



## School of Engineering

InIT Institut für angewandte  
Informationstechnologie

### **Projektarbeit (Software-Engineering)**

Bereitstellung von Strassen- und Verkehrsdaten  
für eine innovative Park-Applikation

---

<b>Autoren</b>	Manuel Simbürger Roman Thöni
----------------	---------------------------------

---

<b>Hauptbetreuung</b>	Mark Cieliebak
-----------------------	----------------

---

<b>Datum</b>	18.12.2014
--------------	------------

# Zusammenfassung

Der Verkehr in Grossstädten wird zunehmend ein Problem und neue Technologien könnten eine Antwort darauf sein. Die School of Engineering (SoE) hat eine Verkehrssimulation entwickelt, mit welcher Suchstrategien für Parkplätze getestet werden können. Ziel dieser Arbeit ist es, realitätsnahe Daten für diese Simulation zu beschaffen und aufzubereiten. Zu diesem Zweck wurden verschiedenste Bezugsquellen wie Programmierschnittstellen und Datenlieferanten analysiert und bewertet.

Im Rahmen dieser Arbeit wurde ein Programm zum Import von Verkehrs- und Strassendaten entwickelt. Zusätzlich wurden weitere Datenquellen gefunden, welche als alternative zu Datenlieferanten genutzt werden könnten. Für die beiden Programmierschnittstellen von Google und TomTom wurden zu diesem Zweck auch Prototypen zum Auslesen der entsprechenden Daten entwickelt.

Während im Bereich der Strassen- und Verkehrsdaten qualitativ hochwertige Daten beschafft werden konnten, mangelt es vor allem im Bereich der Parkplätze noch an verlässlichen Daten. Nach wie vor ist jedoch nicht klar, welche Organisationen effektiv über vollständige Parkplatzdaten verfügen.

# Abstract

Traffic in urban areas is a growing problem and new technologies might hold the answer. The School of Engineering has developed a traffic simulation to evaluate different search strategies for finding parking spaces. The goal of this research project is to acquire and refine realistic data for said simulation. Various sources of supply and programming interfaces have been analyzed and rated for this purpose.

In the scope of this research project a application for importing street and traffic data has been developed. In addition alternative data sources have been found which may be used as replacement to data suppliers. For the following two programming interfaces of Google and TomTom prototypes to access the specified data were developed.

Whereas the amount and quality in the area of street and traffic data acquired is mostly excellent, there is a huge lack of quality data in the area of parking spaces. The main reason is the fact that it is still uncertain which organizations hold quality data in the area of parking spaces.

# Vorwort

Simulationen, die der Realität möglichst nahe kommen sollen sind nur so gut, wie die zu Grunde liegenden Daten. Es gibt eine grosse Menge Kartenmaterial und zahlreiche Medien berichten über Verkehrsstörungen. Sehr viele Fahrzeuge sind mit GPS ausgerüstet und Verkehrskameras überwachen einen Grossteil unserer Strassen. Im Rahmen dieser Arbeit wollen wir diese verschiedenen Quellen analysieren und zusammenführen um sie für eine vorhandene Parkplatz-Simulation nutzbar zu machen. Während des Projekts mussten wir jedoch feststellen, dass oft nicht klar ist, wer über welche Daten verfügt oder dass deutlich weniger Daten tatsächlich öffentlich verfügbar sind, als man auf den ersten Blick denken könnte. Dank der kompetenten Betreuung von Mark Cieliebak und der technischen Unterstützung in Fragen der Simulation durch Alex Werffeli, konnten wir für viele der Problemstellungen gute Lösungen finden. Wir möchten Ihnen an dieser Stelle unseren herzlichen Dank aussprechen.

## Erklärung betreffend das selbständige Verfassen einer Projektarbeit an der School of Engineering

Mit der Abgabe dieser Projektarbeit versichert der/die Studierende, dass er/sie die Arbeit selbständig und ohne fremde Hilfe verfasst hat. (Bei Gruppenarbeiten gelten die Leistungen der übrigen Gruppenmitglieder nicht als fremde Hilfe.)

Der/die unterzeichnende Studierende erklärt, dass alle zitierten Quellen (auch Internetseiten) im Text oder Anhang korrekt nachgewiesen sind, d.h. dass die Projektarbeit keine Plagiate enthält, also keine Teile, die teilweise oder vollständig aus einem fremden Text oder einer fremden Arbeit unter Vorgabe der eigenen Urheberschaft bzw. ohne Quellenangabe übernommen worden sind.

Bei Verfehlungen aller Art treten die Paragraphen 39 und 40 (Unredlichkeit und Verfahren bei Unredlichkeit) der ZHAW Prüfungsordnung sowie die Bestimmungen der Disziplinarmaßnahmen der Hochschulordnung in Kraft.

Ort, Datum:

Unterschriften:

.....

.....

.....

.....

Das Original dieses Formulars ist bei der ZHAW-Version aller abgegebenen Projektarbeiten zu Beginn der Dokumentation nach dem Abstract bzw. dem Management Summary mit Original-Unterschriften und -Datum (keine Kopie) einzufügen.

# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>6</b>
1.1. Ausgangslage . . . . .	6
1.2. Zielsetzung . . . . .	6
1.3. Abgrenzung . . . . .	6
1.4. Begriffserklärung . . . . .	7
<b>2. Grundlagen</b>	<b>8</b>
2.1. Bachelorarbeit Informatik - Placebook . . . . .	8
2.2. Weiterführung des Projektes . . . . .	9
<b>3. Recherche</b>	<b>10</b>
3.1. Standardformat für Geodaten . . . . .	10
3.2. Weitere potenzielle Quellen . . . . .	10
3.3. Stadt Zürich . . . . .	11
3.4. TomTom . . . . .	11
3.5. OpenStreetMap . . . . .	11
3.6. Programmierschnittstellen . . . . .	11
3.6.1. TomTom Routes . . . . .	11
3.6.2. Google Maps . . . . .	12
3.6.3. OpenStreetMap . . . . .	12
<b>4. Resultate</b>	<b>13</b>
4.1. Placebook MapImporter . . . . .	13
4.1.1. Strassendaten . . . . .	13
4.1.2. Parkplätze . . . . .	14
4.1.3. Geschwindigkeiten . . . . .	15
4.1.4. Simulationselemente . . . . .	15
4.2. TomTom-Daten . . . . .	16
4.3. OSM-Daten . . . . .	17
4.4. API Tools . . . . .	19
4.4.1. Google Distance Matrix . . . . .	19
4.4.2. TomTom Routes . . . . .	19
<b>5. Diskussion und Ausblick</b>	<b>21</b>
<b>6. Verzeichnisse</b>	<b>22</b>
Literaturverzeichnis . . . . .	22
Abbildungsverzeichnis . . . . .	24
Listingverzeichnis . . . . .	25
<b>A. Anhang</b>	<b>I</b>
A.1. Offizielle Aufgabenstellung . . . . .	I
A.2. Inhaltsverzeichnis CD . . . . .	III
A.3. Kontaktliste . . . . .	III
A.4. Verwendete Software . . . . .	IV
A.5. Netzwerkverkehrsanalyse routes.tomtom.com . . . . .	IV

# 1. Einleitung

## 1.1. Ausgangslage

Die SoE (School of Engineering) verfügt über eine Simulation für den Parkplatzsuchverkehr in Städten. Während die grundlegende Logik der Simulation bereits funktioniert, wird noch eine stark vereinfachte Karte mit zufällig erzeugten Geschwindigkeiten und Parkplätzen verwendet. Für die Weiterführung des Projektes sind nun realistische Daten erforderlich.

Moderne Navigationsgeräte kennen die aktuelle Verkehrssituation und sind in der Lage, Verspätungen auf der gewählten Routen vorherzusagen. Navigationsgerätehersteller TomTom bietet auf seiner Webseite die Möglichkeit, die zu erwartende Verspätung pro Wochentag und Uhrzeit zu berechnen. Die Daten stammen hierbei von den Navigationsgeräten im Einsatz, die regelmässig Informationen über Position und Geschwindigkeit an den Hersteller übermitteln. Trotz der zahlreichen Anwendungen und Produkte die auf solchen Informationen basieren, sind solche Daten nicht oder nur schwer zugänglich.

Für die leichtere Suche nach Parkplätzen existieren mehrere Mobil-Applikationen wie zum Beispiel Parkonaut oder Parknav. Beide Applikationen basieren darauf, dass Benutzer freie oder frei werdende Parkplätze eintragen, welche dann anderen Benutzern angezeigt werden können. Für eine gute Abdeckung und Datenqualität ist demnach eine grosse Anzahl Benutzer notwendig. Öffentlich verfügbar sind die Daten jedoch oft nicht. Grund dafür dürften Konkurrenz Apps sein, die durch diese Daten dann eine deutlich höhere Abdeckung erreichen könnten.

Öffentliche Daten im Bereich Strassen und Parkplätze sind auf OpenStreetMap (OSM) verfügbar. Die Daten stammen zum grössten Teil von Privatpersonen und vereinzelt von Firmen, die mit GPS Geräten Strassen abfahren oder manuell wichtige Zusatzinformationen wie Parkplätze oder wichtige Gebäude eintragen. Die Datenqualität der Strassen ist gut, variiert jedoch im Bereich der Zusatzinformationen und Metainformationen sehr stark.

Verkehrs- und Strassendaten sind wichtig für Städte und Strassenplaner. Mit Hilfe dieser Daten kann der Verkehrsfluss in Städten optimiert und somit die Verkehrsbelastung reduziert werden. Insbesondere können mit Hilfe von Simulationen Studien zum Verkehrsfluss angestellt werden und wie dieser sich beim Einsatz bestimmter Technologien wie Parking Apps verändert.

## 1.2. Zielsetzung

Das Ziel dieser Arbeit ist es, digitale Verkehrs- und Strassendaten zu beschaffen und so aufzubereiten, dass sie in der bereits bestehenden Verkehrssimulation der SoE verwendet werden können. Mindestens von einer Stadt sollen komplette Daten inklusive Parkplätzen vorhanden sein, wobei die Aktualität der Datensätze in einem ersten Schritt eine untergeordnete Rolle spielt. Die offizielle Aufgabenstellung befindet sich im Anhang dieser Arbeit.

## 1.3. Abgrenzung

Die Datenbeschaffung ist auf Daten bezüglich Strassen und Verkehr beschränkt. Dabei sind lediglich die Geschwindigkeit zu bestimmten Tageszeiten und die Strassenverläufe gemeint. Es werden keine Daten über einzelne Fahrzeuge oder die Auslastung von Strassen oder Parkhäusern beschafft. Das

Gebiet wurde für diese Arbeit auf den Kanton Zürich mit speziellem Fokus auf die Städte Zürich und Winterthur eingeschränkt.

## 1.4. Begriffserklärung

### **Shape Datei**

Shape-Dateien dienen zur Speicherung von nicht topologischen Geometriedaten für räumliche Kenn-  
daten. Die Geometrie wird dabei als Gruppe von Vektor-Koordinaten gespeichert. Unterstützt werden  
hierbei Punkte, Linien und Polygon-Flächen.<sup>[1]</sup>

### **Verkehrsdaten**

Unter dem Begriff Verkehrsdaten wird in dieser Arbeit die Geschwindigkeit auf einem Strassenabschnitt  
während einer bestimmten Zeitdauer verstanden. Daten wie die Anzahl der Fahrzeuge oder der Fahr-  
zeugtyp, sowie Ziel und Herkunft einzelner Fahrzeuge werden nicht berücksichtigt.

### **Strassendaten**

Als Strassendaten werden in dieser Arbeit Strassenverläufe und Kreuzungspunkte verstanden. Die  
Fahrtrichtung sowie Abbiegerestriktionen werden nicht berücksichtigt.

## 2. Grundlagen

Als Grundlage für diese Projektarbeit hat die letztjährige Bachelorarbeit Placebook<sup>[23]</sup>, welche im Rahmen eines interinstitutionellen Sonderprojektes der School of Engineering (SoE) umgesetzt wurde, gedient. Die Bachelorarbeit wurde in Kooperation mit dem Studiengang Verkehrssysteme realisiert. Der daraus entstandene Lösungsansatz wurde im vergangenen halben Jahr parallel zu dieser Projektarbeit weiter ausgebaut.

### 2.1. Bachelorarbeit Informatik - Placebook

#### Grundlage der Bachelorarbeit

Zu Beginn der Bachelorarbeit Placebook befand sich das gleichnamige Projekt der SoE noch in der Konzeptphase. Das entstandene Konzept beinhaltet eine mobile Applikation, welche anhand durch die Benutzer generierter Daten die Parkplatzsuche erleichtern soll. Wird ein Parkplatz frei, so soll dies mittels Placebook angegeben werden können und jedem Benutzer, der in der Nähe einen Parkplatz sucht, angezeigt werden.

#### Aufgabenstellung

Ziel der Bachelorarbeit war neben der Erweiterung des bestehenden Grundkonzeptes für Placebook und der Erstellung eines Demonstrators zur Visualisierung des entstandenen Prototyps auch eine Recherche des aktuellen technischen Stands und die Evaluierung der Grenzen des so entwickelten Konzeptes.

#### Resultate

Im Verlauf der Bachelorarbeit Placebook ist ein vierteiliger Prototyp zur Echtzeitsimulation des Parkplatzsuchverkehrs entstanden. Neben der eigentlichen Applikation und dem Backend in Java wurde im Studiengang Verkehrssysteme eine Simulation in MATLAB umgesetzt. Die Simulation kann dabei durch eine RESTful Schnittstelle angesteuert werden.

Zu Beginn war die Simulation mit strikt rechtwinkligen Karten und zufällig gewählten Werten für Geschwindigkeit und Parkplätzen ausgestattet.

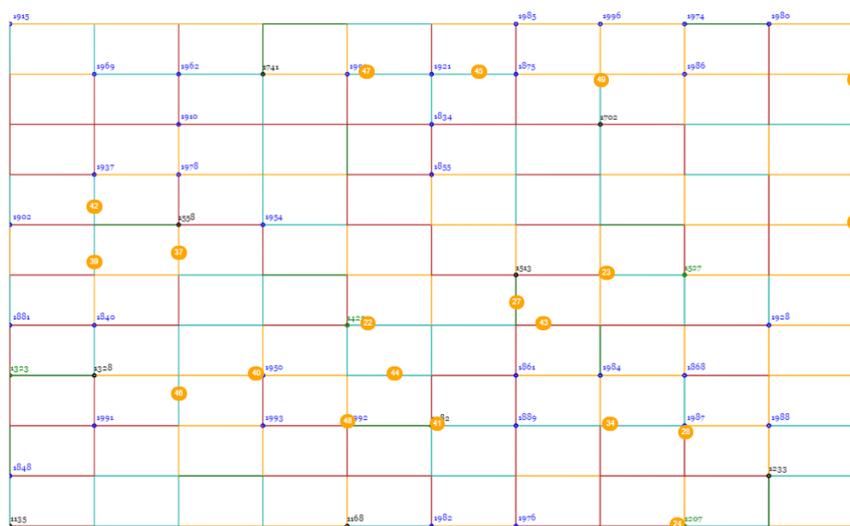


Abbildung 2.1.: Placebook Simulation - Rechtwinklige Karte

## 2.2. Weiterführung des Projektes

Nach Abschluss der Bachelorarbeit Placebook wurde das Projekt innerhalb des Instituts für angewandte Informationstechnologie weitergeführt. Auf Grund der Tatsache, dass bereits eine Applikation mit praktisch derselben Funktionalität wie Placebook auf dem Markt ist, wurde entschieden sich ganz auf die Simulation des Parkplatzsuchverkehrs zu konzentrieren. Der bisher in MATLAB umgesetzte Teil der Simulation wurde mit Hilfe des Play Frameworks vollständig nach Java portiert.

Als nächsten Schritt mussten reale Geschwindigkeits- und Parkplatzdaten beschafft werden und mittels einer Import-Klasse zur Verfügung gestellt werden. Zu diesem Zweck wurde die Projektarbeit ausgeschrieben und bearbeitet.

## 3. Recherche

Im ersten Schritt dieser Arbeit wurden Informationen über mögliche Herkunft und Qualität von Verkehrs- und Strassendaten gesammelt. Dieses Kapitel beinhaltet Ergebnisse und Analysen der verschiedenen Quellen.

### 3.1. Standardformat für Geodaten

Um den Austausch und die Bearbeitung der Verkehrs- und Strassendaten zu vereinfachen, wurde in der Anfangsphase nach einem Standardformat für Geodaten gesucht. Ein international anerkanntes Standardformat für raumbezogene Daten ist nicht verfügbar. Jedoch ist das von der Firma ESRI<sup>1</sup> Inc. entwickelte Dateiformat *Shapefile* ein de facto Standard für Geodaten. Das Dateiformat besteht aus mindestens drei Dateien:

- .shp Geometriedaten
- .dbf Zusatzdaten im dBase Format
- .shx Index für die Geometriedaten und Verknüpfung mit den Zusatzdaten

Die vollständige technische Beschreibung des Dateiformates befindet sich auf der CD im Anhang dieser Arbeit.

### 3.2. Weitere potenzielle Quellen

Die Sendung *Einstein-Spezial* zum Thema "Big Data", gesendet am 2.10.2014 vom Schweizer Fernsehen. Im Beitrag "Verkehrsmeldungen: im Netz der Bewegungsdaten" wurden die drei Organisationen INRIX, Viasuisse und Lost n' Found erwähnt. INRIX und Viasuisse verfügen laut Beitrag über Verkehrs- und Verkehrsflussinformationen sowie Verkehrsfluss-Störungen. Die Firma Lost n' Found ist Anbieter für Flottenmanagement Lösungen und verfügt über die GPS-Tracker in Firmenflotten über Standorte und Geschwindigkeiten von mehreren tausend Fahrzeugen. Bei der Firma INRIX wurde ein Testzugang<sup>[16]</sup> für die angebotene Programmierschnittstelle angelegt. Die Schnittstelle lieferte für die Städte San-Francisco und Hamburg die aktuelle Geschwindigkeit für ausgewählte Strassen. Mehrere Nachfragen bezüglich eines Vollzugangs für Forschungszwecke wurden nicht beantwortet. Zur Einsicht in die Dokumentation der Schnittstelle muss man über ein Konto verfügen, dieses kann kostenlos erstellt werden. Viasuisse bestätigte die Verfügbarkeit der Daten, die jedoch von der Firma INRIX bezogen wurden und deshalb aus rechtlichen Gründen nicht weitergegeben werden durften. Die Firma Lost n' Found bestätigte, dass Datensätze über Position und Geschwindigkeit von einzelnen Fahrzeugen vorliegen, die Daten aber nicht sortiert wären und nicht ohne erheblichen Aufwand zur Verfügung gestellt werden könnten.

---

<sup>1</sup>Environmental Systems Research Institute

### 3.3. Stadt Zürich

Das Tiefbauamt der Stadt Zürich veröffentlichte im Jahr 2010 eine Studie über das Verkehrsaufkommen von Parkfeldern<sup>[2]</sup> und 2011 einen Artikel über die Parkplatzbelegung<sup>[3]</sup>. In beiden Veröffentlichungen finden sich ungefähre Angaben über Parkplätze und die Anzahl Parkfelder, was vermuten lässt, dass noch detailliertere Daten verfügbar sind. Die telefonischen und schriftlichen Anfragen an das Tiefbauamt und weitere Ämter wie die Dienstabteilung Verkehr führten jedoch zu keinem Ergebnis. Eine vollständige Liste der kontaktierten Ämter und Personen befindet sich im Anhang dieser Arbeit.

### 3.4. TomTom

Im Verlauf der Projektarbeit wurden mehrere Navigationsgeräte-Hersteller angefragt. Leider hat darauf lediglich TomTom geantwortet und bezüglich weiterer Abklärungen an ihren Schweizer Distributor der Daten die **bbp geomatik ag** verwiesen. Nach allen nötigen Abklärungen wurde im Rahmen dieser Projektarbeit ein Vertrag unterzeichnet und die Geschwindigkeitsdaten wurden mittels Download zur Verfügung gestellt. Die genaue Beschreibung der Daten ist dem Abschnitt 4.2 zu entnehmen

### 3.5. OpenStreetMap

Als alternative Datenquelle zu den angefragten Navigationsgeräte-Herstellern wurde zusätzlich noch OpenStreetMap analysiert. Bei OpenStreetMap handelt es sich um ein Open Data Projekt mit dem Ziel frei verfügbaren Weltkarte. Registrierte Benutzer können selbstständig Kartenmaterial und dazugehörige Daten verfassen. Hierfür stehen verschiedenste sogenannte Tags zur Verfügung, zum Beispiel "highway" für eine Strasse oder "parking" für einen Parkplatz. Die genaue Funktionsweise kann der Dokumentation im Anhang entnommen werden. Zu Beginn der Projektarbeit wurden die Kartenabschnitte aus OSM Daten generiert, dies hat sich jedoch schnell als nicht praktikabel herausgestellt, da unter anderem benötigte Attribute wie die erlaubte Höchstgeschwindigkeit nur sehr sporadisch eingetragen sind. Dies ist vor allem darauf zurück zu führen, dass es sich bei OpenStreetMap um Benutzer generierten Inhalt handelt. Allgemein variiert die Qualität beziehungsweise die Vollständigkeit und der Detaillierungsgrad je nach Region. In urbanisierten Gebieten sind mehr Daten verfügbar als in ländlichen, wo zum Teil ganze Ortschaften nicht kartografiert sind.

### 3.6. Programmierschnittstellen

Als potenzielle Datenquellen wurden auch eine Reihe von Programmierschnittstellen in Betracht gezogen. Vier Schnittstellen, die mindestens einen Teil der gesuchten Daten anbieten, sind in den folgenden Abschnitten näher beschrieben. Die weiteren analysierten Schnittstellen, namentlich Microsoft Maps<sup>[7]</sup>, Yahoo Maps<sup>[13]</sup>, Nokia Maps<sup>[8]</sup>, OSM Xapi<sup>[10]</sup> und das Geoportal des Bundes<sup>[4]</sup>, waren alle entweder abgeschaltet oder verfügten nicht über die benötigten Daten oder Funktionen. Für die Schnittstellen von TomTom und Google wurden Programme geschrieben um die Möglichkeiten zu evaluieren. Diese werden im Abschnitt 4.4 näher beschrieben.

#### 3.6.1. TomTom Routes

Die von der TomTom Karten Applikation<sup>[12]</sup> verwendete Schnittstelle bietet insgesamt fünf verschiedene Funktionen an:

1. Karten- und Gebietsinformationen
2. Verkehrsstatus

3. Wichtige Ereignisse
4. Geocoding
5. Navigationsanweisungen

Die Karten- und Gebietsinformationen umfassen beispielsweise politische Grenzen und Ländernamen. Verkehrsstatus und wichtige Ereignisse liefern beide Informationen zu Verkehrsflussstörungen wie Staus oder stockender Verkehr und die Geocoding-Funktionalität übersetzt Koordinaten in Strassen- oder Ortsnamen und umgekehrt. Für diese Arbeit relevant sind vor allem die Navigationsanweisungen. Diese Funktion berechnet die Distanz und die voraussichtlich benötigte Zeit für eine gegebene Strecke. Die berechnete Zeit ist abhängig von der Tageszeit an der die Strecke gefahren wird und berücksichtigt allenfalls vorhandene Verkehrsflussstörungen. Die Daten werden im JSON Format zurückgegeben und beinhalten neben der Fahrzeit und Zeitverlust durch Störungen zusätzlich für jede Teilstrecke die benötigte Zeit in Sekunden und die bis dahin zurückgelegte Strecke. Anhand dieser Daten wäre es möglich, die Geschwindigkeit pro Teilstrecke zu berechnen.

### 3.6.2. Google Maps

Google bietet eine Reihe von Programmierschnittstellen im Zusammenhang mit Karten an. Die zwei Schnittstellen, die Verkehrsdaten verfügen sind *Distance Matrix* und *Directions*. Während die *Directions* API die Route zwischen zwei Punkten berechnet, berechnet *Distance Matrix* in der kostenlosen Variante bis zu 100 Strecken auf einmal. Dies wird erreicht in dem ganze Listen von Start- beziehungsweise Zielorten übergeben werden können. Rückgabewerte sind die Distanz pro berechneter Strecke sowie die benötigte Zeit in Minuten. Somit könnten für längere Strecken zumindest Näherungswerte für die Geschwindigkeit berechnet werden. Für eine realitätsnahe Simulation in Stadtgebieten sind die Werte jedoch zu ungenau.

### 3.6.3. OpenStreetMap

Bei der OpenStreetMap API<sup>[18]</sup> handelt es sich um eine Schnittstelle zur Erfassung und Bearbeitung von OpenStreetMap. Die OpenStreetMap API ist jedoch nicht auf die Auswertung von Daten ausgelegt. Dies wird sogar explizit in der Usage Policy<sup>[20]</sup> unter Androhung der Kontensperrung verboten. Für die reine Datennutzung wird unter anderen auf die unten beschriebene Overpass API verwiesen.

#### Overpass API

Auf die Overpass API<sup>[9]</sup> kann mittels URL zugegriffen werden. Als Output wird ein JSON generiert. Die folgende URL würde zum Beispiel alle Strassen und Knoten welche im Attribut **amenity** den Wert *parking* haben. Dies für eine Bounding-Box im Bereich von Winterthur.

```
http://overpass-api.de/api/interpreter?data=[out:json];(way["amenity"="parking"](47.497331,8.722058,47.502782,8.733902);node["amenity"="parking"](47.497331,8.722058,47.502782,8.733902));out;
```

Nach **data** können für Knoten oder Strassen in einer Bounding-Box beliebige OSM-Attribute abgefragt werden. Zudem kann mit der Anweisung **out:json** angegeben werden, dass das Resultat als JSON zurückgegeben werden soll. Einzelne Teilabfragen können mittels Klammern zu einem einzigen Resultat zusammengefasst werden. Zur einfacheren Handhabung bei der Erstellung von Abfragen kann die Webseite <http://overpass-turbo.eu> genutzt werden.

Im Verlauf der Arbeit hat sich herausgestellt, dass keine Verkehrsdaten und nur sehr sporadisch Geschwindigkeiten eingetragen sind. Deshalb wurde entschieden diese Schnittstelle für Verkehrsdaten nicht zu nutzen. Mangels anderer Quellen wurden zum Schluss die Parkplatzdaten über OpenStreetMap bezogen.

## 4. Resultate

Dieses Kapitel beinhaltet Beschreibungen der erstellten Software sowie Informationen über die erhaltenen Daten oder weiteren potenziellen Quellen.

### 4.1. Placebook MapImporter

Für die Integration der Verkehrs- und Strassendaten in die Simulation der SoE, wurde ein neuer Importer entwickelt. Ein minimales Interface existierte bereits. Die folgenden Abschnitte beschreiben die verschiedenen Teilaufgaben und implementierten Lösungen des Programms. Das Flussdiagramm (Abbildung 4.1) zeigt den Ablauf beim Import einer neuen Karte.

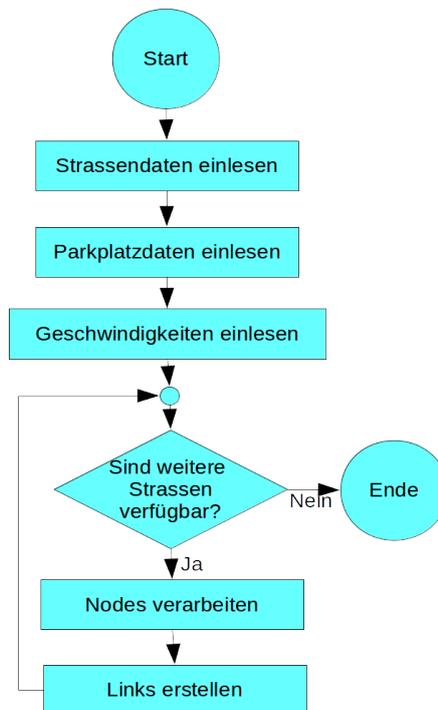


Abbildung 4.1.: MapImporter Flussdiagramm

#### 4.1.1. Strassendaten

Die Grundlage für den Import eines Kartenausschnitts sind die Strassendaten von TomTom, welche im Rahmen dieser Arbeit für den ganzen Kanton Zürich als Shapefile zur Verfügung gestellt wurden. Der Kartenausschnitt kann mittels einer Bounding-Box eingeschränkt werden. Importiert werden sämtliche Strassen, die mindestens einen Knoten im Inneren der Bounding-Box haben. Dieses Vorgehen wurde gewählt, da es die Berechnung der Simulationselemente, vergleiche Abschnitt 4.1.4, stark vereinfacht.

Während dem einlesen wird aus dem Shapefile die Geometrie der Strasse als Gruppe von Koordinaten und die dazugehörige TomTom NetworkID abgespeichert. Das Resultat ist eine unsortierte Liste mit nicht zusammenhängenden Strassen, welche dann mit Geschwindigkeiten und Parkplätzen vervollständigt werden können. Zu einem vollständigen Strassennetz werden die einzelnen Strassen erst beim Import in die Simulation zusammengesetzt. Dies geschieht indem vor dem Einfügen eines Knotens jeweils überprüft wird, ob für eine gegebene Koordinate bereits eine Node in der Simulation existiert.

### 4.1.2. Parkplätze

In der Simulation werden Parkplätze durch eine einzelne Node repräsentiert. Die Parkplatzdaten werden vor der Erstellung der Simulationselemente, vergleiche Abschnitt 4.1.4, eingelesen und in einer HashMap abgelegt. Da die Geodaten von OpenStreetMap und TomTom nicht deckungsgleich sind und zudem Parkhäuser oder grössere Parkplätze auf einem Punkt abseits der Strassen eingetragen sein können, werden einer Node alle Parkplätze im Umkreis von 30 Metern zugewiesen. Dieses Vorgehen kann dazu führen, dass Parkplätze in der Simulation im schlimmsten Fall einer Parallelstrasse zugewiesen werden. Dadurch können in der Simulation leicht längere oder kürzere Fahrzeiten resultieren. Im Mittel sollten sich diese Abweichungen jedoch ausgleichen, weshalb sie vernachlässigt werden können. Behoben werden könnte dieses Problem mit qualitativ besseren Daten für Parkplätze.

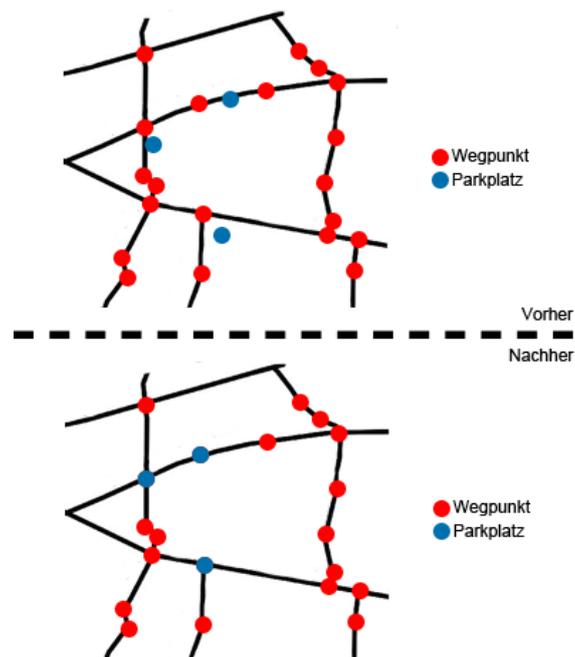


Abbildung 4.2.: Parkplatz Rohdaten und für die Simulation optimiert

### 4.1.3. Geschwindigkeiten

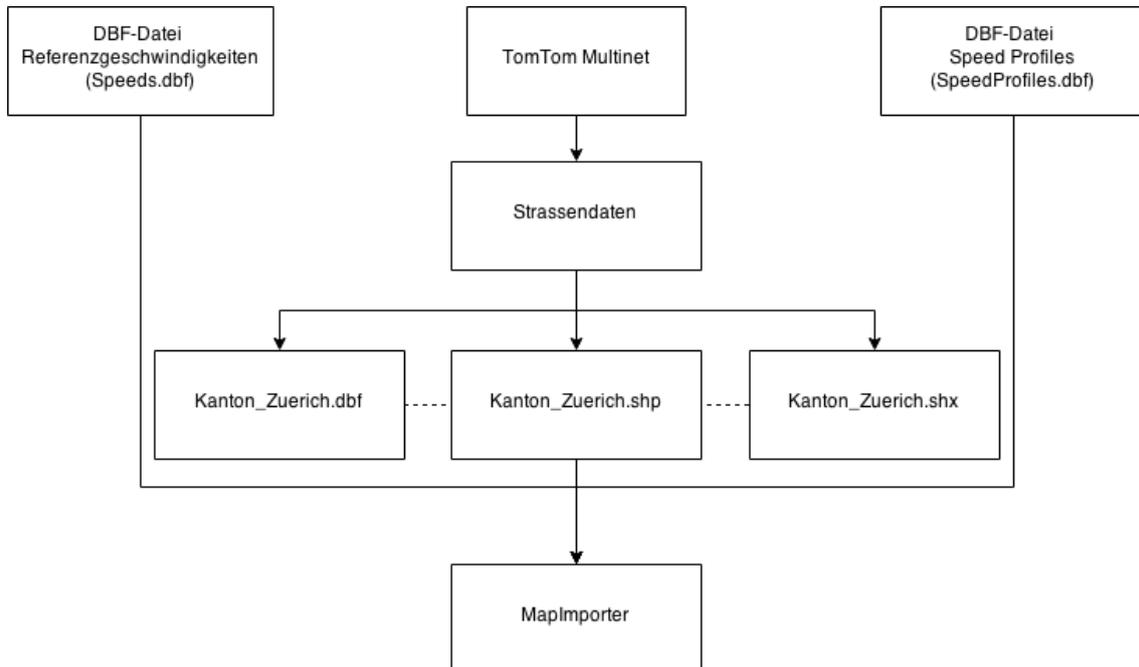


Abbildung 4.3.: TomTomMultinet und Geschwindigkeiten Import

Die Geschwindigkeiten für die einzelnen Strassenabschnitte werden aus den dBase-Dateien von TomTom ausgelesen. Der erste Ansatz war, die Daten mittels eines Type-4 Treibers für dBase-Dateien abzufragen. Da diese Treiber jedoch alle kostenpflichtig sind und die Testversionen zu grossen Einschränkungen unterliegen, wurde diese Idee verworfen. Die dBase-Dateien werden in der aktuellen Lösung Zeilenweise ausgelesen und die relevanten Daten, vergleiche Abschnitt 4.2, in HashMaps abgelegt. Die Strassen ID entspricht jeweils der NetworkID aus den dBase-Dateien, womit im Anschluss die Geschwindigkeiten pro Strassenabschnitt effizient ausgelesen werden können.

### 4.1.4. Simulationselemente

Zusätzlich zu den Nodes für die Geometrie und die Parkplätze, benötigt die Simulation zwei weitere Arten von Wegpunkten. Zum einen werden Edge-Nodes benötigt, auf welcher Fahrzeuge zur Simulation hinzugefügt werden können, oder zu denen sie navigieren um die Simulation zu verlassen. Zum anderen benötigen Fahrzeuge Destination-Nodes, in deren Nähe sie Parkplätze suchen. Da beim Einlesen der Strassendaten eine Bounding-Box verwendet wird, vergleiche Abschnitt 4.1.1, sind sämtliche Nodes die sich am Anfang oder Ende einer Strasse und ausserhalb der Bounding-Box befinden Edge-Nodes. Für die Destination-Nodes wird eine zweite Bounding-Box im Zentrum des Kartenausschnitts berechnet. Nodes, die weder Edge-Nodes noch Parkplätze sind, haben eine zufällige Wahrscheinlichkeit um eine Destination-Node zu werden. Diese Wahrscheinlichkeit ist in der inneren Bounding-Box drei mal höher, als in der äusseren Bounding-Box. Dies beruht auf der Annahme, dass der Kartenausschnitt eine Stadt oder Ortschaft im Zentrum hat und sich im Ortskern häufiger Ziele für Fahrzeuge befinden.

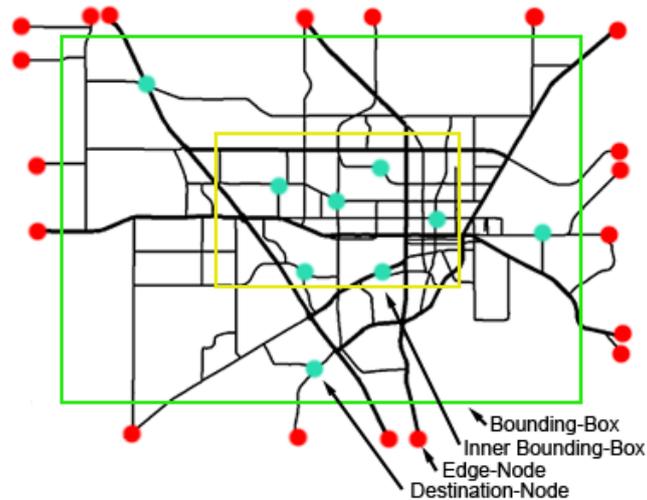


Abbildung 4.4.: Strassendaten mit Bounding-Boxes, Destination- und Edge-Nodes

## 4.2. TomTom-Daten

Die Daten von TomTom wurden einerseits im in Abschnitt 3.1 erwähnten Shape-Format ausgeliefert und andererseits in den beiden dBase-Dateien SpeedProfiles.dbf und Speeds.dbf. Diese können der Simulation mittels MapImporter zur Verfügung gestellt werden. Im folgenden Abschnitt wird der Aufbau der verwendeten Daten erklärt.

Bei den fett markierten Attributen der Abbildung 4.2 handelt es sich um die Primär- und Fremdschlüssel sowie wichtigen Attribute der dBase-Dateien.

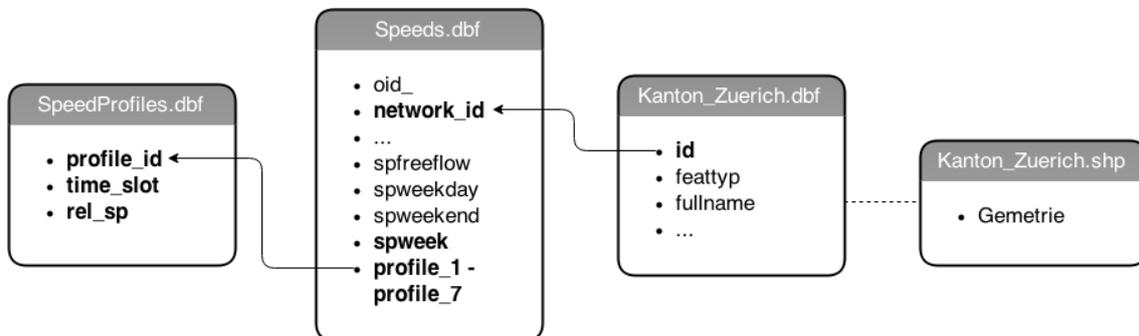


Abbildung 4.5.: Aufbau TomTom-Daten

### Kanton\_Zuerich.shp

Shape-Datei mit gesamter Geometrie in Form einer Gruppe von Vektor-Koordinaten. Die zusätzlichen Attribute der Shape-Datei sind in der dBase-Datei Kanton\_Zuerich.dbf hinterlegt.

### Kanton\_Zuerich.dbf

dBase-Datei mit Strassennamen und weiteren Attributen. Für die Referenz auf Speeds.dbf wird lediglich das Attribut **id** benötigt.

### Speeds.dbf

dBase-Datei mit der **network\_id**, den Durchschnittsgeschwindigkeiten (spfreeflow, spweekday, spweekend, **spweek**) und die Profile 1-7 (profile\_1 für Sonntag, profile\_7 für Samstag). Standardmässig wird beim Import zum momentanen Zeitpunkt lediglich der Wochendurchschnittswert der Geschwindigkeit

(**spweek**) im Verhältnis zur relativen Geschwindigkeit vom Montag (**profile\_2**) zum Zeitpunkt 07:00 Uhr verwendet. Obwohl dies vom MapImporter bereits angeboten wird, verfügt die Simulation zum jetzigen Zeitpunkt noch nicht über die Möglichkeit bei der Erstellung einer neuen Karte den Tag und die Uhrzeit anzugeben.

### SpeedProfiles.dbf

dBase-Datei mit allen verfügbaren Profilen (**profile\_id**) zu einer bestimmten Zeit (**time\_slot**) und der dazugehörigen relativen Geschwindigkeit in Prozent(**rel\_sp**). Die Zeiten sind in Sekunden angegeben und haben ein Intervall von 5 Minuten.

## 4.3. OSM-Daten

Die benötigten Parkplatzdaten wurden mangels besserer Quellen von OpenStreetMap durch Verwendung von Overpass Turbo bezogen. Zu diesem Zweck wurde für die definierten Bereiche der Stadt Zürich, Winterthur und des gesamten Kanton Zürich, ein Export der Daten mittels einer Abfrage erstellt. Die Daten sind als JSON-Datei verfügbar und können durch die Importer-Klasse sehr leicht verarbeitet werden. Nachfolgend sind Beispiele für eine Abfrage und die dadurch erhaltenen Daten.

```
1 <!-- Winterthur -->
2 <osm-script output="json" timeout="25">
3
4 <!-- gather results -->
5 <union>
6   <query type="way">
7     <has-kv k="amenity" v="parking"/>
8     <bbox-query s="47.497331" n="47.502782" w="8.722058" e="
9       8.733902"/>
10   </query>
11   <query type="node">
12     <has-kv k="amenity" v="parking"/>
13     <bbox-query s="47.497331" n="47.502782" w="8.722058" e="
14       8.733902"/>
15   </query>
16   <query type="way">
17     <has-kv k="amenity" v="parking_space"/>
18     <bbox-query s="47.497331" n="47.502782" w="8.722058" e="
19       8.733902"/>
20   </query>
21   <query type="node">
22     <has-kv k="amenity" v="parking_space"/>
23     <bbox-query s="47.497331" n="47.502782" w="8.722058" e="
24       8.733902"/>
25   </query>
26 </union>
27
28 <!-- print results -->
29 <print mode="body"/>
30 <recurse type="down"/>
31 <print mode="skeleton" order="quadtile"/>
32
33 </osm-script>
```

Listing 4.1: OSM Beispiel Abfrage

Um mit Hilfe der Beispiel Abfrage eine Liste aller Parkplätze zu finden, muss das Attribut **amenity** sowohl auf den Wert *parking* als auch auf *parking\_space* überprüft werden. Trotz den OSM Spezifikationen wurde häufig das falsche Attribut verwendet. Konkret wurden Parkhäuser, die nach Spezifikation als *parking* eingetragen sein sollten, als *parking\_space* definiert. Ein *parking\_space* ist eigentlich für Zusatzinformationen eines Parkhauses oder Parkplatzes (*parking*) definiert und sollte nicht für sich alleine stehen. Oft fehlt zudem bei Parkplätzen die Kapazität (**capacity**).

```
1 {
2   "version": 0.6,
3   "generator": "Overpass API",
4   "osm3s": {
5     "timestamp_osm_base": "2014-12-05T09:58:02Z",
6     "copyright": "The data included in this document is from
7       www.openstreetmap.org. The data is made available under
8       ODbL."
9   },
10  "elements": [
11    {
12      "type": "node",
13      "id": 240982910,
14      "lat": 47.4975657,
15      "lon": 8.7294672,
16      "tags": {
17        "amenity": "parking",
18        "name": "Technikum-Parkhaus",
19        "parking": "underground",
20        "capacity": "250",
21        "wheelchair": "yes"
22      }
23    },
24    {
25      "type": "node",
26      "id": 671814540,
27      "lat": 47.5054591,
28      "lon": 8.7145203,
29      "tags": {
30        "amenity": "parking",
31        "name": "Siska Gewerbezentrum"
32      }
33    },
34    {
35      "type": "way",
36      "id": 167052341,
37      "nodes": [
38        1784585144,
39        1784585143,
40        578424395,
41        1784585148
42      ],
43      "tags": {
44        "amenity": "parking_space",
45        "capacity": "6",
46        "parking": "surface"
47      }
48    }
49  ]
50 }
```

```
47   },
48   {
49     "type": "way",
50     "id": 167052392,
51     "nodes": [
52       1784585532,
53       1784585530,
54       1784585533,
55       1784585531
56     ],
57     "tags": {
58       "amenity": "parking",
59       "capacity": "5",
60       "parking": "surface"
61     }
62   },
63   ...
64 ]
65 }
```

Listing 4.2: OSM Beispiel JSON

## 4.4. API Tools

Als alternative zu Datenlieferanten wurden mehrere Programmierschnittstellen analysiert. Die folgenden beiden Programme entstanden während der Analyse der Programmierschnittstellen von Google und TomTom. Die Grundidee war, dass zuerst eine Route berechnet wird und als Folge der benötigten Zeit die Geschwindigkeiten auf den einzelnen Strassen berechnet werden kann. Die beiden Programme, sowie eine Auflistung der API-Aufrufe inklusive Antworten, befinden sich auf der CD im Anhang dieser Arbeit.

### 4.4.1. Google Distance Matrix

Das erste Programm verwendet die beiden Schnittstellen *Distance Matrix*<sup>[5]</sup> und *Geocode*<sup>[6]</sup> von Google. Die Geocode Schnittstelle dient dazu, Ortsnamen in Längen- und Breitengrade zu übersetzen und umgekehrt. In diesem Programm erleichtert es vor allem die manuelle Bedienung und Auswertung, da Strassen und Ortsnamen anstelle der Koordinaten verwendet werden können. Die Distance Matrix Schnittstelle berechnet alle möglichen Strecken von einer gegebenen Liste mit Startpunkten zu einer zweiten Liste mit Zielpunkten. Die Resultate werden bei beiden Schnittstellen in JSON Format zurückgegeben.

**Vorteile** Dadurch, dass ganze Listen von Start- und Zielpunkten angeben zu können, bietet Google Distance Matrix eine sehr effiziente Möglichkeit die Dauer von Streckenabschnitten zu berechnen. Geocoding sowie Reverse-Geocoding erleichtern die Bedienung enorm, sind jedoch für eine automatische Verarbeitung nicht zwingend notwendig.

**Nachteile** Der Nachteil der Google Schnittstellen ist, dass die Zeitangaben immer in ganzen Minuten sind. Werden Start und Zielpunkt auf die exakt selbe Koordinate gelegt beträgt die Fahrzeit immer 1 Minute. Die Schnittstelle ist somit für kurze Streckenabschnitte nicht geeignet.

### 4.4.2. TomTom Routes

Das zweite Programm verwendet die Schnittstelle *api.internal.tomtom.com*. Diese Schnittstelle wurde bei der Analyse des Netzwerkverkehrs der Routenplaner Webseite von TomTom entdeckt. Angebotene

Services dieser Schnittstelle umfassen Geocoding, Informationen über die aktuelle Verkehrslage und Navigationsinstruktionen für eine Route. Im Programm genutzt werden ausschliesslich die Navigationsinstruktionen.

**Vorteile** Der Vorteil dieser Schnittstelle ist, dass für sämtliche Teilstrecken sowohl die zurückgelegte Distanz als auch eine Zeitangabe in Sekunden zurückgegeben wird. Zusätzlich werden sogar die zu erwartenden Verspätungen pro Streckenabschnitt berechnet. Ausserdem ist es möglich die Strecke nicht nur zum aktuellen Zeitpunkt, sondern auch für die Zukunft oder Vergangenheit zu berechnen.

**Nachteile** Da es sich um eine interne Programmierschnittstelle handelt, ist keine öffentliche Dokumentation vorhanden. Zudem sind somit weder die URL noch das Verhalten der Schnittstelle garantiert, was sich speziell bei Fehlerfällen auf die Funktionalität der im Rahmen dieser Arbeit entwickelten Software auswirken kann.

## 5. Diskussion und Ausblick

Obwohl mit den durch diese Arbeit erhaltenen Daten schon eine sehr realitätsnahe Simulation ermöglicht wurde, können vor allem im Bereich der Parkplatzdaten noch Verbesserungen erzielt werden. Konkret durch andere qualitativ hochwertigere Datenbezugsquellen wie zum Beispiel Stadtverwaltungen.

Eine zusätzliche Möglichkeit zur Erweiterung wäre die Ausweitung auf weitere Städte und Länder. Hierfür müsste evaluiert werden, ob der bestehende Datenlieferant, namentlich TomTom, auch in weiteren Gebieten über qualitativ hochwertige Daten verfügt. Gegebenenfalls müsste ein neuer Vertrag ausgehandelt werden.

Neben der Erweiterung der Daten könnte zudem die Simulation um die Funktionalität erweitert werden, dass das volle Potential des Importers und der zur Verfügung gestellten Daten ausgeschöpft werden kann. Vor allem die Möglichkeit einer Simulation auf spezielle Wochentage und Uhrzeiten wäre durch den Importer schon gegeben und wäre durch minimale Anpassungen an der Simulation sehr einfach umzusetzen.

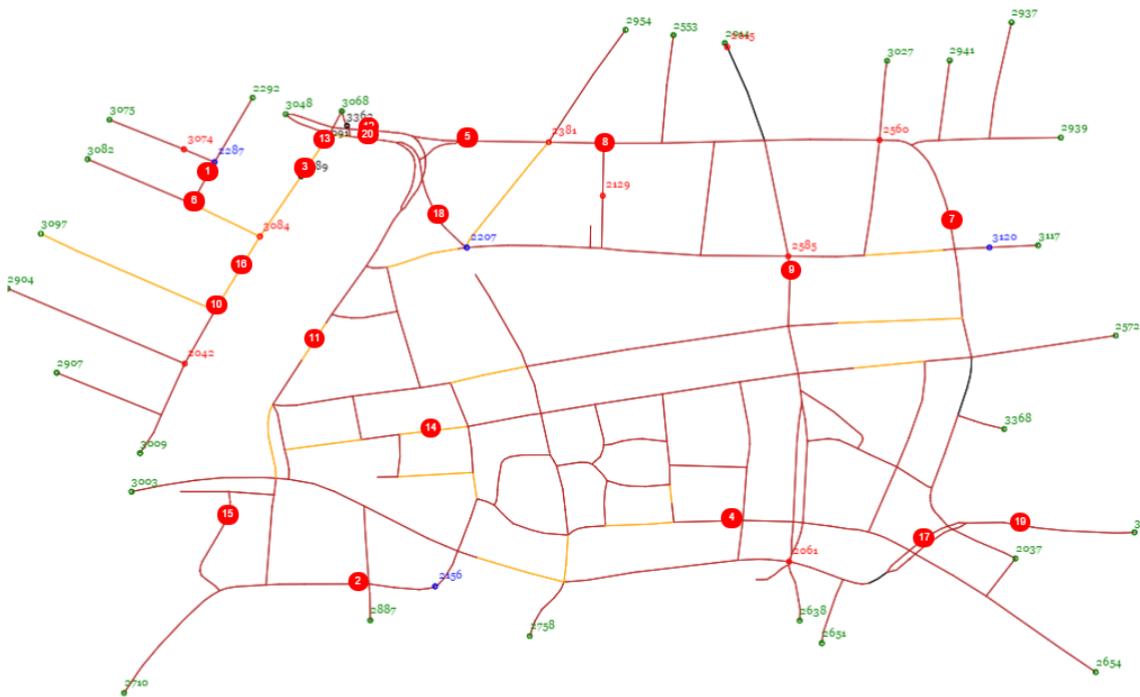


Abbildung 5.1.: Placebook Simulation - Endresultat

## **6. Verzeichnisse**

# Literaturverzeichnis

- [1] o.V. *ESRI Shapefile Technical Description*. Environmental Systems Research Institute, Inc., Juli 1998.
- [2] o.V. Belegung und Verkehrsaufkommen von Parkfeldern in der Stadt Zürich. Technical report, Planungsbüro Jud, September 2010.
- [3] o.V. Parkplatzbelegung in der Stadt Zürich. *Mobilitäts News*, Mai 2011.
- [4] o.V. Geoadmin API documentation, Dezember 2014. URL <http://api3.geo.admin.ch/>.
- [5] o.V. Google Distance Matrix, Oktober 2014. URL <https://developers.google.com/maps/documentation/distancematrix/?hl=de>.
- [6] o.V. Google Geocode, Oktober 2014. URL <https://developers.google.com/maps/documentation/geocoding/?hl=de>.
- [7] o.V. Microsoft Maps API, Dezember 2014. URL <http://msdn.microsoft.com/en-us/library/gg427611.aspx>.
- [8] o.V. Nokia Maps API, Dezember 2014. URL <https://developer.here.com/>.
- [9] o.V. OpenStreetMap - Overpass API, Oktober 2014. URL [http://wiki.openstreetmap.org/wiki/Overpass\\_API](http://wiki.openstreetmap.org/wiki/Overpass_API).
- [10] o.V. OpenStreetMap Extended API, Dezember 2014. URL <http://wiki.openstreetmap.org/wiki/DE:Xapi>.
- [11] o.V. Overpass API, Oktober 2014. URL <http://overpass-turbo.eu>.
- [12] o.V. TomTom Route Planner, Oktober 2014. URL <http://routes.tomtom.com>.
- [13] o.V. Yahoo Maps API, Oktober 2014. URL <https://developer.yahoo.com/maps/>.
- [14] o.V. Environmental Systems Research Institute, November 2014. URL <http://www.esri.com>.
- [15] o.V. Inrix inc., Oktober 2014. URL <http://www.inrix.com>.
- [16] o.V. Inrix inc., Oktober 2014. URL <http://www.inrix.com/resources/dev-zone/>.
- [17] o.V. Lost n' Found, Oktober 2014. URL <https://www.lostnfound.com>.
- [18] o.V. OpenStreetMap APIv0.6, Dezember 2014. URL [http://wiki.openstreetmap.org/wiki/API\\_v0.6](http://wiki.openstreetmap.org/wiki/API_v0.6).
- [19] o.V. OpenStreetMap, September 2014. URL <http://www.openstreetmap.org>.
- [20] o.V. OpenStreetMap Usage Policy, Dezember 2014. URL [http://wiki.openstreetmap.org/wiki/API\\_usage\\_policy](http://wiki.openstreetmap.org/wiki/API_usage_policy).
- [21] o.V. Play Framework, Oktober 2014. URL <https://www.playframework.com/documentation/2.3.x/Home>.
- [22] o.V. Viasuisse, Oktober 2014. URL <http://www.viasuisse.ch>.
- [23] Alex Werffeli und Burak Özçetin. Technologie - Konzept für eine revolutionäre Art zu parken, Juni 2014.

# Abbildungsverzeichnis

2.1. Placebook Simulation - Rechtwinklige Karte . . . . .	8
4.1. MapImporter Flussdiagramm . . . . .	13
4.2. Parkplatz Rohdaten und für die Simulation optimiert . . . . .	14
4.3. TomTomMultinet und Geschwindigkeiten Import . . . . .	15
4.4. Strassendaten mit Bounding-Boxes, Destination- und Edge-Nodes . . . . .	16
4.5. Aufbau TomTom-Daten . . . . .	16
5.1. Placebook Simulation - Endresultat . . . . .	21

# Listings

4.1. OSM Beispiel Abfrage . . . . .	17
4.2. OSM Beispiel JSON . . . . .	18

# **A. Anhang**

## **A.1. Offizielle Aufgabenstellung**



BetreuerInnen: Mark Cieliebak, ciel  
Fachgebiete: Mobilität und Verkehr (MVK)  
Software (SOW)  
Studiengang: IT / VS / WI  
Zuordnung: Institut für angewandte Informationstechnologie (InIT)  
Gruppengröße: 2

---

### **Kurzbeschreibung:**

#### **Kurzbeschreibung**

In dieser PA entwickeln Sie ein System, mit dem man echte Strassenkarten und Verkehrsdaten in einer Verkehrssimulation nutzen kann. Die Simulation existiert bereits für ein stark vereinfachtes Strassenmodell.

#### **Hintergrund**

Wir arbeiten zurzeit an einer Mobile-App, die ein effizientes Parkplatz-Matching ermöglicht: Mit "Placebook" können Autofahrer bekannt geben, wenn sie einen öffentlichen Parkplatz verlassen, und diesen an einen anderen Autofahrer übergeben. Durch ein optimales Matching von Parkplatz-Sucher und Anbieter wird dabei der Parkplatz-Suchverkehr verringert ein Gewinn für Autofahrer und Umwelt.

Es gibt für Placebook bereits ein Demo-System und eine Echtzeit-Simulation, die im Rahmen dieser Arbeit ausgebaut werden soll.

Konkret geht es darum, ein realitätsnahes Verkehrsnetz für die Simulation zu generieren. Im Moment beruht die Simulation auf einem stark vereinfachten Verkehrsnetz: alle Strassen verlaufen in einem Gitter, und für jeden Strassenabschnitt wird eine durchschnittliche Fahrgeschwindigkeit zufällig bestimmt.

#### **Aufgabe**

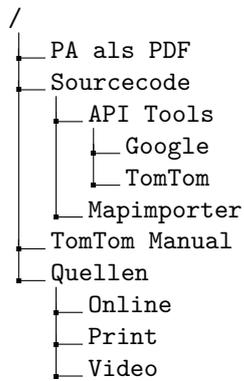
In dieser Arbeit sollen Methoden entwickelt werden, mit der echte Verkehrsdaten in der Simulation genutzt werden können. Dazu müssen geeignete Karten- und Verkehrsdaten gefunden, aufbereitet und in die Simulation integriert werden.

Es gibt verschiedenste Datenquellen, die dafür in Frage kommen. Strassenkarten sind einfach zu finden, z.B. bei Open Street Maps. Schwieriger wird es bei den Verkehrsdaten. Diese Daten sind soweit wir wissen - nicht "einfach so" verfügbar. Trotzdem sind diese Informationen implizit an verschiedenen Orten vorhanden: Jedes moderne Navigationssystem kennt die aktuelle Verkehrslage; Google Maps stellt Informationen zum Verkehrsfluss dar; oft sind die Positionen von öffentlichen Bussen verfügbar, etc. Hier ist Ihre Kreativität gefragt! Das Ziel ist, für jeden Strassenabschnitt eine sinnvolle Angabe zur durchschnittlichen Fahrzeit zu haben.

#### **Voraussetzungen:**

- Programmierkenntnisse
- Ideenreichtum und Kreativität

## A.2. Inhaltsverzeichnis CD



## A.3. Kontaktliste

Name	Organisation	Kontakt	Spezielles
Dienstabteilung Verkehr	Stadt Zürich	044 411 88 01	-
Frau Zollinger	Tiefbauamt Stadt Zürich	044 412 50 99	-
Parkplatzverordnung	Stadt Zürich	044 412 26 17	-
Erich Willi	Stadt Zürich	044 412 23 22	Parkplatzdaten (war nicht erreichbar)
Benno Knapp	Stadt Zürich	benno.knapp@zuerich.ch	Zugang zum Zählernetz-System
Frau Caracciolo	Stadt Zürich	044 411 88 70	ev. Daten über blauen Zonen
-	Lost n' Found	044 500 40 95 welcome@lostnfound.com	Position und Geschwindigkeit von Fahrzeugen, aber keine geordneten Daten
-	Viasuisse	058 329 99 99 info@viasuisse.ch	Daten von INRIX. Verkehrs und Stau-meldungen
Ase Elvebakk	TomTom Schweiz	079 128 47 26 ase.elvebakk@tomtom.com	Geschwindigkeitsprofile von Fahrzeugen
Francine Constantin	bbp geomatik ag	031 950 95 64 francine.constantin@geozen.ch	Schweizer Distributor von TomTom-Daten

## A.4. Verwendete Software

Software	Kommentar
GitHub	Versionsverwaltung
IntelliJ Ultimate	IDE
NetBeans	IDE
phpmyadmin	Tool zur Datenbankverwaltung
Placebook Simulation	Verkehrssimulation des InIT
Q-GIS	Shapefile Viewer für Windows
Spatial Commander	Shapefile Viewer für Linux
TeXstudio	LateX Editor
Wireshark	Analysetool für Netzwerkverkehr

## A.5. Netzwerkverkehrsanalyse [routes.tomtom.com](https://www.routes.tomtom.com)

# Analyse Traffic routes.tomtom.com

## Table of Contents

1Map and Area Data.....	1
2Traffic Status.....	3
3Points of Interest – Traffic Icons.....	4
4Geocoding.....	6
5Navigation Instructions.....	7

Die folgenden Requests werden beim Aufruf von folgender URL abgesetzt:

```
http://routes.tomtom.com/#/route/Romanshorn, CH@47.5657,9.37726@-1/Winterthur,  
CH@47.49988,8.72616@-1/?  
leave=now&traffic=true&center=47.5,8.75&zoom=9&map=basic
```

Die Auflistung beschränkt sich auf Requests mit JSON als Response. Zudem wurden verwandte Requests weggelassen (Bsp. Geocoding nur für einen Ort anstatt beide).

## 1 Map and Area Data

Host	api.internal.tomtom.com
Request	lbs/services/initialize/3/jsonp
Parameter	key=dvbfcb88hkrje9ur2fs84uxn language=en projection=EPSG900913 jsonp=json2
Response	json2({ "initializeResponse":{ "copyright":[ { "description":"Whereis&reg; map data is copyright &copy; 2014 Telstra Corporation Limited and its licensors", "label":"&copy; 2014 Whereis", "logo":"http://s3.amazonaws.com/warta-static-prod/images/copyright/whereis2.gif", "@id":"1" }, { "description":"GeoSmart&reg; map data is copyright &copy; 2014 GeoSmart Maps Limited and its licensors. GeoSmart&reg; is a registered trademark of GeoSmart Maps Limited of New Zealand.", "label":"&copy; 2014 GeoSmart", "logo":"http://s3.amazonaws.com/warta-static-prod/images/copyright/geosmart.gif", "@id":"2" }, ], }, }

```

    {
      "description":"Copyright &copy; 2006-2014 TomTom. All rights reserved. This material is proprietary and the subject of copyright protection, database right protection and other intellectual property rights owned by TomTom or its suppliers. The use of this material is subject to the terms of a license agreement. Any unauthorized copying or disclosure of this material will lead to criminal and civil liabilities.",
      "label":"&copy; 2014 TomTom",
      "logo":"http://s3.amazonaws.com/warta-static-prod/images/copyright/tomtom.png",
      "@id":"0"
    }
  ],
  "mapLocation":{
    "center":{
      "latitude":6024072.0,
      "longitude":974045.56
    },
    "@height":40000.0,
    "@width":40000.0
  },
  "trafficRegion":[
    {
      "areaName":"Netherlands",
      "point":{
        "latitude":6855769.5,
        "longitude":649635.44
      },
      "feedType":"HD",
      "fromZoom":0
    },
    {
      "areaName":"France",
      "point":{
        "latitude":5994782.5,
        "longitude":336549.38
      },
      "feedType":"HD",
      "fromZoom":0
    }
  ],
  "@version":"1.0.31-mascoma"
}
});

```

## 2 Traffic Status

Host	api.internal.tomtom.com
Request	lbs/services/viewportDesc/3/6002364.0025022,915838.7472664,6045780.2345682,1032252.3413447/11/6002364.0025022,915838.7472664,6045780.2345682,1032252.3413447/6/false/jsonp
Parameter	key=dvbfcb88hkrje9ur2fs84uxn language=de jsonp=json3
Response	json3({ "viewpResp":{ "trafficState":{ "@trafficAge":41226, "@trafficModelId":"1413887924080" }, "copyrightIds":"0", "@version":"1.0.31-mascoma", "@maps":"europe" } });

### 3 Points of Interest – Traffic Icons

Host	api.internal.tomtom.com
Request	lbs/services/trafficIcons/3/s3/5968203.1685067,900122.44508643,6046474.6854708,1037097.5997735/11/1413887924080/jsonp
Parameter	key=dvbfcb88hkrje9ur2fs84uxn language=de projection=EPSG900913 map=basic style=s2 jsonp=json5
Response	<pre>json5({   "tm":{     "@id":"1413887924080",     "poi":[       {         "id":"europe_CLUSTER_11_-1789148032",         "p":{           "x":907635.5,           "y":5977923.6         },         "ic":8,         "ty":0,         "cbl":{           "x":907619.0,           "y":5977678.1         },         "ctr":{           "x":907652.1,           "y":5978169.2         },         "cs":2,         "l":200       },       {         "id":"europe_CLUSTER_11_-1805884515",         "p":{           "x":924666.4,           "y":6019774.3         },         "ic":6,         "ty":2,         "cbl":{           "x":924344.9,           "y":6018818.2         },         "ctr":{           "x":924987.8,           "y":6020730.4         },       },     ]   } })</pre>

```
"cs":2,
"l":800
},
{
  "id":"europe_HD_CH_TTL131713829465665",
  "p":{
    "x":947951.9,
    "y":5970927.5
  },
  "ic":6,
  "ty":1,
  "cs":0,
  "d":"dichter Verkehr",
  "f":"Zug",
  "t":"Zug",
  "l":600,
  "dl":55,
  "r":"25"
},
{
  "id":"europe_HD_CH_TTR131676897153040",
  "p":{
    "x":955029.3,
    "y":5977371.7
  },
  "ic":8,
  "ty":4,
  "cs":0,
  "d":"gesperrt",
  "f":"Sihlbrugg (Hauptstrasse/338)",
  "t":"Hirzel (Hauptstrasse/338)",
  "l":3890
},
{
  "id":"europe_HD_CH_TTL131150597401792",
  "p":{
    "x":951427.6,
    "y":6010113.5
  },
  "ic":6,
  "ty":2,
  "cs":0,
  "d":"stockender Verkehr",
  "f":"Friesstrasse",
  "t":"Birchstrasse",
  "l":610,
  "dl":121
}
]
}
});
```

## 4 Geocoding

Host	api.internal.tomtom.com
Request	lbs/services/geocode/3/jsonp
Parameter	key=dvbfcb88hkrje9ur2fs84uxn language=de q=romanshorn jsonp=json6
Response	<pre>json6({   "geoResponse":{     "@duration":"0",     "@count":1,     "@version":"0.78.10.2",     "geoResult":{       "latitude":47.5657,       "longitude":9.37726,       "geohash":"u0qwt7h6610p",       "mapName":"europe",       "type":"city",       "city":"Romanshorn",       "country":"Schweiz",       "countryISO3":"CHE",       "formattedAddress":"Romanshorn, CH",       "widthMeters":10000.0,       "heightMeters":10000.0,       "score":1.0,       "defaultLanguage":"deu",       "mapVersion":"Europe_WBRP\t905\t4758\t0"     }   } })</pre>

## 5 Navigation Instructions

Host	api.internal.tomtom.com
Request	lbs/services/route/3/47.5657,9.37726:47.49988,8.72616/Quickest/jsonp
Parameter	key=dvbfcb88hkrje9ur2fs84uxn language=de projection=EPSG4326 avoidTraffic=true includeTraffic=true day=today time=now iqRoutes=2 trafficModelId=1413887924080 map=basic jsonp=json8
Response	<pre> json8({   "route":{     "summary":{       "routeKey":"q:1043871.8082361324,6034904.53228238:971391.687780632,602 4052.346553691:S:de:t:1:640:t:1413887924080:9:f:b:t",       "iqType":"S",       "bbox":{         "bottomLeft":{           "latitude":47.40802,           "longitude":8.71523         },         "topRight":{           "latitude":47.56569,           "longitude":9.44347         }       },       "trafficModelId":"1413887924080",       "startTimeZone":"Europe/Zurich",       "endTimeZone":"Europe/Zurich",       "totalDelaySeconds":0,       "totalDistanceMeters":79919,       "totalTimeSeconds":3552,       "departureOverview":{         "day":"today",         "time":"12:40",         "message":"Jetzt abfahren in {0}"       },       "arrivalOverview":{         "day":"today",         "time":"13:39",         "message":"Heute ankommen in {0}"       },       "routeURI":"@47.565699999999998,9.3772599999999998:@47.499879999999997 ,8.72616"     },   } } </pre>

```
"instructions":{
  "instruction":[
    {
      "distanceMeters":0,
      "iconPath":"start.png",
      "bbox":{
        "bottomLeft":{
          "latitude":47.5657,
          "longitude":9.37726
        },
        "topRight":{
          "latitude":47.5657,
          "longitude":9.37726
        }
      },
      "point":{
        "latitude":47.5657,
        "longitude":9.37726
      },
      "roadName":"Romanshorn",
      "roadNumber":"",
      "text":"Losfahren in",
      "travelTimeSeconds":0,
      "@isDepartureAction":true
    },
    {
      "distanceMeters":1148,
      "iconPath":"ra_cross.png",
      "bbox":{
        "bottomLeft":{
          "latitude":47.55877,
          "longitude":9.36932
        },
        "topRight":{
          "latitude":47.55877,
          "longitude":9.36932
        }
      },
      "point":{
        "latitude":47.55877,
        "longitude":9.36932
      },
      "roadName":"13 Arbonerstrasse",
      "roadNumber":"",
      "text":"Nehmen Sie am Kreisverkehr die dritte Ausfahrt auf",
      "travelTimeSeconds":168
    },
    {
      "distanceMeters":79919,
      "iconPath":"finish.png",
      "bbox":{
        "bottomLeft":{
```

```
        "latitude":47.49986,  
        "longitude":8.72622  
    },  
    "topRight":{  
        "latitude":47.49986,  
        "longitude":8.72622  
    }  
},  
"point":{  
    "latitude":47.49986,  
    "longitude":8.72622  
},  
"roadName":"Winterthur",  
"roadNumber":"","  
"text":"Ankunft in",  
"travelTimeSeconds":72,  
"@isDestination":true,  
"@isDestinationAction":true  
}  
]  
},  
"@version":"1.0.31-mascoma"  
}  
});
```