



Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

HOCHSCHULE FÜR TECHNIK UND WIRTSCHAFT

PROJEKTARRBEIT

Visualisierung und Auswertung von Positionsdaten der Omnibusse der BVG

Technik mobiler Systeme
Ausgewählte Kapitel mobiler Anwendungen

Pascal Dettmers (551733)
Stefan Neuberger (553849)
Tobias Ullerich (553746)

19. Dezember 2017

Inhaltsverzeichnis

1	Dokumentengeschichte	1
2	Problemstellung	2
3	Aufgabenstellung	3
4	Architektur	4
4.1	Überblick	4
4.1.1	Analyse Datenbank BVG	4
4.1.2	OSM	6
4.2	Komponenten	6
4.3	Server	6
4.3.1	Schnittstellendefinition	6
4.3.2	Datenbankschema	6
5	Nutzung	7
5.1	Code	7
5.1.1	Programmiersprache	7
5.1.2	Bibliotheken	7
5.2	Funktionsweise	7
5.3	Deployment / Runtime	7
6	Vorschläge / Ausblick	8
7	Literaturverzeichnis	9

Abbildungsverzeichnis

4.1 Datenbank Schema aus SC05 und SC51	5
--	---

Tabellenverzeichnis

1.1 Dokumentengeschichte	1
------------------------------------	---

Abkürzungsverzeichnis

RBL rechnerbasiertes Leitsystem

OSM Open Street Map

1 Dokumentengeschichte

Zeitraum	TPL/ Autor(en)	Änderungen
Wintersemester 2017/18 (09.11.2017)	Tobias Ullerich	Initiale Dokumentenstruktur Entwurf Aufgabenstellung
Wintersemester 2017/18 (01.12.2017)	Tobias Ullerich	Analyse BVG Datenbank 1/2
Wintersemester 2017/18 (05.12.2017)	Tobias Ullerich	Analyse BVG Datenbank 2/2
Wintersemester 2017/18 (19.12.2017)	Tobias Ullerich	Update Dokumentenstruktur Update Aufgabenstellung

Tabelle 1.1: Dokumentengeschichte

2 Problemstellung

Problemstellung Busbunching, BVG, Ist-Zustand, Ziel

3 Aufgabenstellung

Das zu entwickelnde System ist eine Komponente, die einem Busfahrer Informationen über die Busse einer Buslinie zur Verfügung stellen soll. Dabei soll das System nicht nur auf Busse beschränkt sein. Denkbar sind hier auch andere Arten von öffentlichen Transportmitteln. Dabei wird die Komponente den zeitlichen Abstand zum vorherigen und nachfolgenden Bus einer Linie visualisieren. Die Visualisierung geschieht durch eine Android Applikation.

4 Archtitektur

4.1 Überblick

4.1.1 Analyse Datenbank BVG

Die BVG nutzt für die Persistierung der Daten, inklusive der Prozessdaten, ein Datenbanksystem der Firma Oracle. Es werden bei der BVG zwischen zwei verschiedenen Systemen unterschieden. Zum Einen gibt es die sogenannte SC05 Schnittstelle. Diese enthält Prozessdaten der aktuellen Betriebslage. Dazu zählen unter anderen Positionen von Bussen und deren Verspätung. (vgl.: „Die Prozessdatenschnittstelle (SC05) spiegelt die aktuelle Situation im RBL wider.“) Zum Anderen gibt es die SC51 Datenbank, entwickelt von der Firma Alcatel. Diese Schnittstelle enthält unterschiedlichste Daten für Durchführung des öffentlichen Nahverkehrs der BVG. Darunter fallen Informationen zu Linien (Bus und Bahn), Informationen über deren Routen mittels geografischer Koordinaten und vieles mehr. Für die Analyse dieser relationaler Datenbanken waren jeweils deren Dokumentationen und ein Dump zur Verfügung.

Der erste Schritt der Analyse bestand darin, die Dumps der Oracle Datenbank zu importieren, um anschließend Zugriff auf die Tabellen und deren Daten zu erlangen. Für den Import fiel die Entscheidung für das Tool „OraDump to MySQL“. Mit diesem Tool ist es möglich ein Oracle Datenbank Dump in eine MySQL Datenbank zu importieren. Vorteil dieser Methode ist, dass auf bestehende Kenntnisse mit dem Umgang von MySQL zurückgegriffen werden kann. Im folgenden wurde mittels der Schnittstellen Dokumentation die Struktur der Datenbank analysiert. Im Fokus dieser Analyse stehen die Routen Information aus der SC51 und die Positionsdaten der Fahrzeuge aus der SC05 Schnittstelle. Bei der Analyse haben sich folgende Datenbanktabellen als Wertvoll gezeigt.

Die Tabelle `CM_VEHICLE_POSITION` aus der SC05 Datenbank enthält Informationen zu der aktuellen geografischen Position mittels Latitude und Longitude, der Abweichung vom Sollfahrplan, sowie eine Einordnung in die Route. Um ein Omnibus auf einer Route einzuordnen, gibt es eine endliche Menge von geografischen Punkten. Zu all diesen Punkten ist ein zeitlicher und örtlicher Abstand bekannt (siehe SC51). Zu jedem Fahrzeug ist der letzte passierte Punkt der Route referenziert (`LAST_POI_ORDER`). Der prozentuale Abstand zum Folgepunkt auf einer Route ist ebenfalls in der Relation durch die Spalte `REL_LNK_DISTANCE` gegeben.

Für die Zuordnung der Fahrzeuge aus der Tabelle `CM_VEHICLE_POSITION` zu einer Route und einem Kurs gibt es in der SC05 zwei Tabellen. Zum Einen hat die Tabelle `CM_ACCT_COURSES` die Aufgabe ein Fahrzeug einem Kurs zuzuordnen. Zum Anderen wird durch `CM_ACCT_JOURNEY` ein Bus einer Route zugeordnet. Somit können die

PointsOnRoute einem Fahrzeug zugeordnet werden.

Die Datenbank SC51 beinhaltet Tabellen für die Linien (Lines). Eine Linie ist im Kontext der BVG zum Beispiel die Buslinie X11. Jede Linie besteht aus mehreren Fahrten, hier COURSES_ON_JOURNEY genannt. Zu einem Kurs gehören Informationen wie Startzeit, Endzeit und eine Kursnummer, die nur im Kontext einer Linie eindeutig ist.

Die geografischen Informationen zum Routenverlauf werden in den Tabellen ROUTE, POINTS_ON_ROUTE und NETWORK_POINTS verwaltet. Zu einer Buslinie können verschiedene Routen gehören. Diese Routen sind in der Tabelle ROUTE zu finden. Dafür enthält auch diese Relation zusätzlich ein Feld Description (Beispieldaten: Falkensee, Bahnhof->S+U Rathaus Spandau). Die Tabelle POINTS_ON_ROUTE kordnet die Punkte einer Route, indem jeder Punkt einen Laufnummer hat (POR_ORDER). Um den zeitlichen und örtlichen Abstand zwischen zwei Punkten zu ermitteln, wird die Relation LINKS verwendet. Die Tabelle NETWORK_POINTS enthält abschließend die eigentliche geografischen Punkte in der Form Latitude und Longitude.

Im Abbildung 4.1 sind die Zusammenhänge der einzelnen Datenbanktabelle von SC05 und SC51 zu sehen. Dabei handelt es sich lediglich um ein Auszug der relevanten Daten für das zu entwickelnde System.

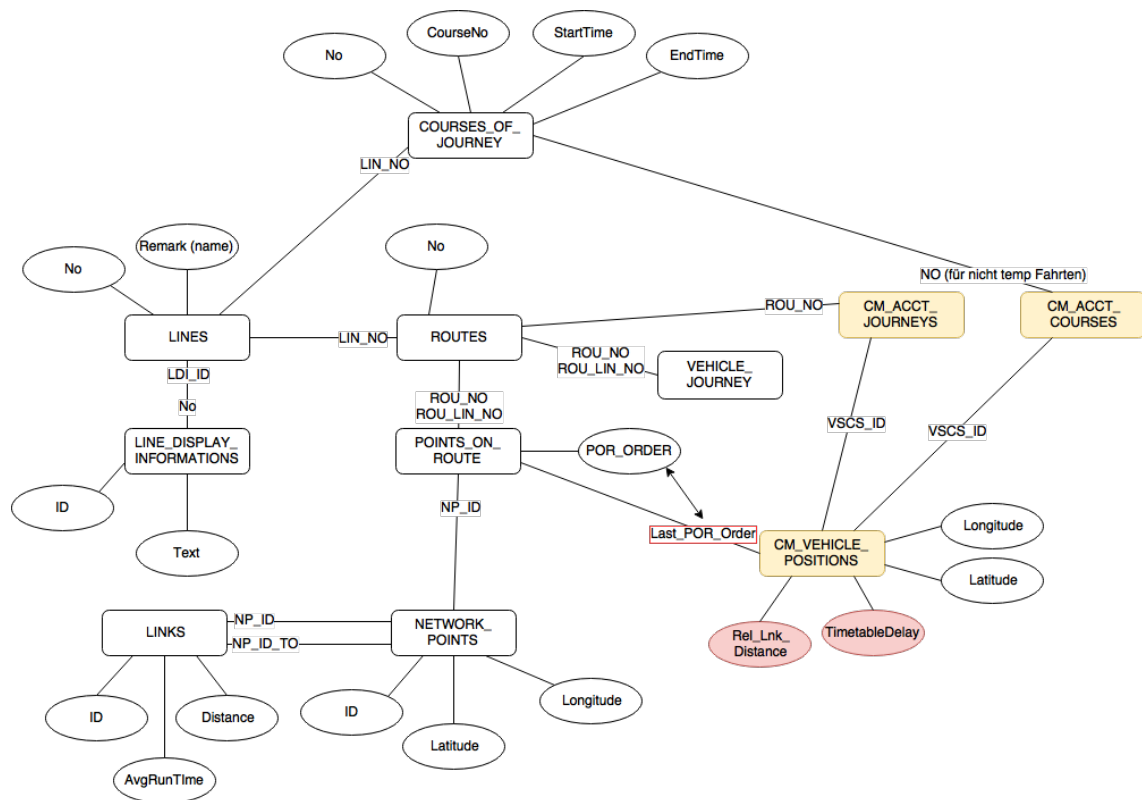


Abbildung 4.1: Datenbank Schema aus SC05 und SC51

4.1.2 OSM

Wie kommt man an die Routen Infos zu den Bus-/Tram Linien. GeoJson.

4.2 Komponenten

Bestandteile des Systems beschreiben (Server, App, ...)

4.3 Server

4.3.1 Schnittstellendefinition

Thema: HTTP Server. Requests, Responses.

4.3.2 Datenbankschema

5 Nutzung

5.1 Code

5.1.1 Programmiersprache

5.1.2 Bibliotheken

5.2 Funktionsweise

Wie funktioniert die Berechnung.

5.3 Deployment / Runtime

6 Vorschläge / Ausblick

7 Literaturverzeichnis