Nutzung einer neuen Entwicklung von Google, *GTFS-Realtime* (GOOGLE DEVELOPERS, 2012b)
- b) Nutzung der VBB-API

Bei GTFS-Realtime (GTFS-RT) handelt es sich um eine Weiterentwicklung des Konzepts von GTFS. Es werden Verkehrsupdates, Serviceinformationen und Fahrzeuginformationen übermittelt. GTFS-RT wird in aktuelleren Versionen von OTP nativ unterstützt, findet im Moment aber fast ausschließlich in der Routingapplikation von Google Maps Verwendung. Der VBB sieht im Moment keinen Grund zusätzlich zum statischen GTFS-Datensatz Aktualisierungen über GTFS-RT zu veröffentlichen, zumal die eigene API diesen Zweck optimal erfüllt (PILZ, 2014).

Die andere Variante betrifft die Nutzung der VBB-API. In diesem Fall ist es denkbar, komplett auf die statischen GTFS-Daten zu verzichten. Routinganfragen für alle Transportmodi, den ÖPNV ausgenommen, würden weiterhin über OTP ablaufen. Anfragen für den Nahverkehr wären ausgelagert auf die API, sodass sofort die aktuellsten Verbindungen übertragen werden. Beide Methoden bedeuten jeweils Implementierungsaufwand auf der entsprechend anderen Seite. GTFS-RT wird zwar von OTP unterstützt, müsste aber vom VBB angeboten werden. Deren API liefert Nahverkehrsinformationen mit Echtzeitdaten, müsste aber in OTP implementiert werden.

Mobile Anwendung

In seiner jetzigen Form handelt es sich bei der Anwendung um einen webbasierten Routenplaner, d.h. es existiert keine native Applikation (*App*) für Smartphone und andere mobile Geräte. Dies ist aber nützlich, da Verbindungsauskünfte oft unterwegs erfolgen müssen und Smartphones unter Studenten weit verbreitet sind. Der Campus Routenplaner lässt sich zwar über den Browser eines Mobilgeräts ansteuern, ist aber nicht für dessen kleinen Bildschirm optimiert (siehe Abbildung 20). Allerdings existiert für OTP eine auf Android beruhende Applikation. Hier lassen sich benutzerdefinierte Serverinstanzen für multimodales Routing einbinden. In einem Test

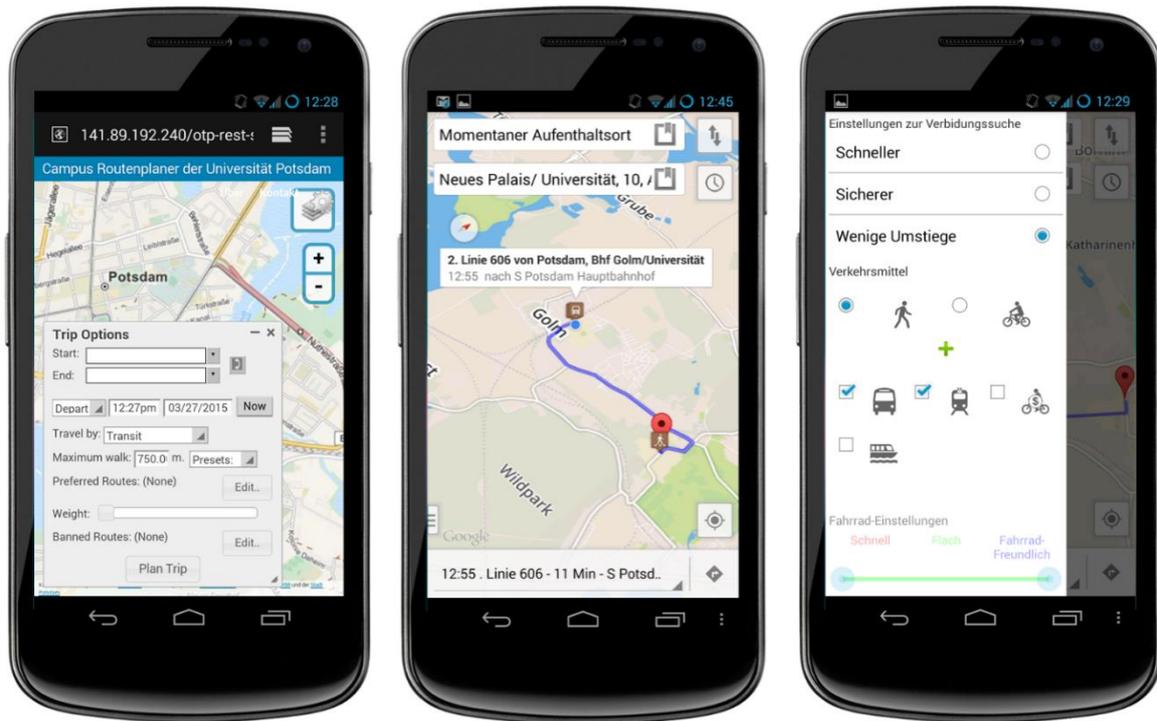


Abbildung 20 - Campus Routenplaner auf dem Smartphone. Links: OTP-Webserver im mobilen Browser. Mitte und rechts: Der eingerichtete Server eingebunden in die OTP-App unter Android.

funktioniert dies gut. Da der Quellcode für die App ebenfalls offen zur Verfügung steht, wäre die Umsetzung als eigenständige Anwendung möglich.

Indoor-Navigation

In seiner jetzigen Form bietet der konzipierte Campus Routenplaner eine lückenlose Haus-zu-Haus-Navigation. Durch eine Erweiterung der Datenstruktur des lokalen OSM-Datensatzes ist auch eine rudimentäre Indoor-Navigation denkbar. In der OpenStreetMap Foundation (2015b) existieren Vorschläge, welche *tags* für Flächen innerhalb von Gebäuden genutzt werden können:

- `level=n` – Markiert, auf welcher Etage sich Objekte befinden
- `room=yes` – Markiert eine Geometrie als Raum innerhalb eines Gebäudes
- `highway={elevator, steps}` – Markiert Fahrstühle und Treppen, wird bereits außerhalb von Gebäuden genutzt
- `highway=corridor` – Vorgeschlagenes *tag* für Innenflächen von Gebäuden

Denkbar wäre z.B. eine Variante einer Grundkarte, welche ab einer bestimmten Vergrößerungsstufe Raumpläne der Gebäude, statt der Grundrisse anzeigt. Im Rahmen des Projekts „Barrierefreiheit“ von Bartsch (2014) werden für die drei Universitätsstandorte Raumpläne digitalisiert und nach Zugänglichkeit kategorisiert. OTP akzeptiert die o.g. *tags* bereits und würde sich für Indoor-Navigation eignen (OPENSTREETMAP FOUNDATION, 2014b). Problematisch gestaltet sich dabei die Visualisierung verschiedener dreidimensionaler Ebenen eines Gebäudes auf einer zweidimensionalen Grundkarte. Auf diesem Gebiet besteht in besonderem Maße Forschungsbedarf.

Netzwerkanalyse

In Abschnitt 5.1.1 wird darauf verwiesen, dass die Software *OpenTripPlanner* über ein Modul für Netzwerkanalysen verfügt. Theoretisch lässt es sich über die graphische Nutzeroberfläche aufrufen, im Rahmen dieser Arbeit ist diese Funktion aber nicht verfügbar. Über den *Analyst-Client* lassen sich Erreichbarkeitsanalysen für die verschiedenen Transportmodi durchführen und visualisieren. Webbasierte Erreichbarkeitsanalysen werden bereits bei HOLLBURG U. A. (2012) anhand von Daten der OSM für Fußgänger vorgestellt. Mit Hilfe des OTP-Moduls lassen sich darüber hinaus verschiedene Transportmodi miteinander kombinieren, z.B. ÖPNV und Fahrrad. Eine beispielhafte Anwendung dafür wäre das Finden eines gemeinsamen Treffpunkts innerhalb gleicher Reisezeit. Abbildung 21 zeigt eine solche Anfrage. Es ist anzumerken, dass sich das Analysemodul ebenfalls in Entwicklung befindet und bei der Visualisierung Optimierungsbedarf besteht.

Person A startet beim grünen Marker und nutzt den ÖPNV, Person B startet beim roten Marker und nutzt das Fahrrad. Lila markierte Flächen zeigen Gebiete, die von beiden Startpunkten aus innerhalb von maximal 30 Minuten zu erreichen sind.

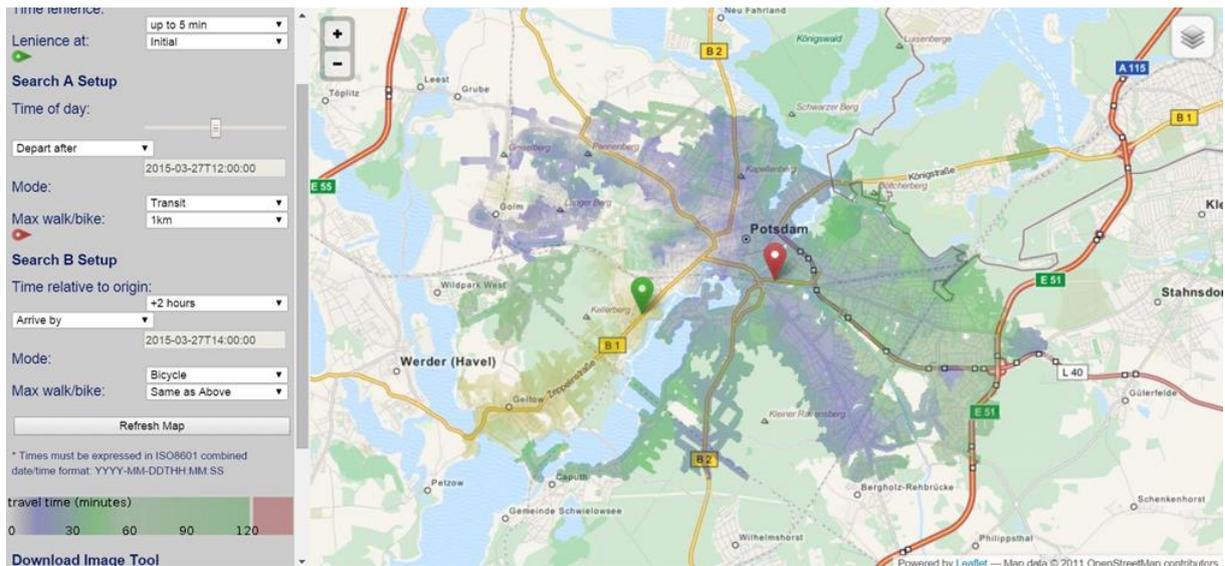


Abbildung 21 - Analysemodul. Erreichbarkeitsanalyse mit verschiedenen Transportmitteln.

Mobile.UP

Die Einbindung des Routenplaners im Rahmen der Applikation *Mobile.UP* für Smartphones ist denkbar. Die als mobiler Studienbegleiter konzipierte Anwendung ließe sich um die Routingfunktion mittels der von *OpenTripPlanner* bereitgestellten API erweitern.

7.3 ZUSAMMENFASSUNG

Die erfolgreiche Konzeption und prototypische Implementierung eines multimodalen Routenplaners am Beispiel der Universität Potsdam zeigt, dass mit Hilfe freier Daten und quelloffener Software die Bereitstellung eines professionellen Routingdienstes möglich ist (Abbildung 22). Damit ist die Grundlage für ein Modul zur Navigation mittels PKW, Fahrrad, ÖPNV und zu Fuß auf und zwischen den Campusstandorten Neues Palais, Golm und Griebnitzsee gelegt. Mit Hilfe der webbasierten Nutzeroberfläche werden Studienanfänger und Gäste der Universität Potsdam in die Lage versetzt, sich bei der erstmaligen Begehung des Geländes der Universität zu orientieren. Aber auch Studenten, Lehrpersonal oder Mitarbeitern, welche einen zuvor nicht besuchten Standort aussuchen müssen, kann der Campus Routenplaner als Hilfe dienen.

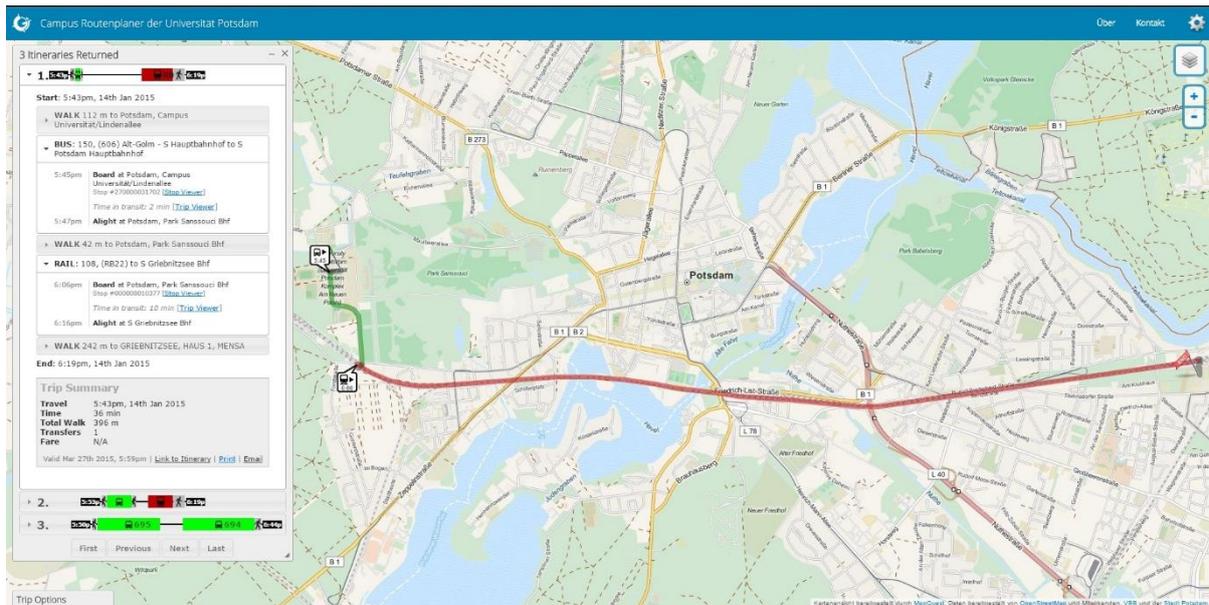


Abbildung 22 - Campus Routenplaner

Die quelloffene und freie Software *OpenTripPlanner* bietet durch die Nutzung freier Daten einen eindeutigen Kostenvorteil gegenüber Anbietern kommerzieller Daten. Allerdings ist der Einstieg in die Arbeit mit der Software anspruchsvoll, da die online verfügbare Dokumentation teilweise unvollständig, veraltet oder nicht aussagekräftig genug ist. In diesen Fällen ist das Ausweichen auf Entwicklerforen und Mailing Listen notwendig. Eine Umsetzung aller im Konzept vorgestellten Funktionen ist möglich, und die Erweiterung um neue Funktionen dank des offen liegenden Quelltexts möglich.

In kleinem Rahmen ist eine Aufbereitung der freien *OpenStreetMap*-Daten nötig. Es zeigen sich Lücken in der Attributierung universitärer Einrichtungen und bei der geometrischen Vollständigkeit. Diese Probleme können durch die Einbindung eines externen Geocoders und der Nachbearbeitung der Daten mittels eines freien Werkzeugs behoben werden.

Die Nutzung der freien Software und der freien Daten stellt aus lizenzrechtlicher Sicht kein Problem dar. Nach Betrachtung aller im Projekt verwendeten Daten und deren Lizenzen ist die Veröffentlichung des Campus Routenplaners im Rahmen einer freien Lizenz zur Namensnennung des Urhebers, der nichtkommerziellen Verwendung und Weitergabe unter gleichen Bedingungen möglich.

Abschließend bleibt festzuhalten, dass es sich bei Umsetzung in Form eines Campus Routenplaners lediglich um ein Beispiel handelt. Theoretisch ist der Einsatz von *OpenTripPlanner* in Kombination mit *OpenStreetMap*- und *General Transit Feed Specification*-Daten in jedem beliebigen Kontext möglich. Die Software unterstützt auch das Routing über mehrere Graphen hinweg. Würden mehr Anbieter von Nahverkehr ihre Daten veröffentlichen, wie der Verkehrsverbund Berlin-Brandenburg, könnte eine quelloffene Alternative zu Anbietern wie *Quixxit* (DB VERTRIEB GMBH, 2014) geschaffen werden, deren Ziel *seamless navigation* ist – also die möglichst unterbrechungsfreie Navigation von A nach B mit jedem beliebigen Transportmittel XY.

8 LITERATURVERZEICHNIS

Barbeau, Sean J. (2012): „barbeau/OpenTripPlanner - GitHub“. Abgerufen am 15.09.2014 von <https://github.com/barbeau/OpenTripPlanner>.

Bartelme, Norbert (2005): *Geoinformatik*. 4. Aufl. Berlin, Heidelberg: Springer.

Bartsch, Stefan (2014): „Projekt Barrierefreiheit: Universität Potsdam - Wirtschafts- und Sozialwissenschaftliche Fakultät / Professur für Technologie und berufliche Orientierung“. Abgerufen am 27.03.2015 von <http://www.uni-potsdam.de/projekt-barrierefreiheit/index.html>.

Bergson-Shilcock, Nicholas (2011): „Architecture OpenTripPlanner“. Abgerufen am 08.03.2015 von <https://github.com/opentripplanner/OpenTripPlanner/wiki/Architecture>.

BerlinOnline Stadtportal GmbH & Co. KG (2015): „FAQ | Offene Daten Berlin“. Abgerufen am 03.03.2015 von <http://daten.berlin.de/faq>.

Bertin, Jaques (1974): *Graphische Semiotik: Diagramme, Netze, Karten*. Berlin, New York: de Gruyter.

Bischoff, Andreas (2011): „Dienste für Smartphones an Universitäten - ein plattformunabhängiges Augmented Reality Campus-Informationssystem für iPhone und Android-Smartphones“. In: Werner, Martin; Roth, Jörg (Hrsg.) *8. GI/KuVS-Fachgespräch Ortsbezogene Anwendungen und Dienste*. Berlin, S. 127–135.

Bleeke, Melanie (2006): „Fußgängernavigation auf kombinierten Datenbeständen“. Universität Hannover.

Brandt, Roland (2013): „Lagepläne Universität Potsdam“. Abgerufen am 19.03.2015 von <http://www.uni-potsdam.de/db/zeik-portal/gm/lageplan-up.php?komplex=2>.

Braun, Manuel (2012): „Multi-Model Route Planning with Transfer Patterns“. Albert-Ludwigs-Universität Freiburg.

Literaturverzeichnis

- Byrd, Andrew (2011): „Elevation - OpenTripPlanner Wiki“. Abgerufen am 24.03.2015 von <https://github.com/opentripplanner/OpenTripPlanner/wiki/Elevation>.
- Byrd, Andrew (2013): „Frequently Asked Questions - OpenTripPlanner Wiki“. Abgerufen am von <https://github.com/opentripplanner/OpenTripPlanner/wiki/FrequentlyAskedQuestions>.
- Caprioli, Mauro; Tarantino, Eufemia (2003): „Standards and Quality in GIS Contexts“. In: *Approaches and new Concepts in SIM.*, S. 1–13.
- Creative Commons Corporation (2014a): „Creative Commons — Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 3.0 Deutschland — CC BY-NC-SA 3.0 DE“. Abgerufen am 07.03.2015 von <http://creativecommons.org/licenses/by-nc-sa/3.0/de/>.
- Creative Commons Corporation (2014b): „Creative Commons – Namensnennung 3.0 Deutschland – CC BY 3.0 DE“. Abgerufen am von <http://creativecommons.org/licenses/by/3.0/de/>.
- Czebotar, Jehiah (2014): „Transit Agencies Providing GTFS Data“. Abgerufen am 17.12.2014 von <http://www.gtfs-data-exchange.com/agencies>.
- DB Vertrieb GmbH (2014): „Quixxit - Kontakt“. Abgerufen am 27.03.2015 von <http://www.qixxit.de/kontakt/>.
- Delling, Daniel; Dibbelt, Julian; Pajor, Thomas; u. a. (2012): „Computing and Evaluating Multimodal Journeys“. In: KIT, Fakultät für Informatik.
- Dibbelt, Julian; Pajor, Thomas; Wagner, Dorothea (2012): „User-Constrained Multi-Modal Route Planning“. In: *networks*. 6 , S. 10.
- DiBiase, David; MacEachren, Alan M.; Krygier, John B; u. a. (1992): „Animation and the Role of Map Design in Scientific Visualization“. In: *Cartography and Geographic Information Systems*. Taylor & Francis 19 (4), S. 201–214, DOI: 10.1559/152304092783721295.
- Diestel, Reinhard (2000): *Graphentheorie. Elektronische Ausgabe 2000*. 2. Aufl. Heidelberg: Springer.
- Dijkstra, Edsger Wybe (1959): „A note on two problems in connexion with graphs“. In: *Numerische Mathematik*. Springer-Verlag 1 (1), S. 269–271, DOI: 10.1007/BF01386390.
- Foursquare (2014): „OpenStreetMap: Eine kostenlose Weltkarte, die sich bearbeiten lässt“. Abgerufen am 16.12.2014 von <https://de.foursquare.com/about/osm>.
- Free Software Foundation Inc. (2007): „GNU Lesser General Public License, Version 3.0 - GNU-Projekt - Free Software Foundation“. Abgerufen am von <http://www.gnu.org/licenses/lgpl-3.0>.

- Geisberger, Robert (2008): „Contraction Hierarchies : Faster and Simpler Hierarchical Routing in Road Networks of“. Universität Karlsruhe.
- Gendre, Patrick (2014): „OpenTripPlanner API“. Abgerufen am 23.03.2015 von <https://github.com/opentripplanner/OpenTripPlanner/wiki/API>.
- Geofabrik GmbH (2013): „OpenStreetMap Data Extracts“. Abgerufen am 17.12.2014 von <http://download.geofabrik.de/>.
- Gerwinski, Peter (2007): „LGPL v3“. Abgerufen am 03.03.2015 von <http://www.gnu.de/documents/lgpl-3.0.de.html>.
- Google Code (2014): „OtherGTFSTools“. Abgerufen am 17.12.2014 von <https://code.google.com/p/googletransitdatafeed/wiki/OtherGTFSTools>.
- Google Developers (2013): „General Transit Feed Specification Reference - Transit — Google Developers“. Abgerufen am 30.01.2014 von <https://developers.google.com/transit/gtfs/reference>.
- Google Developers (2012a): „What is GTFS? - Transit — Google Developers“. Abgerufen am 30.01.2014 von <https://developers.google.com/transit/gtfs/>.
- Google Developers (2012b): „What is GTFS-realtime? - Transit — Google Developers“. Abgerufen am 27.03.2015 von <https://developers.google.com/transit/gtfs-realtime/>.
- Google Inc. (2014a): „Google Maps“. Abgerufen am 31.07.2014 von <https://www.google.de/maps/dir/52.474309,12.9519167/52.4033156,12.9810057/@52.4389588,12.9039632,12z/data=!4m4!4m3!1m0!1m0!3e1>.
- Google Inc. (2014b): „Google Maps“. Abgerufen am 31.07.2014 von <https://www.google.de/maps/dir/52.474309,12.9519167/52.4033156,12.9810057/@52.4554993,12.9725799,11z/am=t/data=!3m1!4b1!4m6!4m5!1m0!1m0!2m1!6e4!3e2>.
- Haklay, Mordechai (2010): „How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets“. In: *Environment and Planning B: Planning and Design*. 37 (4), S. 682–703, DOI: 10.1068/b35097.
- Hart, Peter E.; Nilsson, Nils J.; Raphael, Bertram (1968): „A Formal Basis for the Heuristic Determination of Minimum Cost Paths“. *Systems Science and Cybernetics, IEEE Transactions on.*, DOI: 10.1109/TSSC.1968.300136. — ISBN: 0536-1567 VO - 4
- Hibbe, Marcel; Robeck, Mathias (2013): „Konzeption und Implementierung eines multimodalen und multitemporalen Campus-Routenplaners am Beispiel der Universität Potsdam“. Potsdam: Universität Potsdam.
- Hillsman, Edward L.; Barbeau, Sean J. (2011): *Enabling Cost-Effective Multimodal Trip Planners through Open Transit Data*. Tampa, FL.

Literaturverzeichnis

- Hochschule Bochum (2014): „CampusInformationssystem - Hochschule Bochum“. Abgerufen am 09.03.2015 von <https://meinebo.hochschule-bochum.de/login/campusinformationssystem.html>.
- Hollburg, Henning; Sinn, Christoph; Volland, Patrick (2012): „Hier bin ich–was kann ich erreichen? Webbasierte, interaktive Erreichbarkeitsanalyse touristischer Ziele der Stadt Potsdam“. In: *Angewandte Geoinformatik 2012*. Salzburg, S. 317–322.
- Howe, Jeff (2006): „The Rise of Crowdsourcing“. *Wired*.
- Jiang, Bin (1996): „Cartographic visualization: analytical and communication tools“. In: *Cartography*. Taylor & Francis 25 (2), S. 1–11.
- Jordán Aldasorro, Juan G.; Vidal Peña, José; Segura García, Jaume (2009): „Sistema multimodal para la planificación de viajes en transportes públicos de código abierto y basado en estándares „de facto“ = A multimodal journey planner based on open source software and „de facto“ standards“. Universitat de Girona. Servei de Sistemes d'Informació Geogràfica i Teledetecció. — ISBN: 978-48-691-9409-6
- komoot GmbH (2014): „komoot - imprint“. Abgerufen am 27.03.2015 von <https://www.komoot.de/imprint>.
- Kreil, Michael (2012): „Die ersten Fahrplandaten“. Abgerufen am 24.03.2015 von <http://openplanb.tumblr.com/post/30878899005/die-ersten-fahrplandaten>.
- Li, Zhilin; Pun-cheng, Lilian; Shea, Geoffrey (2004): „Design of web maps for navigation purpose“. In: Altan, Orhan (Hrsg.) *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*. Istanbul, S. 353–358.
- MAIRDUMONT GmbH & Co. KG (2014): „Fußgänger-Routenplaner und Wanderplaner“. Abgerufen am 31.07.2014 von <http://www.falk.de/routenplaner-fussgaenger>.
- Martin-Anderson, Brandon (2004): „Graphserver - Appropedia: The sustainability wiki“. Abgerufen am 12.03.2015 von <http://www.appropedia.org/Graphserver>.
- Martin-Luther-Universität Halle-Wittenberg (2010): „MLU Campus Maps“. Abgerufen am von <http://www.maps.uni-halle.de/>.
- Matzat, Lorenz (2012): „Verpasste Open Data-Chance: Deutsche Bahn schenkt einzig Google seine Fahrplandaten | netzpolitik.org“. *Netzpolitik.org*. Abgerufen am 30.06.2014 von <https://netzpolitik.org/2012/verpasste-open-data-chance-deutsche-bahn-schenkt-einzig-google-seine-fahrplandaten/>.
- Merrifield, Timothy (2010): „Heuristic Route Search in Public Transportation Networks“. University of Illinois at Chicago.
- Microsoft Inc. (2014): „Bing Karten – Anfahrtsbeschreibungen, Verkehrsinfos und Straßenbedingungen“. Abgerufen am 31.07.2014 von

<http://www.bing.com/maps/?FORM=Z9LH3#Y3A9NTluNDE2MjMxfjEzLjA0MTE2OCZsdmw9MTlmc3R5PXlmcnRwPXBvcy41Mi40NDI4NTBfMTMuMDQ1MTU0X2luJTlwZGVyJTlwTiVDMyVBNGhlJTlwdm9uJTlwQW0lMjBSZWWh3ZWclMjAyMiUyQyUyMDE0NDc2JTlwTmV1JTlwRmFocmxhbmRfX19hX35wb3MuNTluMzg5MjQzXzEzLjAyNjk1OF9pbiUyMGRlciUyME4lQzMIQTRoZSUyMHZvbiUyMFJvc2VnZ2Vyc3RyYSVDMYU5RmUIMkMIMjAxNDQ3MSUyMFBvdHNkYW1fX19hXyZtb2RlPvc=>.

Open Knowledge Foundation (2015): „ODC Open Database License (ODbL) Summary | Open Data Commons“. Abgerufen am 02.03.2015 von <http://opendatacommons.org/licenses/odbl/summary/>.

OpenStreetMap Foundation (2015a): „DE:Legal FAQ – OpenStreetMap Wiki“. Abgerufen am 02.03.2015 von http://wiki.openstreetmap.org/wiki/DE:Legal_FAQ.

OpenStreetMap Foundation (2014a): „DE:OpenStreetMap benutzen – OpenStreetMap Wiki“. Abgerufen am 04.11.2014 von <http://wiki.openstreetmap.org/wiki/DE:Anwender>.

OpenStreetMap Foundation (2015b): „Indoor Mapping – OpenStreetMap Wiki“. Abgerufen am 27.03.2015 von http://wiki.openstreetmap.org/wiki/Indoor_Mapping.

OpenStreetMap Foundation (2014b): „OpenTripPlanner – OpenStreetMap Wiki“. Abgerufen am 06.01.2015 von <http://wiki.openstreetmap.org/wiki/OpenTripPlanner>.

OpenStreetMap Foundation (2014c): „Stats – OpenStreetMap Wiki“. Abgerufen am 04.11.2014 von <http://wiki.openstreetmap.org/wiki/Statistics>.

OpenStreetMap Foundation (2014d): „Tags | OpenStreetMap Taginfo“.

OpenStreetMap Foundation (2008): „The licence: where we are, where we’re going | OpenStreetMap Blog“. Abgerufen am 02.03.2015 von <https://blog.openstreetmap.org/2008/01/07/the-licence-where-we-are-where-were-going/>.

OpenStreetMap Foundation (2012): „Your First ODbL Planet | OpenStreetMap Blog“. Abgerufen am 04.11.2014 von <https://blog.openstreetmap.org/2012/09/06/your-first-odbl-planet/>.

Pilz, Alexander (2014): „Gesprächsnotizen, geführt vom Verfasser. 26.04.2014“. Berlin.

Ramm, Frederik; Topf, Jochen (2010): *OpenStreetMap - Die freie Welt nutzen und mitgestalten*. 3. überarb. Berlin: Lehmanns Media.

Rezić, Slaven (2014): „BBBike - Routenplaner für Radfahrer in Berlin und Brandenburg“. Abgerufen am 31.07.2014 von <http://www.bbbike.de/cgi-bin/bbbike.cgi?begin=1>.

Rezić, Slaven (1999): „Entwicklung eines Informationssystems für die Fahrradstreckenplanung“. Technische Universität Berlin.

Literaturverzeichnis

- Roick, Oliver; Neis, Pascal; Zipf, Alexander (2011): „Volunteered Geographic Information– Datenqualität und Nutzungspotentiale am Beispiel von OpenStreetMap“. In: *Kommission „Angewandte Kartographie– Geovisualisierung“ der Deutschen Gesellschaft für Kartographie (DGfK)– Symposium Königslutter*.
- RWTH Aachen (2014): „CAMPUS-Informationssystem - RWTH AACHEN UNIVERSITY IT Center - Deutsch“. Abgerufen am 09.03.2015 von <http://www.itc.rwth-aachen.de/cms/IT-Center/Dienste/kompletter-Servicekatalog/~esqv/Campus-Plattform/>.
- Sunter, Patrick; Byrd, Andrew (2014): „Guidance on complying with OpenStreetMap license when using OSM to build OpenTripPlanner graphs?–Google Groups“. Abgerufen am 07.03.2015 von <https://groups.google.com/forum/#!topic/opentripplanner-users/Az0X-yfZmKI>.
- Technische Universität Dresden (2012): „Über den Campusnavigator TU Dresden“. Abgerufen am 04.11.2014 von <https://navigator.tu-dresden.de/ueber>.
- Thoma, Jörg (2013): „Open Data: Verkehrsverbund Berlin-Brandenburg gibt Fahrplan frei - Golem.de“. *Golem.de*. Abgerufen am 30.06.2014 von <http://www.golem.de/news/open-data-verkehrsverbund-berlin-brandenburg-gibt-fahrplan-frei-1304-99026.html>.
- Thuerkow, Detlef (2010): „Campus-Informationssystem“. Abgerufen am 09.03.2015 von <http://www.campus3d.uni-halle.de/>.
- Topf, Christian; Topf, Jochen (2014): „Über Taginfo | OpenStreetMap Taginfo“. Abgerufen am 15.02.2015 von <http://taginfo.openstreetmap.org/about>.
- Universität Hamburg (2013): „Campus Navigator der Universität Hamburg“. Abgerufen am 09.03.2015 von <https://www.uni-hamburg.de/onTEAM/campus/>.
- Universität Heidelberg (2014): „Campus-Informationssysteme - Rechenzentrum der Universitaet Heidelberg“. Abgerufen am 09.03.2015 von <http://www.urz.uni-heidelberg.de/service-katalog/campusinfo/>.
- Universität Potsdam - Dezernat für Planung Statistik Forschungsangelegenheiten und Organisation (2014): „Universität Potsdam - Statistiken/Zahlen über die Universität“. Abgerufen am 30.06.2014 von <http://www.uni-potsdam.de/u/verwaltung/dezernat1/statistik/>.
- VBB Verkehrsverbund Berlin-Brandenburg GmbH (2015): „Alles ist erreichbar! Aufgaben des VBB. / VBB GmbH: VBB - Alles ist erreichbar“. Abgerufen am 06.01.2015 von <http://www.vbb.de/de/article/vbb-gmbh/geschaeftsfelder-aufgaben-bereiche/18734.html>.
- VBB Verkehrsverbund Berlin-Brandenburg GmbH (2014): „Datenschema des VBB“. Berlin.

Literaturverzeichnis

- VBB Verkehrsverbund Berlin-Brandenburg GmbH (2012): „Einladung zum Open Data Entwicklertag am 29. November 2012 / Presse / Pressearchiv / 2012 - 11: VBB - Alles ist erreichbar“. Abgerufen am 06.01.2015 von <http://www.vbb.de/de/article/presse/open-data-einladung-zum-entwicklertag-am-29-november-2012/12223.html>.
- Wikimedia Labs (2014): „Wikipedia on OpenStreetMap“. Abgerufen am 16.12.2014 von <http://tools.wmflabs.org/wiwosm/osm-on-ol/kml-on-ol.php?lang=de&uselang=de&zoom=5&lat=51.70274&lon=10.1479>.
- Van den Worm, Jeroen (2001): „Web map design in practice“. In: Kraak, Menno-Jan; Brown, Allan (Hrsg.) *Web Cartography - Developments and Prospects*. London, New York: Taylor & Francis, S. 213.
- Yahoo Inc. (2008): „Around the world and back again | Flickr Blog“. Abgerufen am 16.12.2014 von <http://blog.flickr.net/en/2008/08/12/around-the-world-and-back-again/>.
- Zielstra, Dennis; Zipf, Alexander (2010): „A comparative study of proprietary geodata and volunteered geographic information for Germany“. In: *13th AGILE International Conference on Geographic Information Science 2010 Guimarães, Portugal*. 1 , S. 1–15.

9 ANHANG

9.1 ANHANG A – APPLICATION-CONTEXT.XML

```
1 <!-- Universitäre Einrichtungen -->
2 <bean id="geocoderManual"
3     class="org.opentripplanner.geocoder.manual.ManualGeocoder">
4     <property name="pois">
5         <map>
6             <entry key="GOLM, HAUS 25, INSTITUT FÜR BIOCHEMIE UND CHEMIE"
7                 value="52.410289 12.974330" />
8             <entry key="GOLM, HAUS 25, INSTITUT FÜR BIOLOGIE"
9                 value="52.410289 12.974330" />
10            <entry key="GOLM, HAUS 26, INSTITUT FÜR BIOCHEMIE UND CHEMIE"
11                value="52.410207 12.975240" />
12            <entry key="GOLM, HAUS 26, INSTITUT FÜR BIOLOGIE"
13                value="52.410207 12.975240" />
14            <entry key="GOLM, HAUS 28, DEKANAT DER MATHEMATISCH-
15                NATURWISSENSCHAFTLICHEN FAKULTÄT" value="52.409660
16                12.974012" />
17            <entry key="GOLM, HAUS 28, INSTITUT FÜR PHYSIK UND
18                ASTRONOMIE" value="52.409660 12.974012" />
19            <entry key="GOLM, HAUS 27, INSTITUT FÜR ERD- UND
20                UMWELTWISSENSCHAFTEN, GEOWISSENSCHAFTEN" value="52.409590
21                12.974792" />
22            <entry key="GOLM, HAUS 29, EXZELLENZBEREICH
23                KOGNITIONSWISSENSCHAFTEN" value="52.409006 12.974677" />
24            <entry key="GOLM, HAUS 29, INSTITUT FÜR BIOCHEMIE UND
                BIOLOGIE" value="52.409006 12.974677" />
                <entry key="GOLM, HAUS 29, INSTITUT FÜR CHEMIE"
                value="52.409006 12.974677" />
                <entry key="GOLM, HAUS 29, INSTITUT FÜR ERD- UND
                UMWELTWISSENSCHAFTEN" value="52.409006 12.974677" />
                <entry key="GOLM, HAUS 29, POTSDAM TRANSFER" value="52.409006
                12.974677" />
                <entry key="GOLM, HAUS 29, UP TRANSFER GMBH" value="52.409006
                12.974677" />
                <entry key="GOLM, HAUS 8, HOCHSCHULGEBÄUDEMANAGEMENT (HGP)
                BEREICH UNIVERSITÄT POTSDAM" value="52.408985 12.976443" />
                <entry key="GOLM, HAUS 8, ZENTRALE EINRICHTUNG FÜR
                INFORMATIONSPERARBEITUNG UND KOMMUNIKATION (ZEIK)"
                value="52.408985 12.976443" />
                <entry key="GOLM, HAUS 6, DEPARTMENT FÜR LEHRERBILDUNG /
                MUSIK" value="52.409084 12.978140" />
                <entry key="GOLM, HAUS 6, KUNST" value="52.409084 12.978140"
                />
                <entry key="GOLM, HAUS 7, STUDENTENWOHNHEIM" value="52.408614
                12.978016" />
                <entry key="GOLM, HAUS 5, AUDIOVISUELLES ZENTRUM (AVZ)"
                value="52.408555 12.978760" />
                <entry key="GOLM, HAUS 5, HÖRSAAL, SEMINARRÄUME"
```

```
25         value="52.408555 12.978760" />
26 <entry key="GOLM, HAUS 12, HÖRSAAL" value="52.408928
27         12.979640" />
28 <entry key="GOLM, HAUS 12, INSTITUT FÜR ERD- UND
29         UMWELTWISSENSCHAFTEN" value="52.408928 12.979640" />
30 <entry key="GOLM, HAUS 12, GEOÖKOLOGIE" value="52.408928
31         12.979640" />
32 <entry key="GOLM, HAUS 18, UNIVERSITÄTSBIBLIOTHEK (IKMZ)"
33         value="52.408712 12.972918" />
34 <entry key="GOLM, HAUS 9" value="52.408492 12.977268" />
35 <entry key="GOLM, HAUS 10, FUHRPARK" value="52.407853
36         12.976462" />
37 <entry key="GOLM, HAUS 10A, PÄDAGOGISCHE WERKSTATT"
38         value="52.407853 12.976462" />
39 <entry key="GOLM, HAUS 10A, ZENTRUM FÜR LEHRERBILDUNG"
40         value="52.407853 12.976462" />
41 <entry key="GOLM, HAUS 2" value="52.407976 12.977645" />
42 <entry key="GOLM, HAUS 4, MENSA" value="52.407985 12.978550"
43         />
44 <entry key="GOLM, HAUS 17, ELTERN-KIND-RAUM 0.07"
45         value="52.407861 12.973453" />
46 <entry key="GOLM, HAUS 17, STUDENTENWOHNHEIM"
47         value="52.407861 12.973453" />
48 <entry key="GOLM, HAUS 16, STUDENTENWOHNHEIM"
49         value="52.407567 12.973370" />
50 <entry key="GOLM, HAUS 15, STUDENTENWOHNHEIM"
51         value="52.407390 12.973337" />
52 <entry key="GOLM, HAUS 14A, LESECAFE" value="52.407818
53         12.974299" />
54 <entry key="GOLM, HAUS 14, DEKANAT DER
55         HUMANWISSENSCHAFTLICHEN FAKULTÄT" value="52.407839
56         12.973964" />
57 <entry key="GOLM, HAUS 14, DEPARTMENT FÜR
58         ERZIEHUNGSWISSENSCHAFTEN" value="52.407839 12.973964" />
59 <entry key="GOLM, HAUS 14, DEPARTMENT FÜR LINGUISTIK"
60         value="52.407839 12.973964" />
61 <entry key="GOLM, HAUS 14, DEPARTMENT FÜR PSYCHOLOGIE"
62         value="52.407839 12.973964" />
63 <entry key="GOLM, HAUS 14, SPRACHZENTRUM" value="52.407839
64         12.973964" />
65 <entry key="GOLM, GEWÄCHSHAUS" value="52.407701 12.974929" />
66 <entry key="GOLM, HAUS 11, ALTE SPORTHALLE" value="52.407165
67         12.976974" />
68 <entry key="GOLM, HAUS 11, ARBEITSLEHRE / TECHNIK"
69         value="52.407165 12.976974" />
70 <entry key="GOLM, HAUS 1, INSTITUT FÜR ERD- UND
71         UMWELTWISSENSCHAFTEN, GEOÖKOLOGIE" value="52.407503
72         12.977729" />
73 <entry key="GOLM, HAUS 1, KLIMATOLOGIE, HYDROLOGIE,
74         LANDSCHAFTSÖKOLOGIE, BODENKUNDE" value="52.407503 12.977729"
75         />
76 <entry key="GOLM, HAUS 3, SONDERPÄDAGOGIK" value="52.407424
77         12.978478" />
78 <entry key="GOLM, HAUS 3, STUDENTENWOHNHEIM" value="52.407424
79         12.978478" />
80 <entry key="GOLM, HAUS 13, INSTITUT FÜR ERD- UND
81         UMWELTWISSENSCHAFTEN" value="52.407419 12.978987" />
82 <entry key="GOLM, BUD" value="52.406779 12.977435" />
83 <entry key="GOLM, HAUS 24, INSTITUT FÜR ERD- UND
84         UMWELTWISSENSCHAFTEN, GEOGRAPHIE UND NATURRISIKENFORSCHUNG"
85         value="52.406828 12.976373" />
86 <entry key="GOLM, HAUS 24, DEPARTMENT LINGUISTIK / ALLGEMEINE
87         SPRACHWISSENSCHAFT" value="52.406828 12.976373" />
88 <entry key="GOLM, HAUS 24, DEPARTMENT PSYCHOLOGIE"
89         value="52.406828 12.976373" />
90 <entry key="GOLM, HAUS 24, INSTITUT FÜR GEOGRAPHIE"
91         value="52.406828 12.976373" />
92 <entry key="GOLM, HAUS 24, INSTITUT FÜR GRUNDSCHULPÄDAGOGIK"
```

59 value="52.406828 12.976373" />
<entry key="GOLM, HAUS 24, ABTEILUNG GEOINFORMATION UND
VISUALISIERUNG" value="52.406828 12.976373" />
60 <entry key="GOLM, HAUS 19, SPORTHALLE" value="52.406212
12.975613" />
61 <entry key="GOLM, HAUS 35, DEPARTMENT LINGUISTIK / ALLGEMEINE
SPRACHWISSENSCHAFT" value="52.406357 12.976260" />
62 <entry key="GOLM, HAUS 20, INSTITUT FÜR BIOCHEMIE / BIOLOGIE"
value="52.406412 12.976388" />
63 <entry key="NEUES PALAIS, HAUS T3, STUDENTENWOHNHEIM"
value="52.404756 13.009245" />
64 <entry key="NEUES PALAIS, HAUS T2, STUDENTENWOHNHEIM"
value="52.404776 13.009976" />
65 <entry key="NEUES PALAIS, HAUS T1, STUDENTENWOHNHEIM"
value="52.404641 13.010756" />
66 <entry key="NEUES PALAIS, HAUS 15" value="52.402611
13.009665" />
67 <entry key="NEUES PALAIS, HAUS 13" value="52.402418
13.008816" />
68 <entry key="NEUES PALAIS, HAUS 14" value="52.402183
13.009682" />
69 <entry key="NEUES PALAIS, HAUS 5, INSTITUT FÜR GERMANISTIK"
value="52.402181 13.010488" />
70 <entry key="NEUES PALAIS, HAUS 12, DEPARTMENT FÜR SPORT- UND
GESUNDHEITSWISSENSCHAFTEN" value="52.402181 13.010488" />
71 <entry key="NEUES PALAIS, HAUS 12, HOCHSCHULAMBULANZ"
value="52.402181 13.010488" />
72 <entry key="NEUES PALAIS, HAUS 12, MENSA" value="52.402181
13.010488" />
73 <entry key="NEUES PALAIS, HAUS 11, DEKANAT DER
PHILOSOPHISCHEN FAKULTÄT" value="52.401877 13.012558" />
74 <entry key="NEUES PALAIS, HAUS 11, HISTORISCHES INSTITUT"
value="52.401877 13.012558" />
75 <entry key="NEUES PALAIS, HAUS 11, INSTITUT FÜR JÜDISCHE
STUDIEN / JEWISH STUDIES" value="52.401877 13.012558" />
76 <entry key="NEUES PALAIS, HAUS 11, INSTITUT FÜR
LEBENSGESTALTUNG-ETHIK-RELIGIONSKUNDE" value="52.401877
13.012558" />
77 <entry key="NEUES PALAIS, HAUS 11, INSTITUT FÜR PHILOSOPHIE"
value="52.401877 13.012558" />
78 <entry key="NEUES PALAIS, HAUS 11, INSTITUT FÜR
RELIGIONSWISSENSCHAFTEN" value="52.401877 13.012558" />
79 <entry key="NEUES PALAIS, HAUS 11, KLASSISCHE PHILOLOGIE"
value="52.401877 13.012558" />
80 <entry key="NEUES PALAIS, HAUS 11, STUDENTENKELLER NIL"
value="52.401877 13.012558" />
81 <entry key="NEUES PALAIS, HAUS 11, UNIVERSITÄTSBIBLIOTHEK"
value="52.401877 13.012558" />
82 <entry key="NEUES PALAIS, HAUS 16" value="52.401310
13.009668" />
83 <entry key="NEUES PALAIS, HAUS 6, AG LEARNING"
value="52.400764 13.010016" />
84 <entry key="NEUES PALAIS, HAUS 6, KOORDINATIONSBURO FÜR
CHANCENGLEICHHEIT" value="52.400764 13.010016" />
85 <entry key="NEUES PALAIS, HAUS 6, NETZWERK STUDIENQUALITÄT
BRANDENBURG SQB" value="52.400764 13.010016" />
86 <entry key="NEUES PALAIS, HAUS 6, PERSONALRÄTE"
value="52.400764 13.010016" />
87 <entry key="NEUES PALAIS, HAUS 6, STUDIERENDENVERTRETUNG -
ASTA" value="52.400764 13.010016" />
88 <entry key="NEUES PALAIS, HAUS 6, UP TRANSFER - ZENTRALER
UNIVERSITÄRER PARTNER FÜR WISSENS- UND TECHNOLOGIETRANSFER"
value="52.400764 13.010016" />
89 <entry key="NEUES PALAIS, HAUS 6, ZENTRALE ABTEILUNG /
AUDIOVISUELLES ZENTRUM (Z / AVZ)" value="52.400764
13.010016" />
90 <entry key="NEUES PALAIS, HAUS 6, ZENTRUM FÜR
QUALITÄTSENTWICKLUNG IN LEHRE UND STUDIUM ZFQ"

91 value="52.400764 13.010016" />
<entry key="NEUES PALAIS, HAUS 3, DEZERNAT 3 PERSONAL- UND RECHTSANGELEGENHEITEN" value="52.400502 13.010613" />
92 <entry key="NEUES PALAIS, HAUS 3, DEZERNAT 4 HAUSHALT UND BESCHAFFUNG" value="52.400502 13.010613" />
93 <entry key="NEUES PALAIS, HAUS 9, CAREER SERVICE" value="52.400597 13.012673" />
94 <entry key="NEUES PALAIS, HAUS 9, ELTERN-KIND-RAUM 1.01" value="52.400597 13.012673" />
95 <entry key="NEUES PALAIS, HAUS 9, GESCHÄFTSSTELLE DES SENATS" value="52.400597 13.012673" />
96 <entry key="NEUES PALAIS, HAUS 9, HÖRSÄLE SEMINARRÄUME" value="52.400597 13.012673" />
97 <entry key="NEUES PALAIS, HAUS 9, KANZLERIN" value="52.400597 13.012673" />
98 <entry key="NEUES PALAIS, HAUS 9, PRÄSIDENT" value="52.400597 13.012673" />
99 <entry key="NEUES PALAIS, HAUS 9, PRÄSIDIALAMT" value="52.400597 13.012673" />
100 <entry key="NEUES PALAIS, HAUS 9, PRÄSIDIUM" value="52.400597 13.012673" />
101 <entry key="NEUES PALAIS, HAUS 9, REFERAT FÜR PRESSE- UND ÖFFENTLICHKEITSARBEIT" value="52.400597 13.012673" />
102 <entry key="NEUES PALAIS, HAUS 9, UNISHOP" value="52.400597 13.012673" />
103 <entry key="NEUES PALAIS, HAUS 9, ZENTRALE EINRICHTUNG FÜR INFORMATIONSVERRARBEITUNG UND KOMMUNIKATION ZEIK" value="52.400597 13.012673" />
104 <entry key="NEUES PALAIS, HAUS 10, UNIVERSITÄTSBIBLIOTHEK" value="52.400260 13.012709" />
105 <entry key="NEUES PALAIS, HAUS 22, INSTITUT FÜR MATHEMATIK" value="52.400014 13.010041" />
106 <entry key="NEUES PALAIS, HAUS 22, INSTITUT FÜR PHYSIK" value="52.400014 13.010041" />
107 <entry key="NEUES PALAIS, HAUS 22, ZENTRALE STUDIENBERATUNG STUDIERENDENMARKETING UND ALUMNIPROGRAMM" value="52.400014 13.010041" />
108 <entry key="NEUES PALAIS, HAUS 2, POSTSTELLE" value="52.400037 13.010533" />
109 <entry key="NEUES PALAIS, HAUS 2, POTSDAM GRADUATE SCHOOL" value="52.400037 13.010533" />
110 <entry key="NEUES PALAIS, HAUS 2, STUIDUMPLUS" value="52.400037 13.010533" />
111 <entry key="NEUES PALAIS, HAUS 2, ZENTRALE ABTEILUNG / AUDIOVISUELLES ZENTRUM" value="52.400037 13.010533" />
112 <entry key="NEUES PALAIS, HAUS 1, DEZERNAT 1 PLANUNG STATISTIK FORSCHUNGSANGELEGENHEITEN ORGANISATION" value="52.399583 13.010777" />
113 <entry key="NEUES PALAIS, HAUS 1, INSTITUT FÜR KÜNSTE UND MEDIEN" value="52.399583 13.010777" />
114 <entry key="NEUES PALAIS, HAUS 1, INSTITUT FÜR SLAVISTIK" value="52.399583 13.010777" />
115 <entry key="NEUES PALAIS, HAUS 8, AUDITORIUM MAXIMUM AUDIMAX" value="52.399793 13.011915" />
116 <entry key="NEUES PALAIS, HAUS 8, BÜRO DER FRANZÖSISCHEN BOTSCHAFT FÜR DIE AKADEMISCHEN BEZIEHUNGEN" value="52.399793 13.011915" />
117 <entry key="NEUES PALAIS, HAUS 8, CAFETERIA" value="52.399793 13.011915" />
118 <entry key="NEUES PALAIS, HAUS 8, DEZERNAT 2 STUDENTISCHE ANGELEGENHEITEN - ZENTRALE STUDIENBERATUNG, STUDIERENDENSEKRETARIAT, PRÜFUNGSAMT, AKADEMISCHES AUSLANDSAMT" value="52.399793 13.011915" />
119 <entry key="NEUES PALAIS, HAUS 8, DEZERNAT 2 CHIPKARTENSERVICE" value="52.399793 13.011915" />
120 <entry key="NEUES PALAIS, HAUS 8, INSTITUT FÜR MATHEMATIK" value="52.399793 13.011915" />
121 <entry key="NEUES PALAIS, HAUS 8, WELCOME CENTER POTSDAM"

122 value="52.399793 13.011915" />
<entry key="NEUES PALAIS, HAUS 8, ZENTRALE EINRICHTUNG FÜR
INFORMATIONSVERRARBEITUNG UND KOMMUNIKATION" value="52.399793
13.011915" />
123 <entry key="NEUES PALAIS, HAUS 19, INSTITUT FÜR ANGLISTIK /
AMERIKANISTIK" value="52.398327 13.011939" />
124 <entry key="NEUES PALAIS, HAUS 19, INSTITUT FÜR ROMANISTIK"
value="52.398327 13.011939" />
125 <entry key="NEUES PALAIS, HAUS 19, ZENTRUM FÜR SPRACHEN UND
SCHLÜSSELKOMPETENZEN ZEISSKO" value="52.398327 13.011939" />
126 <entry key="NEUES PALAIS, SPORTPLATZ" value="52.401566
13.007550" />
127 <entry key="GRIEBNITZSEE, HPI HASSO-PLATTNER-INSTITUT
HAUPTGEÄUDE" value="52.394002 13.133376" />
128 <entry key="GRIEBNITZSEE, HPI HASSO-PLATTNER-INSTITUT
HÖRSÄLE" value="52.394022 13.132685" />
129 <entry key="GRIEBNITZSEE, HPI HASSO-PLATTNER-INSTITUT HÄUSER
A B C" value="52.393516 13.131932" />
130 <entry key="GRIEBNITZSEE, HAUS 5, UNIVERSITÄTSBIBLIOTHEK"
value="52.393950 13.130785" />
131 <entry key="GRIEBNITZSEE, HAUS 4, INSTITUT FÜR INFORMATIK"
value="52.393552 13.130061" />
132 <entry key="GRIEBNITZSEE, HAUS 7" value="52.393160 13.129576"
/>
133 <entry key="GRIEBNITZSEE, HAUS 3, FSR POLITIK VERWALTUNG"
value="52.393590 13.129226" />
134 <entry key="GRIEBNITZSEE, HAUS 3, FSR SOZIOLOGIE"
value="52.393590 13.129226" />
135 <entry key="GRIEBNITZSEE, HAUS 3, FSR
WIRTSCHAFTSWISSENSCHAFTEN" value="52.393590 13.129226" />
136 <entry key="GRIEBNITZSEE, HAUS 3, HOCHSCHULGEBÄUDEMANAGEMENT
HGP BEREICH UNIVERSITÄT POTSDAM - BABELSBERG"
value="52.393590 13.129226" />
137 <entry key="GRIEBNITZSEE, HAUS 3, WIRTSCHAFTS- UND
SOZIALWISSENSCHAFTLICHE FAKULTÄT" value="52.393590
13.129226" />
138 <entry key="GRIEBNITZSEE, HAUS 2, WIRTSCHAFTS- UND
SOZIALWISSENSCHAFTLICHE FAKULTÄT" value="52.393718
13.128451" />
139 <entry key="GRIEBNITZSEE, HAUS 6, HÖRSAALGEBÄUDE"
value="52.392997 13.128666" />
140 <entry key="GRIEBNITZSEE, HAUS 6, STUDIENBÜRO DER
JURISTISCHEN FAKULTÄT" value="52.392997 13.128666" />
141 <entry key="GRIEBNITZSEE, HAUS 6, ZENTRUM FÜR SPRACHEN UND
SCHLÜSSELKOMPETENZEN" value="52.392997 13.128666" />
142 <entry key="GRIEBNITZSEE, HAUS 1, ELTERN-KIND-RAUM"
value="52.392733 13.127136" />
143 <entry key="GRIEBNITZSEE, HAUS 1, HÖRSÄLE UND SEMINARRÄUME"
value="52.392733 13.127136" />
144 <entry key="GRIEBNITZSEE, HAUS 1, JURISTISCHE FAKULTÄT"
value="52.392733 13.127136" />
145 <entry key="GRIEBNITZSEE, HAUS 1, MENSA" value="52.392733
13.127136" />
146 <entry key="GRIEBNITZSEE, HAUS 1, MENSCHENRECHTSZENTRUM"
value="52.392733 13.127136" />
147 <entry key="GRIEBNITZSEE, HAUS 1, WIRTSCHAFTS- UND
SOZIALWISSENSCHAFTLICHE FAKULTÄT" value="52.392733
13.127136" />
148 <entry key="GRIEBNITZSEE, HAUS 1, ZENTRALE EINRICHTUNG FÜR
INFORMATIONSVERRARBEITUNG UND KOMMUNIKATION ZEIK"
value="52.392733 13.127136" />
149 <entry key="BABELSBERG, HAUS 1 - STUDENTENWOHNHEIM"
value="52.4068 13.09953" />
150 <entry key="BABELSBERG, HAUS 2 - STUDENTENWOHNHEIM"
value="52.40647 13.09906" />
151 <entry key="BABELSBERG, HAUS 3 - STUDENTENWOHNHEIM"
value="52.40688 13.09823" />
152 <entry key="BABELSBERG, HAUS 4 - STUDENTENWOHNHEIM"

```
153         value="52.40574 13.098487" />
154     <entry key="BABELSBERG, HAUS 5 - GESCHÄFTSSTELLE MASTER OF
155     PUBLIC MANAGEMENT" value="52.405998 13.098487" />
156     <entry key="BABELSBERG, HAUS 6" value="52.406194 13.09739" />
157     <entry key="BABELSBERG, HAUS 7 - KOMMUNALWISSENSCHAFTLICHES
158     INSTITUT" value="52.406163 13.098106" />
159     <entry key="POSTHOFSTRASSE, NR. 11" value="52.398776
160     13.063231" />
161     <entry key="POSTHOFSTRASSE, NR. 15" value="52.398791
162     13.063269" />
163     <entry key="BOTANISCHER GARTEN, NR. 1" value="52.403528
164     13.026898" />
165     <entry key="BOTANISCHER GARTEN, NR. 2" value="52.403759
166     13.026243" />
167     <entry key="BOTANISCHER GARTEN, NR. 2A" value="52.403671
168     13.026342" />
169     <entry key="NEXTBIKE FAHRRADVERLEIH, BAHNHOF CHARLOTTENHOF
170     #5203" value="52.39289 13.03661" />
171     <entry key="NEXTBIKE FAHRRADVERLEIH, BAHNHOF GOLM #5201"
172     value="52.40901 12.97243" />
173     <entry key="NEXTBIKE FAHRRADVERLEIH, BAHNHOF GRIEBNITZSEE
174     #5221" value="52.39398 13.12901" />
175     <entry key="NEXTBIKE FAHRRADVERLEIH, BAYRISCHES HAUS / IM
176     WILDPARK" value="52.37298 12.98361" />
177     <entry key="NEXTBIKE FAHRRADVERLEIH, BRANDENBURGER STR/
178     LINDENSTR #5209" value="52.39982 13.05187" />
179     <entry key="NEXTBIKE FAHRRADVERLEIH, DORTUSSTR/
180     CHARLOTTENSTRASSE #5210" value="52.3992 13.05295" />
181     <entry key="NEXTBIKE FAHRRADVERLEIH, FILMMUSEUM #5215"
182     value="52.39524 13.05844" />
183     <entry key="NEXTBIKE FAHRRADVERLEIH, HAUPTBAHNHOF NORD #5211"
184     value="52.39288 13.06601" />
185     <entry key="NEXTBIKE FAHRRADVERLEIH, HAUPTBAHNHOF SÜD/
186     TAXISTAND #5217" value="52.39093 13.06712" />
187     <entry key="NEXTBIKE FAHRRADVERLEIH, HOCHSCHULE FÜR FILM UND
188     FERNSEHEN #5223" value="52.38833 13.11761" />
189     <entry key="NEXTBIKE FAHRRADVERLEIH, JÄGERTOR/ LINDENSTRASSE
190     #5208" value="52.40212 13.05242" />
191     <entry key="NEXTBIKE FAHRRADVERLEIH, KIRSCHALLEE/PAPPELALLEE
192     #5204" value="52.41384 13.03257" />
193     <entry key="NEXTBIKE FAHRRADVERLEIH, KOPERNIKUSSTRASSE #5219"
194     value="52.38917 13.09995" />
195     <entry key="NEXTBIKE FAHRRADVERLEIH, LUISENPLATZ SÜD/
196     SPARKASSE #5207" value="52.39888 13.04758" />
197     <entry key="NEXTBIKE FAHRRADVERLEIH, NEUES PALAIS/
198     UNIVERSITÄT #5202" value="52.40182 13.01094" />
199     <entry key="NEXTBIKE FAHRRADVERLEIH, PLANTAGENPLATZ #5220"
200     value="52.39429 13.10246" />
201     <entry key="NEXTBIKE FAHRRADVERLEIH, PLATZ DER EINHEIT #5214"
202     value="52.3992 13.05798" />
203     <entry key="NEXTBIKE FAHRRADVERLEIH, RATHAUS BABELSBERG
204     #5218" value="52.39225 13.09172" />
205     <entry key="NEXTBIKE FAHRRADVERLEIH, SCHIFFBAUERGASSE #5216"
206     value="52.40381 13.07515" />
207     <entry key="NEXTBIKE FAHRRADVERLEIH, STADTHAUS POTSDAM #5212"
208     value="52.40575 13.05813" />
209     <entry key="NEXTBIKE FAHRRADVERLEIH, VOLKSPARK SÜD #5205"
210     value="52.41586 13.04121" />
211     <entry key="NEXTBIKE FAHRRADVERLEIH, WALDSTADT/HEINRICH-MANN-
212     ALLEE #5213" value="52.36602 13.09359" />
213     <entry key="NEXTBIKE FAHRRADVERLEIH, ZENTRUM OST/HUMBOLDTRING
214     #5222" value="52.39486 13.07644" />
215     <entry key="NEXTBIKE FAHRRADVERLEIH, ZEPPELINSTR/ BREITE STR.
216     #5206" value="52.39689 13.04363" />
217 </map>
218 </property>
219 </bean>
```

9.2 ANHANG B – GRAPH-CONFIG.XML

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xmlns:context="http://www.springframework.org/schema/context"
5     xsi:schemaLocation="http://www.springframework.org/schema/beans
6         http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
7         http://www.springframework.org/schema/context
8         http://www.springframework.org/schema/context/spring-context-2.5.xsd">
9
10     <bean id="gtfsBuilder"
11         class="org.opentripplanner.graph_builder.impl.GtfsGraphBuilderImpl">
12         <property name="gtfsBundles">
13             <bean id="gtfsBundles"
14                 class="org.opentripplanner.graph_builder.model.GtfsBundles">
15                 <property name="bundles">
16                     <list>
17                         <bean
18                             class="org.opentripplanner.graph_builder.model.
19                             GtfsBundle">
20                             <!-- Pfad GTFS -->
21                             <property name="path" value="C:\Master\otp-
22                                 data\gtfs\20641.zip" />
23                             <property name="defaultBikesAllowed" value="true"
24                                 />
25                         </bean>
26                     </list>
27                 </property>
28             </bean>
29         </property>
30     </bean>
31
32     <bean id="osmBuilder"
33         class="org.opentripplanner.graph_builder.impl.osm.
34         OpenStreetMapGraphBuilderImpl">
35         <property name="provider">
36             <!-- dieses bean verwenden, um eine lokale OSM-Datei zu laden -->
37             <bean class="org.opentripplanner.openstreetmap.impl.
38                 AnyFileBasedOpenStreetMapProviderImpl">
39                 <property name="path" value="C:\Master\otp-
40                     data\osm\pdm081214.osm" />
41             </bean>
42         </property>
43         <property name="defaultWayPropertySetSource">
44             <bean class="org.opentripplanner.graph_builder.impl.osm.
45                 DefaultWayPropertySetSource" />
46         </property>
47     </bean>
48
49     <!-- Bean für DGM! als GeoTIFF einbinden, unten den NED Builder noch
50     aktivieren! -->
51     <bean id="nedBuilder"
52         class="org.opentripplanner.graph_builder.impl.ned.NEDGraphBuilderImpl">
53         <property name="gridCoverageFactory">
54             <bean
55                 class="org.opentripplanner.graph_builder.impl.ned.
56                 GeotiffGridCoverageFactoryImpl">
57                 <property name="path" value="C:\Master\otp-
58                     data\DGM\dgm_potsdam_ProjectRaster.tif" />
59             </bean>
60         </property>
61     </bean>
62
63     <bean id="floatingIslands"
64         class="org.opentripplanner.graph_builder.impl.PruneFloatingIslands">
65         <property name="islandWithoutStopsMaxSize" value="1000" />
66     </bean>
67 </beans>
```

```
46     <property name="islandLogFile" value="{graphPath}/island.csv"/>
47     <property name="islandWithStopsMaxSize" value="20"/>
48     <property name="transitToStreetNetwork" ref ="transitStreetLink"/>
49 </bean>
50
51 <bean id="transitStreetLink"
52     class="org.opentripplanner.graph_builder.impl.
53     TransitToStreetNetworkGraphBuilderImpl" />
54
55 <bean id="graphBuilderTask"
56     class="org.opentripplanner.graph_builder.GraphBuilderTask">
57     <!-- Hier soll der Graph gespeichert werden -->
58     <property name="path" value="C:\Master\otp-data\graph" />
59     <property name="graphBuilders">
60     <list>
61     <!-- GTFS comes before OSM, because we use the loaded set of
62     stops to determine our OSM coverage -->
63     <ref bean="gtfsBuilder" />
64     <ref bean="osmBuilder" />
65     <ref bean="nedBuilder" />
66     <!-- the transmit link should became before the floating
67     islands in order to find islands with stops-->
68     <ref bean="transitStreetLink" />
69     <ref bean="floatingIslands" />
70     </list>
71     </property>
72 </bean>
73 </beans>
```

Quellcode 8 - Konfigurationsdatei zur Erzeugung des Graph