



HAMK UNIVERSITY OF APPLIED SCIENCES  
AND RAMBOLL GROUP

PROJECT REPORT

# Transportation modeling and Web content

*František Kolovský*  
*Pierre-Vincent Vrot*

Erasmus Exchange from  
University of West Bohemia  
and IMERIR



25th November 2015  
version 0.1



# Abstract

The Geographic Information Systems (GIS) allow, among others, to acquire, to treat, to organize and to present geographical data, to produce clear, accurate and intuitive maps, via an accessible web component from any browser.

The TraMap project, proposed by HAMK and supervised by Ramboll, is a simple and flexible tool that allows a user to view the transportation habits of information from open source geographic data and show informations in an application (web or mobile).

It in this context, of consultation tool and operating data, that our project takes place. Our role is to look for open data sources available on the Internet, to extract the information, to build an usual transport model and then develop a consultation multiplatform web application.

**Keywords:** GIS, Transport Modeling, Web, Algorithm, Open Sources

# Acknowledgments

acknowledgments.tex

# Contents

<b>Abstract</b>	<b>2</b>
<b>Acknowledgments.tex</b>	<b>3</b>
<b>Introduction</b>	<b>5</b>
<b>1 Transportation Modeling</b>	<b>6</b>
1.1 Transportation model . . . . .	6
1.1.1 Trip destination . . . . .	7
1.1.2 Count traffic . . . . .	7
1.2 Implementation . . . . .	8
1.2.1 Static shortest path search . . . . .	8
1.2.2 Data store . . . . .	9
<b>2 Web content</b>	<b>10</b>
2.1 Prototypes . . . . .	10
2.1.1 Map content . . . . .	10
2.1.2 Website content . . . . .	13
2.2 Development . . . . .	14
2.3 Practice . . . . .	15
<b>3 Merge</b>	<b>16</b>
<b>Conclusion</b>	<b>17</b>
<b>Glossary</b>	<b>20</b>

# Introduction

In Finland, cycling, walking and public transport are increasingly used, it becomes necessary to bring the most useful information possible means in order to anticipate the movements.

In the current era, companies are always on the lookout for new products, particularly in the GIS domain, either issue new tools or technology. Each companies develop and market their traffic information tools. That is why our project is centered on the sharing of information and free access to sources and data.

Today, open source and open data sources of information are increasingly used and allow to evolve in cooperative community. Our main objective for this project is to recover the data and open source tools on which anyone could build or develop a data model of consultation on transportation habits of a city. Our research work will focus on the city of Hyvinkää and bicycle habits data.

The work is divided into three parts: the first is to determine the theoretical and transportation model data to be consumed; the second explains how an application can be developed for the retrieval of data; and the third and final section combines the previous parts in order to get a result viewed on different platforms.

# 1 Transportation Modeling

## 1.1 Transportation model

Traditional transportation modelling have a several independent step (e.g trip generation, trip destination, ...). Our implementation of transportation modelling consist from 3 steps.

**trip generation** - In this part we want to determine number of trip origin, destination in zones. This step is depend on data (e.g demographic data, socio-economic data, weather, local habits, ...). Quality of model depend a lot on these data.

**trip destination** - In this section we want determine *transportation matrix*  $T$  (OD matrix). It say how many people travel from zone  $i$  to zone  $j$ . For it was used Gravity model.

**counting traffic** - This is last step. We compute traffic for every road link (edge of graph).

In upcoming section following expressions will be used:

$T$	transportation matrix (number of trips between zones)
$C$	travel cost matrix (between zones)
$T_i, T_j$	sum value in row/column in $T$
$n$	number of zones (size of matrix)

### 1.1.1 Trip destination

For determine transportation matrix  $T$  was used Gravity model:

$$T_{ij} = K_i K_j T_i T_j f(C_{ij})$$

where

$$T_i = \sum_{j=1}^n T_{ij}$$

$$T_j = \sum_{i=1}^n T_{ij}$$

$$K_i = \frac{1}{\sum_j K_j T_j f(C_{ij})}$$

$$K_j = \frac{1}{\sum_i K_i T_i f(C_{ij})}$$

where in our case

$$f(x) = x^{-2}$$

.  $T_i$  is number of trips outcoming from the zone (origin in the zone)  $i$ ,  $T_j$  is number of trips incoming to the zone (destination in zone)  $j$ . So sometimes transportation matrix is called *OD Matrix*.

Now we must determine  $K_i$  and  $K_j$ . We used iterative proportional fitting. It is iterative solution. First we compute  $T^1$  with  $K_i, K_j = 1$ . After we can use iteration equation for  $T$ :

$$T_{ij}^m = \frac{Z_i}{T_i^{m-1}} T_{ij}^{m-1}$$

$$T_{ij}^m = \frac{Z_j}{T_j^{m-1}} T_{ij}^{m-1}$$

where  $Z_i$  and  $Z_j$  are origin and destination trips (we know),  $m$  is iteration.

### 1.1.2 Count traffic

Now we know how many people travel from zone  $i$  to zone  $j$ , so we can find path from  $i$  to  $j$  and attributed this value into every edges in path.



In our solution we compute  $N$  paths for every pair of zones. Every path is based on another cost. Cost is based on length, time and vertical distance. Final cost is linear combination these partition cost.

$$c = \begin{pmatrix} k_t & k_l & k_h \end{pmatrix} \begin{pmatrix} t \\ l \\ h \end{pmatrix}$$

where  $c$  is cost,  $t$  is time,  $l$  is length and  $h$  is vertical distance. Number of trip is split evenly among all the paths.

## 1.2 Implementation

In upcoming section following expressions will be used:

- $n$  number of nodes
- $m$  number of edges
- $z$  number of zones
- $p$  number of path computed for one pair of zone

Transportation modelling described in previous section was implemented in Python programming language.

### 1.2.1 Static shortest path search

For determined  $C$  we use Dijkstra's algorithm (complexity  $O(m + n \log(n))$ ). So final complexity is

$$z(m + n \log(n))$$

For traffic count we used also Dijkstra's algorithm. Final complexity for traffic count is

$$pz(m + n \log(n))$$

For Dijkstra's algorithm we used Python library iGraph. iGraph is written in C programming language so it is fast. For example one Dijkstra running 30 ms ( $n = 8000$   $m = 18000$ ).

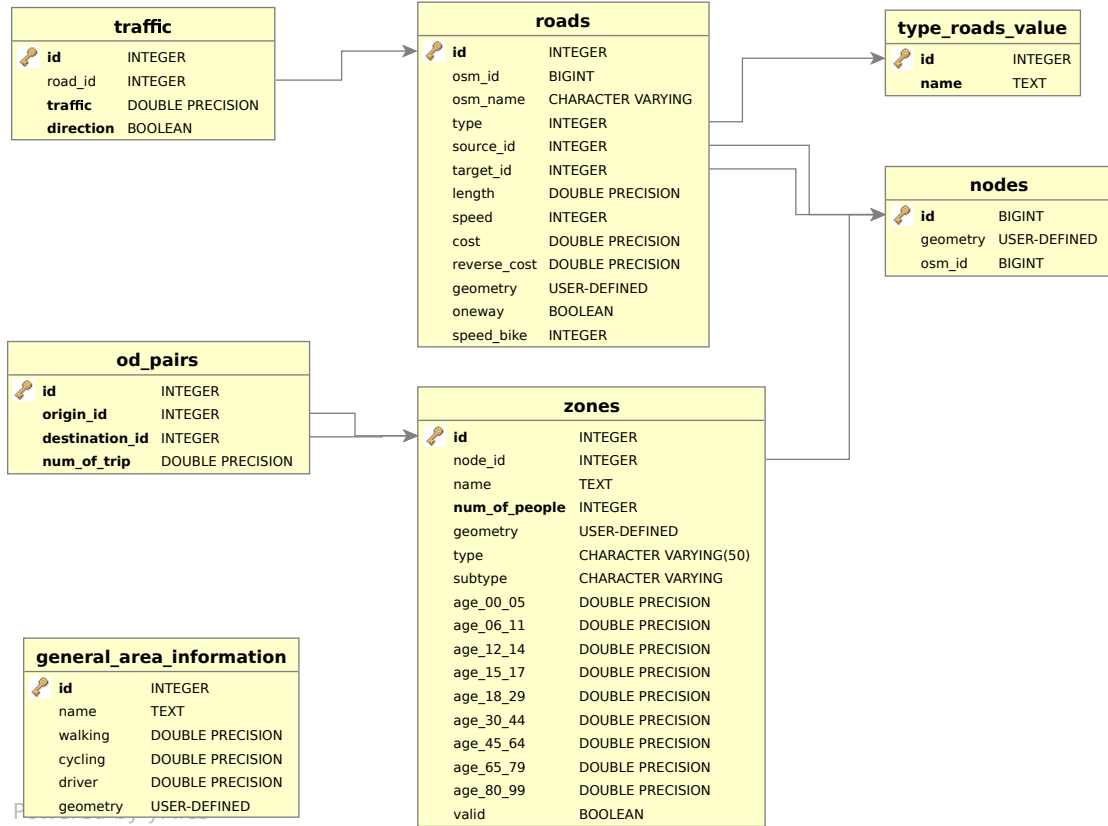


Figure 1.1: Database model

### 1.2.2 Data store

All data for transportation modelling are stored in relation database PostgreSQL with extension PostGIS. In database there are 4 main table:

roads	road links (edges)
nodes	nodes (vertexes)
zones	list of zones
traffic	
general_area_information	contains interested area geometry
od_pair	Database implementation for $T$ matrix

More details about DB you can find in project documentation on GitHub.

## 2 Web content

Before starting the development of the application, it is necessary to make a statement of needs and expectations. Thus, by structuring our interface will make it easier to conduct its graphic development, anticipate the classes and functions necessary for treatment. Initially, we will structure our interfaces in order to be closer to the users expectations, then we will detail the tools available and how they work, and then we will proceed to the developement of the client application.

### 2.1 Prototypes

In this section we will detail each of the interfaces that will be implemented in the application. The application will be built in two different ways:

**Map content :**

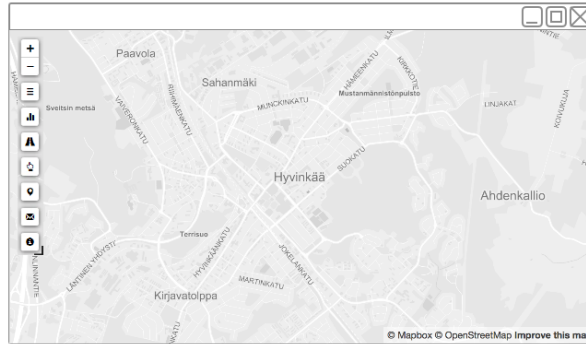
containing a basemap and whose tools are over the map

**Website content :**

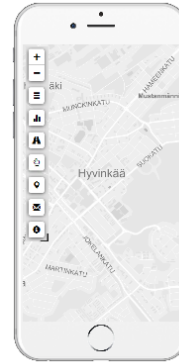
containing independent map tools

#### 2.1.1 Map content

The map content determines the basis of our application. Indeed, the latter consists of three major components: the base map, table of contents and the interaction with map buttons.



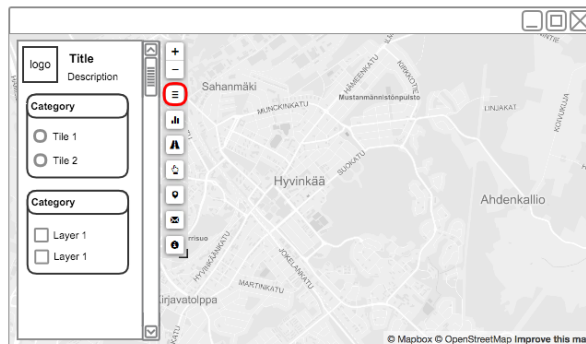
(a) Web



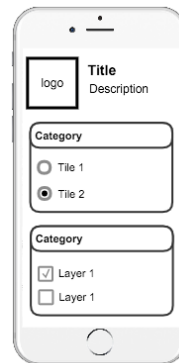
(b) Mobile

Figure 2.1: Application: Basemap

The base map allows you to locate geographically in a recognizable environment. It shall be set up raster image on which we will be able to overlay vector data from the server. For our application we will build on the maps tiles proposed by MapBox.



(a) Web



(b) Mobile

Figure 2.2: Application: Table Of Content

The table of contents (abbreviated TOC) can list the elements of the map and interact with them. It will contain the different layers component of the application. The checkbox or radio button should be there to show or hide items.

The interactions buttons are of three types: the onset of action over the map (popup), the onset of action with interaction on the map (marker), or

open a specific page of the mini-website.

Among the action buttons on the map, an indispensable tool is the route search. This button turns the marker removal by hand directly on the map to place two points that serve as departure and arrival. After dropping the markers, the shortest route is calculated and displayed on the map.



Figure 2.3: Application: SearchPointer

Another of the tools necessary for the application is a button to geotag. This button opens a popup over the map to choose to locate by keyword or by coordinates (latitude and longitude).



Figure 2.4: Application: Focus

The last two buttons that need to be added are not necessary for the application but merely useful and informative. These are contact buttons

and information about the application. These two buttons open a popup over the map. The contact button provides a form to send a message directly to the developers of the application. As for the information button is a simple overlay that displays information about the developers and the sources of the application.



Figure 2.5: Application: Contact

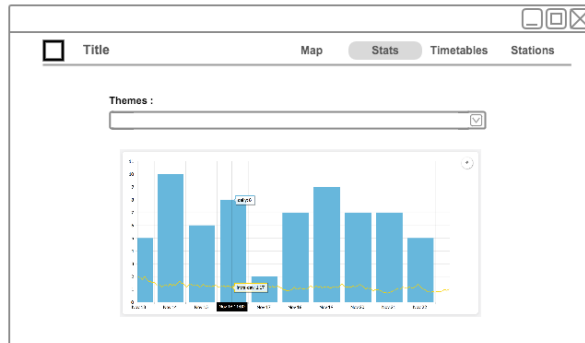
### 2.1.2 Website content

The last buttons that were not detailed in the previous section are direct links to the Internet website content. Indeed, it is easier to go directly to the desired content from the map rather than browse in the site. Among the non-explicit buttons, we have a button access to statistics and trains timetables.

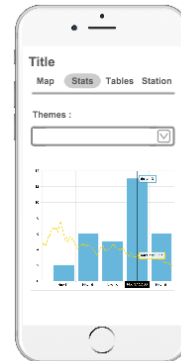
The website consists of a small top navigation bar and a central content. A link in the navigation bar allows us to go back on the map. The central content of the page changes depending on the desired chart.

The central content of the statistical part is a charts list and an overall chart that is updated as the selected object on the list.

Regarding train schedules, it is an interface to retrieve the trains that interest the user. The user then has the choice of listing all train departures of a city, all arrivals to a city or to determine the route between two cities. The choice made, the user fills in the desired towns in the affected areas. The application then retrieves the data and signs the user in a list to open.

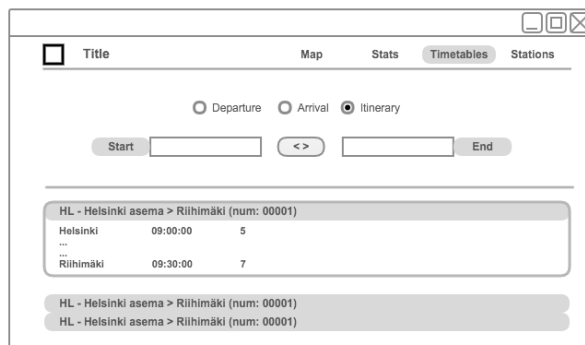


(a) Web

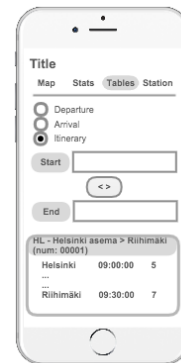


(b) Mobile

Figure 2.6: Application: Statistics



(a) Web



(b) Mobile

Figure 2.7: Application: Timetables

## 2.2 Development

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed tempus faucibus nisi maximus commodo. Suspendisse potenti. Duis viverra scelerisque sem, ac convallis nibh tristique commodo. Donec quis leo id ligula eleifend efficitur sed vel est. Mauris ac pharetra erat. Fusce sit amet ligula elementum, aliquet sem in, aliquam tortor. Sed dapibus leo vel facilisis commodo. Sed porttitor nisl sem, eu ultricies dui fermentum a. Cras sed dictum leo, ut posuere turpis. Pellentesque interdum nisl odio, ut accumsan mauris fermentum a. Aliquam accumsan sagittis magna, a sodales magna sagittis quis. Nulla rutrum nisl libero, sit amet egestas arcu consectetur vitae. Aliquam fringilla rutrum

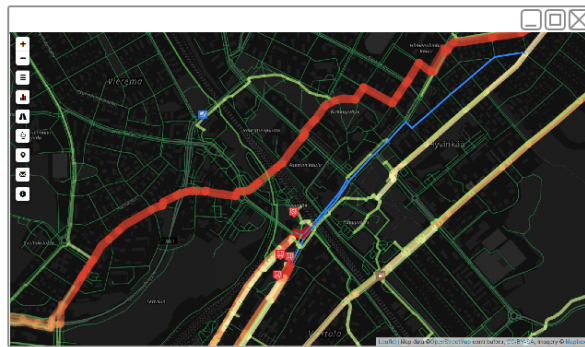
auctor. Nullam tempor placerat massa, quis gravida nisi dignissim vitae.

## 2.3 Practice

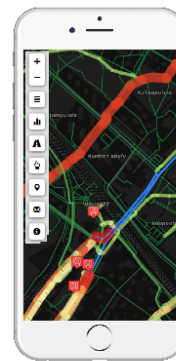


### 3 Merge

s01-data.tex



(a) Web



(b) Mobile

Figure 3.1: Application: Focus

# Conclusion

conclusion.tex

# Bibliography

- [REF] <http://leafletjs.com/>  
*Leaflet website*. 2015 Vladimir Agafonkin. Maps - OpenStreetMap contributors.
- [REF] <https://github.com/Turbo87/leaflet-sidebar>  
*Leaflet extension Sidebar*. leaflet-sidebar is free software, and may be redistributed under the MIT license.
- [REF] <https://github.com/CliffCloud/Leaflet.EasyButton>  
*Leaflet extension EasyButton*. Leaflet-EasyButton maintained by CliffCloud
- [REF] <http://getbootstrap.com/>  
*Bootstrap CSS Framework website*. Code licensed under MIT, documentation under CC BY 3.0.
- [REF] <http://silviomoreto.github.io/bootstrap-select/>  
*Bootstrap extension Select*. Bootstrap-select maintained by caseyjhol
- [REF] <https://cordova.apache.org/>  
*Cordova website*. Copyright 2012, 2013, 2015 The Apache Software Foundation, Licensed under the Apache License, Version 2.0
- [REF] <https://github.com/jsdoc3/jsdoc>  
*JsDoc3 : Generate Javascript documentation*. JSDoc 3 is free software, licensed under the Apache License, Version 2.0.

# List of Figures

1.1	Database model . . . . .	9
2.1	Application: Basemap . . . . .	11
2.2	Application: Table Of Content . . . . .	11
2.3	Application: SearchPointer . . . . .	12
2.4	Application: Focus . . . . .	12
2.5	Application: Contact . . . . .	13
2.6	Application: Statistics . . . . .	14
2.7	Application: Timetables . . . . .	14
3.1	Application: Focus . . . . .	16

## List of Tables