BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF INFORMATION SYSTEMS
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

DYNAMIC VISUALISATION TOOL FOR BRNO PUBLIC TRANSPORT DATA

NÁSTROJ PRO DYNAMICKOU VIZUALIZACI DAT HROMADNÉ DOPRAVY MĚSTA BRNA

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR — PATRIK GÁFRIK
AUTOR PRÁCE

SUPERVISOR — Ing. JURAJ LAZÚR
VEDOUCÍ PRÁCE

BRNO 2024

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Bachelor's Thesis Assignment

154338

Institut: Department of Information Systems (DIFS)
Student: **Gáfrik Patrik**
Programme: Information Technology
Title: **Dynamic Visualisation Tool for Brno Public Transport Data**
Category: Information Systems
Academic year: 2023/24

Assignment:

1. Learn about storing, sharing and visualising geographic data.
2. Study available data sources describing the public transport system of Brno and its behaviour.
3. Explore the issues of real-time visualisation of large volumes of data.
4. Design an interactive tool to enable dynamic visualisations over the Brno public transport system using historical data.
5. After consultation with the supervisor implement the proposed solution.
6. Test the functionality and usability of the system in collaboration with a selected sample of users.

Literature:

- Fotheringham, Stewart, and Peter Rogerson, eds. *Spatial analysis and GIS*. Crc Press, 2013.
- HYNEK, Jirí; KACHLÍK, Jakub; RUSNÁK, Vít. Geovisto: A Toolkit for Generic Geospatial Data Visualization. In: *VISIGRAPP (3: IVAPP)*. 2021. p. 101-111.
- Torre-Bastida, A. I., Del Ser, J., Laña, I., Ilardia, M., Bilbao, M. N., & Campos-Cordobés, S. (2018). Big Data for transportation and mobility: recent advances, trends and challenges. IET Intelligent Transport Systems, 12(8), 742-755.
- Leaflet: Leaflet API reference [online]. 2023 [cit. 2023-09-30]. Available at: https://leafletjs.com/reference-1.9.4.html

Requirements for the semestral defence:
Items 1. to 4.

Detailed formal requirements can be found at https://www.fit.vut.cz/study/theses/

Supervisor: **Lazúr Juraj, Ing.**
Head of Department: Kolář Dušan, doc. Dr. Ing.
Beginning of work: 1.11.2023
Submission deadline: 9.5.2024
Approval date: 30.10.2023

Abstract

Public transport plays a key role in urban mobility, especially in cities like Brno, where it accounts for almost half of all transport modes. In order to optimise the efficiency of public transport, cities face the challenge of developing effective strategies based on data analysis. However, existing tools for visualizing public transport data in Brno are limited and often only provide visualizations of static data and are not designed to display dynamic data changes based on user inputs. This thesis aims to address this problem by developing a tool that processes and visually represents public transport data in Brno to facilitate analysis. The tool is able to visualize dynamic changes in traffic data over time and allows the user to analyze the density of connections and transfers, the duration of trips or the location and delay of public transport vehicles in real-time.

Abstrakt

Verejná doprava zohráva kľúčovú úlohu v mestskej mobilite, najmä v mestách ako Brno, kde tvorí takmer polovicu všetkých druhov dopravy. V záujme optimalizácie efektivity verejnej dopravy čelia mestá výzve vypracovať účinné stratégie založené na analýze dát. Existujúce nástroje na vizualizáciu dát o verejnej doprave v Brne sú však obmedzené a často poskytujú len vizualizáciu statických dát a nie sú navrhnuté tak, aby zobrazovali dynamické zmeny dát na základe vstupov užívateľa. Cieľom tejto práce je vyriešiť tento problém vývojom nástroja, ktorý spracúva a vizuálne reprezentuje dáta o verejnej doprave v Brne a uľahčuje tak analýzu. Nástroj je schopný vizualizovať dynamické zmeny dopravných dát v čase a umožňuje užívateľovi analyzovať hustotu spojov a prestupov, dĺžku trvania ciest alebo polohu a meškanie vozidiel hromadnej dopravy v reálnom čase.

Keywords

public transport, data visualization, geospatial data, big data, real-time visualization, Brno

Klíčová slova

hromadná doprava, vizualizácia dát, geografické dáta, vizualizácia v reálnom čase, Brno

Reference

GÁFRIK, Patrik. *Dynamic Visualisation Tool for Brno Public Transport Data.* Brno, 2024. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Juraj Lazúr

Rozšířený abstrakt

Verejná doprava je jedným z najpoužívanejších spôsobov dopravy v mestách na celom svete. V Brne predstavuje takmer 50% všetkých spôsobov dopravy, ktoré obyvatelia využívajú, čo z nej robí najrozšírenejší spôsob dopravy v meste, za ktorým nasleduje automobilová doprava, ktorá predstavuje 32 %. V záujme našej spoločnosti je podporovať využívanie verejnej dopravy ako alternatívy k individuálnej automobilovej doprave s cieľom zlepšiť kvalitu ovzdušia, keďže využívanie verejnej dopravy výrazne znižuje emisie oxidu uhličitého v porovnaní s automobilovou dopravou. Okrem toho podpora využívania verejnej dopravy prináša množstvo ďalších výhod, ako napríklad menšie dopravné zápchy, ktoré vedú k menšiemu počtu dopravných nehôd.

Aby verejná doprava fungovala čo najefektívnejšie, mestá musia vypracovať rôzne stratégie. Rozhodnutia prijaté v rámci týchto stratégií sú založené na analýze zozbieraných dát. Tieto dáta môžu predstavovať rôzne faktory, napríklad intenzitu dopravy v rôznych častiach mesta v určitom čase alebo dostupnosť autobusov a električiek.

V súčasnosti existuje pomerne málo nástrojov na vizualizáciu dát o verejnej doprave v Brne a tieto existujúce nástroje často poskytujú len vizualizáciu statických dát a nie sú navrhnuté tak, aby sa dynamicky menili na základe vstupov užívateľa.

Cieľom tejto práce bolo vyvinúť nástroj, ktorý spracuje zbierky dát týkajúcich sa verejnej dopravy v Brne a vytvorí grafickú reprezentáciu týchto dát tak, aby bolo možné ich ľahšie pochopiť a analyzovať.

Najprv bolo potrebné oboznámiť sa s vizualizáciou geopriestorových dát v širšom kontexte a preskúmať moderné metódy a nástroje. Potom bolo potrebné preštudovať koncept využitia veľkých objemov dát v kontexte verejnej dopravy a preskúmať súčasné riešenia, ktoré sú založené na dátach z verejne dostupných zdrojov. Ďalší krok zahŕňal skúmanie v súčasnosti používaných nástrojov na vizualizáciu dopravných dát v Brne a hľadanie možných zlepšení. Na základe týchto poznatkov bolo potom možné navrhnúť architektúru a základné funkcie nástroja tak, aby spĺňal požiadavky užívateľov.

Výsledkom je nástroj, ktorý dokáže vizualizovať dynamické zmeny zaujímavých dát o prvkoch verejnej dopravy v Brne na základe vstupov od užívateľov. Tento nástroj možno využiť na analýzu hustoty spojov a prestupov, dĺžky trvania ciest alebo polohy a meškania vozidiel hromadnej dopravy v reálnom čase. Nástroj je schopný aktualizovať dáta o doprave z oficiálnych zdrojov na plánovanej báze. Vývoj nástroja preukázal veľkú praktickú využiteľnosť GTFS dát a nástroj sa dokáže ľahko prispôsobiť akýmkoľvek zmenám vstupných dát v tomto štandardizovanom formáte.

Vizualizačný nástroj je implementovaný ako webová aplikácia s prístupom klient-server, kde server zabezpečuje zber a spracovanie dát, ako aj odosielanie odpovedí na požiadavky klienta, zatiaľ čo úlohou klienta je poskytnúť rozhranie pre interakciu užívateľa, odosielanie požiadaviek na dáta a zobrazovanie vizualizovaných dát. Výhodou tohto prístupu je, že aplikácia je schopná bežať vo webovom prehliadači, takže je nezávislá od operačných systémov a je prístupná pre každého. Rozhranie pre interakciu užívateľa zahrňuje mapu a rozťahovací bočný panel so záložkami, kde každá záložka predstavuje jednu z kľúčových funkcií nástroja.

Hoci v súčasnosti existujú väčšie a komplexnejšie aplikácie vyvinuté tímami s väčšími zdrojmi, tento nástroj sa zameriava na veci, ktoré týmto aplikáciám chýbajú, ako je napríklad prehrávanie časozberu nad dátami, ktoré sa menia v čase, alebo rozšírenie funkcionality pre vizualizáciu dĺžky trvania ciest z hľadiska užívateľského zážitku. Tento nástroj by mohol byť v praxi využitý plánovačmi verejnej dopravy, čo im umožní prijímať informované rozhodnutia v dynamickom mestskom prostredí.

Dynamic Visualisation Tool for Brno Public Transport Data

Declaration

I hereby declare that this Bachelor's thesis was prepared as an original work by the author under the supervision of Ing. Juraj Lazúr. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

. . . .
Patrik Gáfrik
May 5, 2024

Acknowledgements

First of all, I would like to thank my supervisor Ing. Juraj Lazúr for his very helpful and diligent approach and for providing me with all the information, resources and valuable advice necessary for the preparation of this thesis. I am also deeply thankful to my family and friends for their collective support.

Contents

Chapter 1

Introduction

Public transport is one of the most used ways of transport in cities around the world. In the city of Brno, it accounts for almost 50% of all ways of transport used by its inhabitants which makes it the most common way of transport within the city followed by car transport which accounts for 32% [18]. It is in the interest of our society to promote the use of public transport as an alternative to individual car transport in order to improve air quality as using public transport significantly reduces carbon emissions compared to car transport. In addition, encouraging the use of public transport presents a number of other benefits, such as lower traffic congestion leading to fewer traffic accidents.

To make public transport work as efficiently as possible the cities are challenged to develop various strategies. The decisions made within these strategies are based on the analysis of collected data. This data may represent various factors, such as traffic intensity in different parts of the city at specific times or the availability of buses and trams.

Currently, there are relatively few tools for visualizing public transport data in Brno, and these existing tools often provide visualization of static data only and are not designed to dynamically change based on user's input.

The purpose of this thesis is to develop a tool that processes collections of data related to public transport in Brno and creates a graphical representation of this data so that it is easier to understand, analyze, and make decisions based on the results. The tool should be able to offer the user a look at how the interesting data is dynamically changing over time. It should also provide an option to visualize data based on user's preferences to add interactivity to data analysis experience.

The chapter 2 describes geospatial data in general and the methods of visualization. The chapter 3 deals with processing and visualization of large volumes of data, which is one of the key concepts that needs to be researched before designing such tool. The chapter 4 analyzes currently available solutions for Brno public transport visualization and tries to discover the shortcomings of these solutions and the user requirements. The chapter 5 focuses on the design of the visualization tool and informs about the rationale behind utilized implementation methods. The chapter 6 describes the tool implementation itself, along with the problems that had to be solved. Finally, the chapter 7 deals with testing of data reusability and functionality of the implemented tool with a sample of users.

Chapter 2

Geospatial Data Visualization

Public transport systems produce huge amounts of geospatial data every day. In today's world where data-driven decision-making is becoming more and more useful for strategy development in all kinds of fields including public transport, the data needs to be analyzed [22]. However, the data is collected in raw and visually unappealing format and doing analysis would be tremendously inefficient and impractical as human brains are not designed to process data that is not visualized. To deal with this issue, one must study the concept of geospatial data and design effective visualization methods. Geographic visualizations always played an important role in human history, especially in the earth sciences, long before computer visualizations became popular [19]. Without geospatial data visualization, today's challenges in big data applications such as earth observation, geographic information system/building information modeling (GIS/BIM) integration, and 3D/4D city planning cannot be solved [6].

The most common way to visualize geospatial data is a map. The science that deals with the creation and use of maps is known as cartography [12]. The oldest maps were probably drawn in the sand thousands of years ago. Obviously, drawing maps in sand has many disadvantages and over the centuries more reliable methods of mapmaking have been discovered [12].

Currently, the most widespread way of visualizing geospatial data are interactive web applications or geographic information systems [13]. The most significant advantage of these applications in comparison to standard paper maps is that they are simple to disseminate, and in case modifications are made, the supplier simply needs to update a single map to ensure that all users have access to the most recent version [12]. Another benefit is the interactivity as users expect maps to be clickable and dynamic [13].

The intention of this chapter is to introduce the issue of geospatial data visualization in a broader context and describe currently used methods and visualization tools.

2.1 What is geospatial data?

Geospatial data is information about any objects or events that are located on the surface of the Earth. It includes location information (usually coordinates on the earth) as well as information about the object or event itself [21]. It may also include temporal data that refers to the moment in time for which both the locational and attribute data are valid. These three aspects provide answers to three basic questions: 'Where?', 'What?' and 'When?' and define the nature of an object [13]. An object's location, properties, and time can be defined in multiple ways, such as different coordinate systems, multiple variables, or different kinds of time [13]. The figure 2.1 describes the three key components of geospatial data: (a) object's location, attribute, and time; (b) the object itself; (c) detailed characteristics.

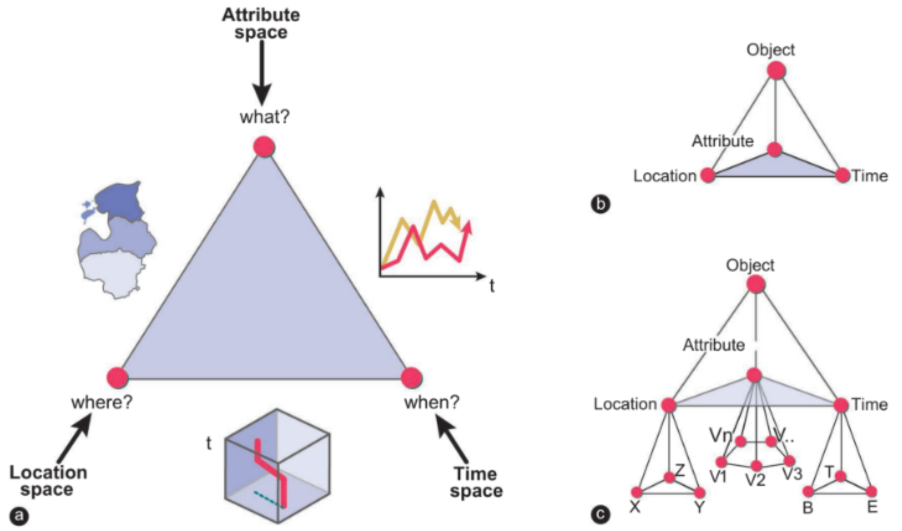


Figure 2.1: The characteristics of geospatial data[1]

2.2 GIS

The most common modern way of visualizing geospatial data is the use of geographic information systems (GIS) [13]. GIS is a computer-based information system for acquiring, managing, analyzing, modeling and visualizing geoinformation. The data it uses describes the geometry, topology, attributes, and dynamics of geoobjects [9].

Currently, GIS is used in almost all academic fields and occupations that need geographic information to carry out their duties or address their issues [13]. Its ability to combine geospatial and non-geospatial data from many sources in a geospatial analysis process is what makes GIS a very useful tool [13]. The figure 2.2 shows a schematic representation of GIS and all its components.

There are three crucial parts that can be thought of as making up a GIS [13]:

- computer hardware
- software applications and modules
- people connected through databases and network infrastructure

Any information that contains a location can be used by GIS. There are numerous ways to specify the location, including using latitude and longitude, an address, or a ZIP code [1]. It is possible to see spatial linkages and linear networks using GIS technology. In a GIS, highways, rivers, and public utility grids are frequently depicted as linear networks, sometimes known as geometric networks [1]. A road or highway may be indicated by a line on a map. However, using GIS layers, the road may also represent the border of a school district, a public park, or another area of habitation [1].

In order for the data from all the many maps and sources to fit together on the same scale, GIS must align it. The correspondence between a distance on a map and its actual measurement on Earth is known as a scale [1].

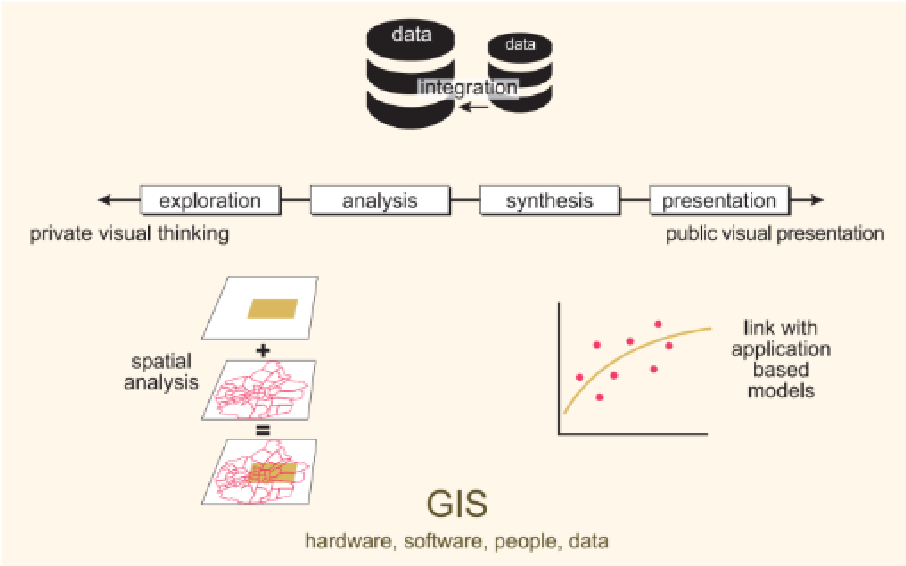


Figure 2.2: View on GIS[2]

2.3 Methods of visualization

There are many ways how geospatial data are visualized - using different styles of maps [2]. Certain map styles are more effective at representing particular types of information than others, depending on their benefits and limitations [2]. It is important to take this into consideration and choose the right map style based on the type of data that should be visualized [2].

Point distribution maps and symbol maps

The most basic geovisualizations for the graphical representation of datasets that contain elements with only geospatial information are point distribution maps [10]. If the elements contain the geospatial and one descriptive numerical value, they are usually shown on symbol maps [10]. The value of the symbol is determined by its physical size, which could be a circle or glyph. Since it can be distinguished between an element's size, shape, and color, symbol maps are also helpful when the element has two or three descriptive attributes [10]. For example, it is useful when visualizing the results of a country's election where the position and color of the symbol would indicate the election's winner in different regions while the size of that symbol would represent the number of voters. The downside of using this type of map is that trying to put too many data points on a large-scale map

[1]Figure source: https://www.taylorfrancis.com/books/mono/10.1201/9780429464195/cartography-menno-jan-kraak-ferjan-ormeling

when they are spread across small geographic areas can lead to overlapping [2]. The figure 2.3 shows an example of a symbol map visualization method.

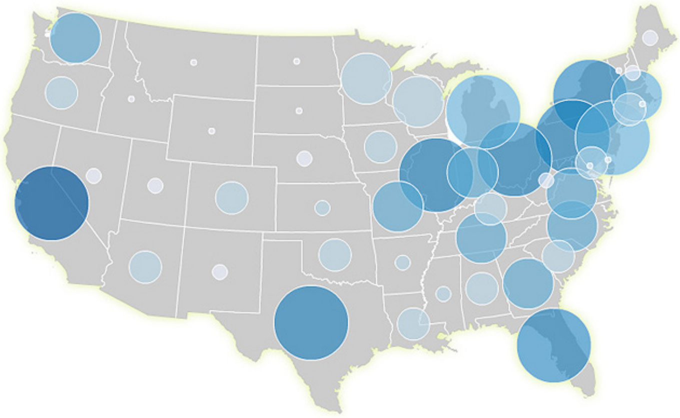


Figure 2.3: Visualization using symbol map[3]

Choropleth Map

Choropleth maps display data by designating various zones with various colors or shading patterns. Each shade of a color or pattern of shading represents a distinct value or range of values that a variable may have [2]. This type of map is effective for visualizing data clusters across a geographical area while retaining the context of regional boundaries [2]. However, larger sections might draw undue attention despite not necessarily having a high concentration of the measured variable. For instance, in a map of the United States, states with larger land areas might overshadow smaller states, even if they are more relevant to the analysis [2]. In such cases, it may be necessary to include an inset map to highlight smaller areas that could otherwise be overlooked [2]. The figure 2.4 visualizes percentage of US population that uses public transport, where the darker color intensity represents higher percentage.

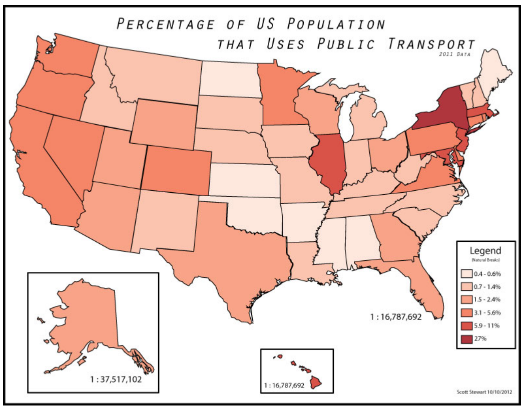


Figure 2.4: Visualization using choropleth map[4]

[2]Figure source: https://www.taylorfrancis.com/books/mono/10.1201/9780429464195/cartography-menno-jan-kraak-ferjan-ormeling

[3]Figure source: https://www.axismaps.com/guide/bivariate-proportional-symbols

Heat Map

A heat map resembles a choropleth map by utilizing colors or shades to depict various values or ranges. Yet, unlike choropleth maps, it showcases these values and ranges as a smooth continuum rather than discrete cells limited by geographic or political boundaries [2].

This approach allows for a more nuanced visualization of patterns indicating high („hot spots") and low concentrations of a variable. However, achieving this precision often involves converting discrete data points into a continuous spectrum through algorithms, which may sacrifice some level of accuracy.[2]. The figure 2.5 shows an example of heat map visualization.

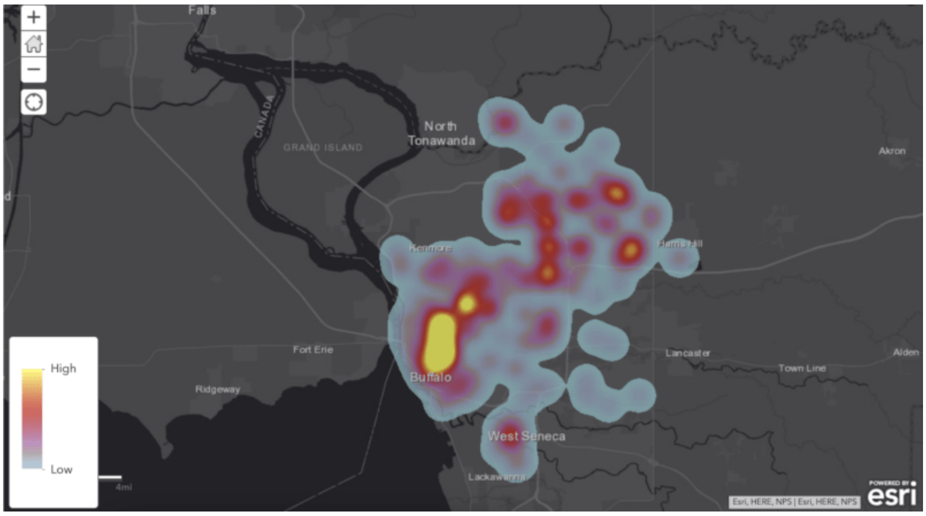


Figure 2.5: Visualization using heat map[5]

2.4 Currently used tools

Nowadays, there are various tools for visualizing geospatial data which can be put in different categories. For example, there are already-built robust web applications ready to interact with the user. On the other hand, there are libraries and APIs that were made for developers and are meant to be integrated in newly developed applications.

This subchapter lists and describes the most relevant visualization tools in terms of their popularity and usefulness. These tools provide a variety of features and abilities to aid in the efficient analysis and presentation of geographic data.

Leaflet.js[6]

Leaflet.js is an open-source JavaScript library with a thriving and active community that creates mobile-friendly interactive maps. The Leaflet includes many features and plugins that enable user to use a map for practically anything.

[4]Figure source: https://scottmaps.wordpress.com/2012/11/06/choropleth-map-color-version-percentage-of-us/

[5]Figure source: https://www.safegraph.com/guides/visualizing-geospatial-data

[6]Ref.https://leafletjs.com/

It is used by the major new media outlets, government organizations, startups, and internet businesses with millions of users [8]. It provides options to plot point- and shape-based markers on a map, control which map layer is displayed, and navigate the map using zoom-controls and panning [7].

As the figure 2.6 shows, the base of a Leaflet.js application is the map object which is at the very bottom layer. On top of this layer other layers can be added - for example a tile layer to adjust the layer's maximum zoom level, the credit text, and the URL template for the tile images.

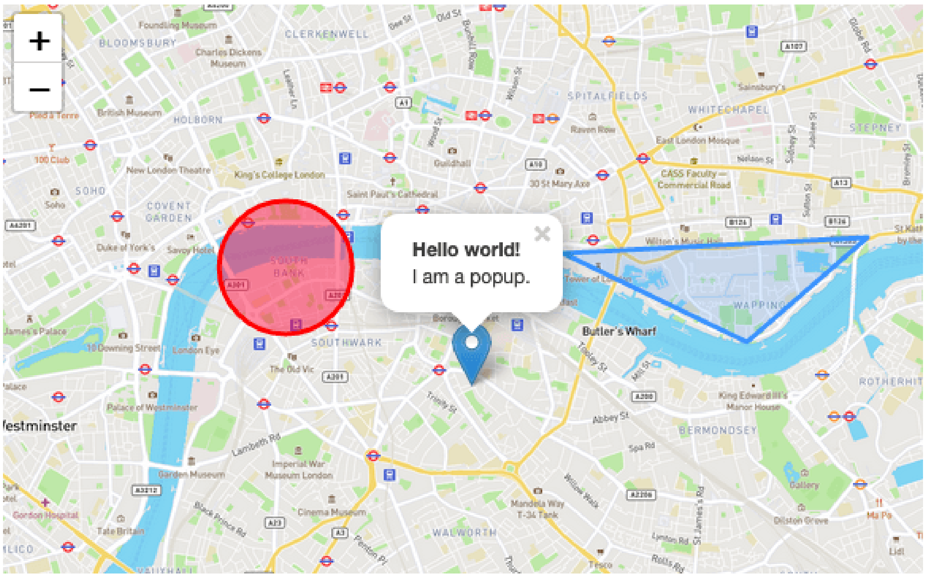


Figure 2.6: Leaflet.js map[7]

OpenStreetMap[8]

OpenStreetMap is an open-source project to build a free geographic database of the world. Its aim is to eventually have a record of every single geographic feature on the planet [5]. While this started with mapping streets, it has already gone far beyond that to include footpaths, buildings, waterways, pipelines, woodland and even individual trees [5]. The database is built by contributors, usually called mappers within OpenStreetMap, who gather information by driving, cycling, or walking along streets and paths recording their every move using GPS receivers. This information is then used to create a set of points and lines that can be turned into maps or used for navigation [5].

In comparison to other visualization tools OSM allows users to alter and modify the map data as needed which makes it a suitable tool for particular use cases. It also offers the option to display and reuse elements and overlays created by the community [5].

The data from OpenStreetMap is available for use by anyone, for any purpose. It is distributed under a license that allows you to duplicate, modify, and distribute the data [5]. Therefore, it is possible to use it in commercial projects. The figure 2.7 shows the OpenStreetMap interface for map editing.

[7]Figure source: https://leafletjs.com/examples/quick-start/
[8]Ref.https://www.openstreetmap.org/

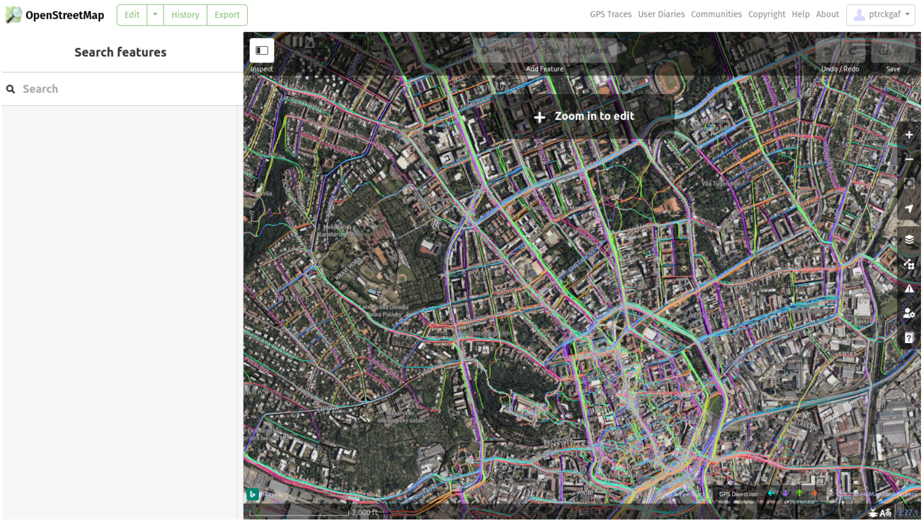


Figure 2.7: OpenStreetMap Editor[9]

Google Maps API

Google Maps is the most popular platform for location data and visualization on the web [11]. However, Google Maps' capabilities go far beyond its mobile app; if somebody manages a company website or app and wishes to incorporate location displays in any way, Google Maps provides developer APIs that give access to Google's vast geographic data [11]. Using Google Maps API you can customize and display interactive maps on your website. Maps can be used by travel websites to assist users in creating itineraries [11]. Google Maps may be used by delivery services or ride-sharing apps to display driver routes. Additionally, Google Maps data updates in real-time, ensuring that any maps you produce using the Google API are always up-to-date for users [11]. The figure 2.8 shows an example of Google Maps API usage.

Figure 2.8: Google Maps API[10]

[9] Figure source: https://www.openstreetmap.org/
[10] Figure source: https://blog.hubspot.com/website/google-maps-api

Geovisto

Geovisto is a toolkit for creating geospatial data visualizations that can be used in web-based dashboard applications or as a component of visual analytics workflows. It combines the capabilities of React, Leaflet and D3.js frameworks to process geospatial data and display it on reusable map widgets [10].

It focuses on improving the limitations of other visualization tools in use - it supports the upload of input data in object-oriented format (GeoJSON), offers wider configuration options and better options of interaction with user interface [10]. As can be seen in figure 2.9, the UI is made up of separate layers that present data in various ways (such as choropleth, markers, and connections). The user can choose which layer(s) to include in the map because each layer is independent [10].

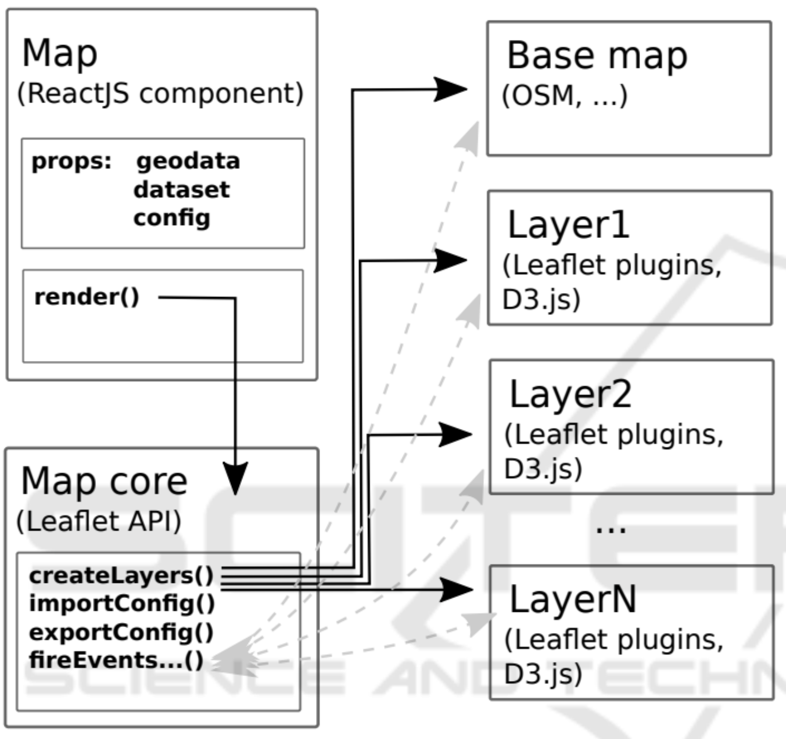


Figure 2.9: Geovisto architecture[11]

[11]Figure source: [10]

Chapter 3

Big Data in Public Transport

Big Data is a new paradigm that is currently gaining a lot of attention worldwide, particularly within the transportation industry, and it is considered as a new pledge to efficiently handle all the data needed to offer passengers more customized transportation experience in addition to safer, cleaner, and more efficient modes of transportation [22]. Compared to the early years of this century, transportation and mobility data were captured, processed, and analyzed on much smaller scales than they are today [22]. The importance of data-intensive tasks like integration, visualization, querying, and analysis for large-scale real-time systems is growing due to the volume, variety (source, type, and format), and variability of transport and mobility data [22]. The utilization of Big Data methods has the capacity to exploit information and handle these tasks at unprecedented scales. This creates new opportunities for transportation models, services and applications that remained unexplored - mainly because the required data had not been collected nor processed to date [22]. One example of the former is real-time processing of data streams for traffic control - while this particular issue is not new, the decision process gets crushed by the volume, speed, and variety at which data streams are nowadays collected [22].

Solutions and tools based on this new technological paradigm are able to collect, handle, and analyze enormous amounts of structured and unstructured data to enhance the transportation sector and address the issues raised above [22]. The main strategy involves developing transport and mobility services based on principles of the Big Data paradigm while extracting valuable insights from the vast amount of available real-time data [22]. Ultimately, the goal is to ensure that the transport industry derives meaningful value from its data, leading to future developments focused on enhancing safety, environmental sustainability, and efficiency in transportation methods [22].

This chapter discusses several approaches of Big Data utilization in the context of transport data processing. It describes how big data differs from regular data and why it is an important part of transport visualization. It informs about the main sources from which big data can be drawn and the different analytics methods. It also highlights the benefits of real-time processing of large volumes of data and categorises systems using this functionality. Finally, it shows examples of the use of publicly open data in public transport systems.

3.1 Core Features of transportation Big Data

There are three key features that make Big Data differ from 'normal' data according to Laney's 3 V model [22]:

- **Volume** represents the total amount of data captured, managed and analyzed. Over time, datasets have grown significantly, reaching sizes of several dozen terabytes to several petabytes in a single dataset today [22].

- **Velocity** represents the description of two ideas: the frequency at which data instances are generated, and the speed at which these data samples are received and processed. Current applications require real-time processing speed in the nanosecond range. This is a strict requirement that involves critical decisions in the design of computing infrastructure and models for data analysis. Big Data technology makes it possible to analyze data as it is being generated, eliminating the need to even store the data in databases [22].

- **Variety** represents the diversity of collected datasets from the varied digitalized domains associated with the relevant application or service. The requirement to collectively process and analyze these datasets also has significant implications for system design, necessitating the inclusion of functionalities for data integration and fusion [22].

3.2 Big Data sources

As already covered, Big Data represents a significant change in the amount, variety and availability of data. According to [22] in the domain of transportation, three main Big Data sources can be identified:

- **Social media** platforms these days retain massive amounts of geo-tagged data that, when eventually analyzed, may reveal important information. It is considered an essential data source for applications, that operate with geo-localised knowledge about the mobility of people who produce data traces [22].

- **Sensor data** collected by the activity of embedded systems driven by the IoT paradigm significantly innovates the transport industry [22].

- **Open Data**, in the context of transport, refers to data that is 'open,' meaning it is freely accessible, usable, modifiable, and shareable by anyone for any purpose. The availability of open transport data holds potential benefits for both citizens and the public sector. Governments are encouraged to make efforts in publishing transport-related information in machine-readable and easily consumable formats [22]. A growing number of countries have recognized this potential and have started publishing data repositories on national public transportation services rather than at the level of individual transit systems [22]. Notably, the United Kingdom stands out as one of the pioneers, having introduced a National Public Transport Data Repository [22]. This repository recorded every bus, train, and coach trip across the country from 2004 to 2011, facilitating analyses such as identifying regions with deficient bus services, among other minor purposes [22]. An example of such data source could be the usage of static and real-time GTFS datasets that is described later in this chapter.

3.3 Big Data Analytics methods

Big Data Analytics, understood as the pinnacle of a Big Data platform, involves the application of pattern inference, learning, and optimization techniques. This is currently a prominent global technological trend, particularly in the transportation industry [22]. Depending on the analytics methods and technologies used in models of Big Data Analytics, three essential approaches are recognized [22]:

- **Descriptive Analytics** covers models that aim to summarize and explain collected data by eliminating irrelevant information and deducing regularities or patterns, often manifested as groups or fitted statistical distributions. It integrates aspects of unsupervised machine learning (such as clustering and outlier detection), pattern recognition, and statistics to identify consistent patterns in data [22].
- **Predictive Analytics** involves the application of learning algorithms on supervised data examples to capture the relationship between observed features or predictors and a target variable. Trained models can then predict the value of an unknown target variable for new input data, even if it doesn't match any historical examples used to build the model. In the transportation domain, these learning models find diverse applications, including self-driving technologies for real-time perception or prediction of traffic scenes [22].
- **Prescriptive Analytics** suggests the optimal action or decision from a set of possibilities, utilizing optimization techniques, expert systems, and elements from Computational Intelligence, Mathematical Programming, and Operations Research. Prescriptive analytics leverages the knowledge acquired from its descriptive and predictive counterparts. Examples of usage include applications in logistics, active traffic management, and user-driver-passenger information services [22].

3.4 Real-Time Big Data Processing

The ability to process and analyze data in real-time is the next frontier for innovation and productivity [22]. Analytical processes that once required months, days, or hours have been reduced to minutes, seconds, or even fractions of seconds [22]. Nevertheless, the reduced processing durations have elevated user expectations. Users now demand results in less than a minute, almost at the speed of thought [22].

When talking about transportation, modern infrastructure and platforms deployed are not built for processing data in real-time, nor are they capable of analyzing the data at the speeds required by critical applications (like safety) [22]. This is a big issue, because for the majority of transportation and mobility problems, a short response time is essential to ensure that information and decision-making are closely linked in time, and thus, useful in practice [22].

Based on the necessary latency level three major categories of Big Data systems can be identified [22]:

- **Batch systems** currently operate with non-existing or extremely relaxed latency requirements, often ranging from hours to days. Essentially, these systems follow a sequential approach: gathering data, processing it, and generating results in distinct stages. An illustration of this is observed in delay-insensitive applications, such as control panels used in Intelligent Transportation Systems (ITS) for tasks like monitoring and planning medium-term freight transportation (e.g., weekly cycles) [22].

- **Soft real-time systems** operating under mild real-time limitations typically in the range of seconds to minutes, include online systems where user responses need to be fast but not exceptionally immediate. It could be used in systems such as the query service on the occupation of parking lots in a SmartCity, where more responsive interface significantly improves the user experience, although the application's capability to handle real-time latency constraints is not strictly required [22].
- **Hard real-time systems** are characterized by ultra-low latency constraints typically ranging from milliseconds to nanoseconds. This requirement is so crucial to the system, that if it is not met by the designed data processing lifecycle, the system fails and cannot support the application for which it had been designed. Instances of such latency-critical applications include predicting incidents and calculating the road state for autonomous vehicles [22].

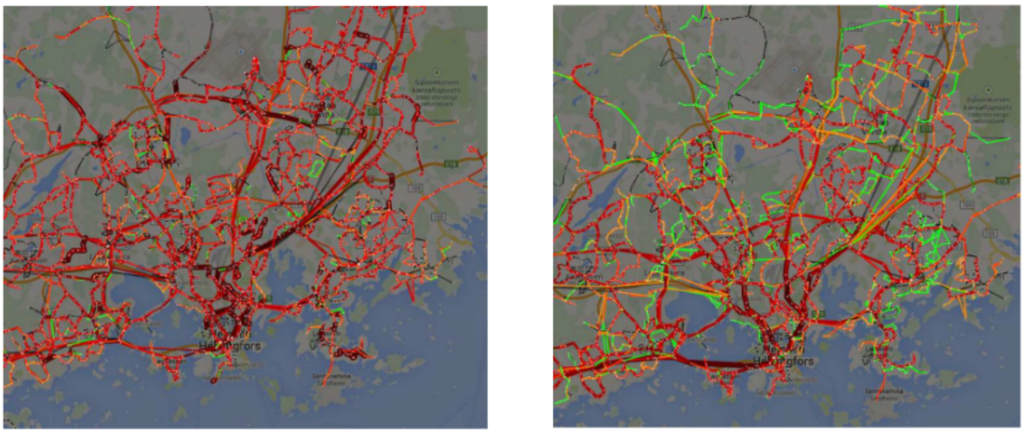
3.5 Open Data usage in public transport systems

The concept of Open Data represents the idea that data generated by public entities such as governments and institutions should be published in an open, machine-readable format so it can be used by developers to build applications [17]. It is an important Big Data source for the transportation domain and can provide valuable knowledge about transport systems [22].

The processing of this data in the development of modern visualization systems consists of several steps [14]. First, it is necessary to find data sources that are available under public domain or licenses which allow redistributing the data and using it for scientific purposes [14]. This data normally has a very large volume and in most cases comes in text files such as CSV. Searching through these files would be very inconvenient and therefore the files are imported into a relational database once downloaded [14]. Then, the database must be filtered both spatially and temporally to ensure ease-of-access and consistency across the provided data extracts [14]. Finally, it is possible to query the database and implement various algorithms that transform the data into the resulting form that is interesting for the visualization [14].

Urban Traffic Analysis

Traffic analysis is one of the major topic where it is useful to utilize Open Data concept. It allows to get information about traffic density or the usage of public transport [22]. The work mentioned in [25] similarly focuses on analyzing data of the bus transport service. It involves the analysis of 45 million samples of bus arrivals to gain insights into urban traffic in the Helsinki area (Finland) [22]. This analysis is done on publicly available visualized datasets collected by the city of Helsinki and uses graph theory and statistical methods to calculate the correlation between urban traffic and bus traffic [25]. The main visualization tool used in this work was Google Maps Service which has the capacity to visualize datasets from Open Data sources and display hot traffic areas in the city [25]. The figure 3.1 shows the comparison of visualized urban and bus traffic of Helsinki city area at different times.



(a) Urban Traffic (Mon 8-9 am) (b) Bus Traffic (Mon 8-9 am)

Figure 3.1: Urban and bus traffic analysis of Helsinki city area[1]

Public Transport Network Analysis

Designing public transport networks presents a complex task, requiring both the static structure of the network as well as the detailed schedules of vehicle departures [14]. Successful planning demands tools for the analysis and optimization of public transport networks and expertise of understanding the structural organization of these networks across cities [14].

The concept of developing such tools is addressed in research paper [14] which came up with the idea to collect public transport network datasets of 25 cities across the globe in publicly open GTFS format and convert it to multiple, easy-to-use data formats including SQLite databases or GeoJSON files so it can be used for public transport analysis. The figure 3.2 describes the data processing workflow that consists of first downloading the GTFS feeds from multiple sources by automated scripts, importing it into a relational SQLite database and performing spatial and temporal filtering of the data to ensure the consistency of resulting data extracts [14].

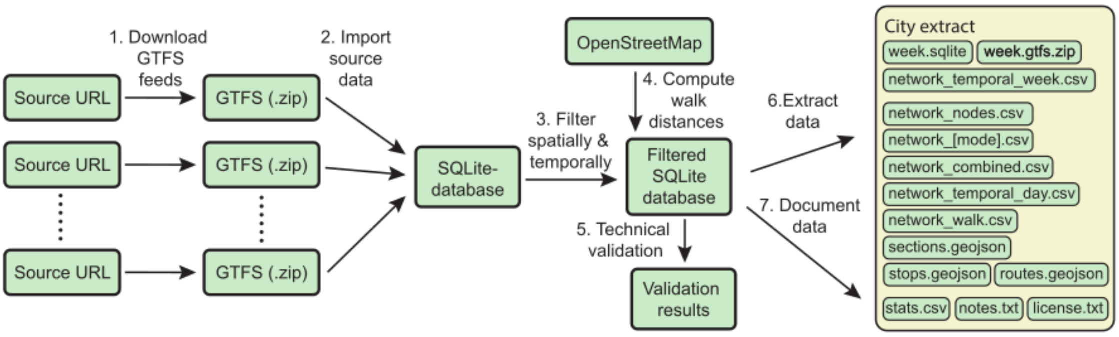


Figure 3.2: The workflow of GTFS data processing[2]

[1]Figure source: [25]
[2]Figure source: [14]

GTFS data

The General Transit Feed Specification (GTFS) establishes a standardized format for public transportation schedules and related geographic information, including station locations [15]. GTFS enables the provision of transit data in a common format, facilitating its utilization for analysis by engineers and researchers or the development of practical applications by developers [15]. A standard GTFS feed data consists of a collection of text files, each detailing various characteristics of public transit, including trips, routes, and sequences [15]. GTFS datasets have been employed in the creation of variety of applications, including trip planners, mobile applications, timetable generation software, tools for transit planning and operational analysis, as well as other applications [15].

GTFS Schedule is a GTFS feed that defines a common format for public transport schedules with associated geographic information [24]. It contains static transit data and is composed of a number of text (.txt) files that are contained in a single ZIP file [15]. Each file describes a particular aspect of transit information: stops, routes, trips, fares, etc [15].

GTFS Realtime is a GTFS feed specification that allows public transportation agencies to provide real-time updates about their fleet (such as vehicle location and road congestion level) to application developers [24]. However, it cannot provide practical real-time information without having a companion GTFS feed that defines the schedule [4].

Another example of using GTFS format in public transport visualizations is the subject of a research paper [20], of which the main goal is developing a visualization tool called PubtraVis that comprises six visualization modules each of them focusing on various operational aspects of transit systems: mobility, speed, flow, density, headway, and analysis. Users can compare two modules simultaneously. As the figure 3.3 shows, the mobility module shows transit vehicles' movement and the number of trips. The Speed module displays individual vehicle speeds. Flow module depicts transit flow direction at each station. Density module shows vehicle distribution. Headway module illustrates average time between vehicles [20].

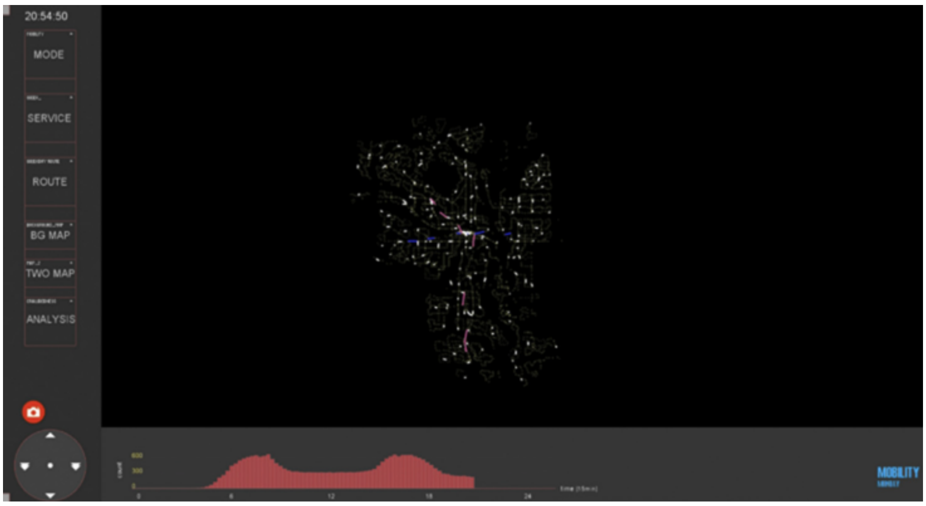


Figure 3.3: Mobility module of PubtraVis[3]

[3] Figure source: [20]

Chapter 4

Visualization of Brno Public Transport

Both, our planet and the region, have limited resources, which inevitably leads to the prioritization of the public transport over cars [3]. Therefore, the Integrated Transport System of the South Moravian Region (IDS JMK), which provides a cheap, fast, reliable and environmentally friendly transport option for 1.2 million people, was introduced in Brno and its surroundings in 2010. It has 161 zones, 322 lines, and hundreds of thousands of people rely on it every day [3].

Traffic planners are thus facing a difficult challenge of ensuring that the whole system runs as smooth and efficient as possible. With this never-ending task, spatial visualizations of timetable data allow them to get a unique look at its behavior and properties [3].

Brno collects and stores many datasets that contain information about the usage of transport in the city [7]. This information may include pedestrian movement in different parts of the city, bike traffic intensity, traffic accident localization or public transport routes and stops [7]. As mentioned earlier the goal is to get some value out of this data, and therefore it must be visualized. There are currently several tools for visualizing Brno's public transport data. However, the vast majority of these tools seem like they can only work with static data and are unable to adapt to dynamically changing dataset collection.

This chapter focuses on research and exploration of currently available solutions of Brno Public Transport data collection, processing and visualization. It tries to discover the shortcomings of these existing solutions in the context of user requirements, which is an important step before designing the tool.

4.1 data.Brno

The data.Brno open data platform was initially launched in 2018, reflecting Brno's commitment to transparency and openness [7]. This platform publishes over 140 different data sets aimed at three different user categories. The first category is the general public, since the platform allows them to gain insights into their city through resources like the widely embraced State of the City Report, which is consistently released at the end of April [7]. The second set of users includes professionals and students who can access interactive, long-term statistics about the city and applications available on the website, gaining significant popularity [7]. The third user category encompasses developers and IT professionals work-ing with machine-processable datasets to develop a wide range of applications including visualization tools [7].

data.Brno allows its users to choose from five distinct formats for each dataset, including CSV, Shapefile, KML, file geodatabase, and GeoJSON [7]. Additionally, JSON feeds with API interface can be generated for each dataset, making it simple to integrate them into systems or applications without requiring manual file downloads. The majority of datasets are updated every day, every week, or every month, but some have different intervals, like yearly or tens of seconds [7].

Explore Brno's Public Transport System[1]

Explore Brno's Public Transport System is a web application that aims to visualize public transport datasets collected by data.Brno. It displays the routes and stops of buses, trams and trains where each of these three types of transport is represented by its own color. It is also able to show the distance from five different zones of the city to the main train station in terms of time spent while commuting, including walking to the bus or tram stop. Additionally, it is capable of visualizing the frequency of connections as an important indicator of traffic accessibility of a place or the maximum wait time between two consecutive trips [3].

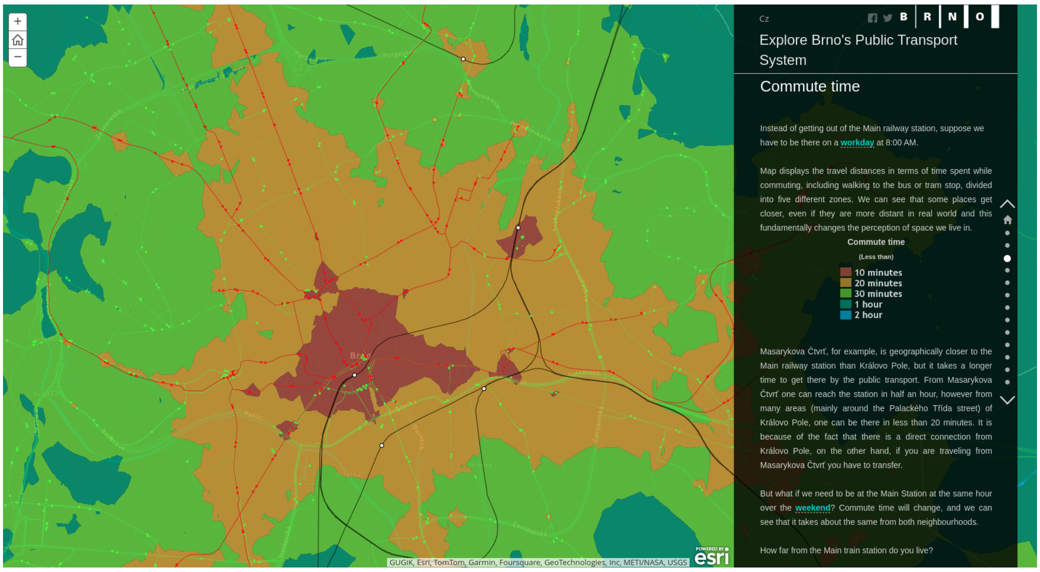


Figure 4.1: An interface of Explore Brno's Public Transport System

The figure 4.1 shows a choropleth map that visualizes travel time to stop „Hlavní nádraží" from different city zones as one of the tool's features. The problem is, however, that this tool only shows the travel time for this one specific stop, but does not provide the user an option to see what the travel time looks like for all the other stops. Another issue is that the travel time varies during the day based on factors like frequency of trips or number of transfers that the path requires in specific time of the day, but the tool only shows travel time for one specific hour. The solution to this would be adding a feature that allows to select any stop and time range the user wants and the map would dynamically adjust according to this input.

Delays of public transport vehicles[2]

Another application that utilizes data.Brno datasets is Delays of public transport vehicles. It provides graphical representation of the likelihood that a public transport vehicle will be delayed in a specific part of the city as well as the travel duration between two different stops. Users can select the bus or tram lines they want to visualize and they can see the heat map or point representation of geographical places where delays are most likely to occur. As the figure 4.2 shows, it also offers the feature to look at graphs of the travel duration of specific line between two stops in a specific hour of the day and the difference between the measured duration compared to the official duration provided by IDS JMK.

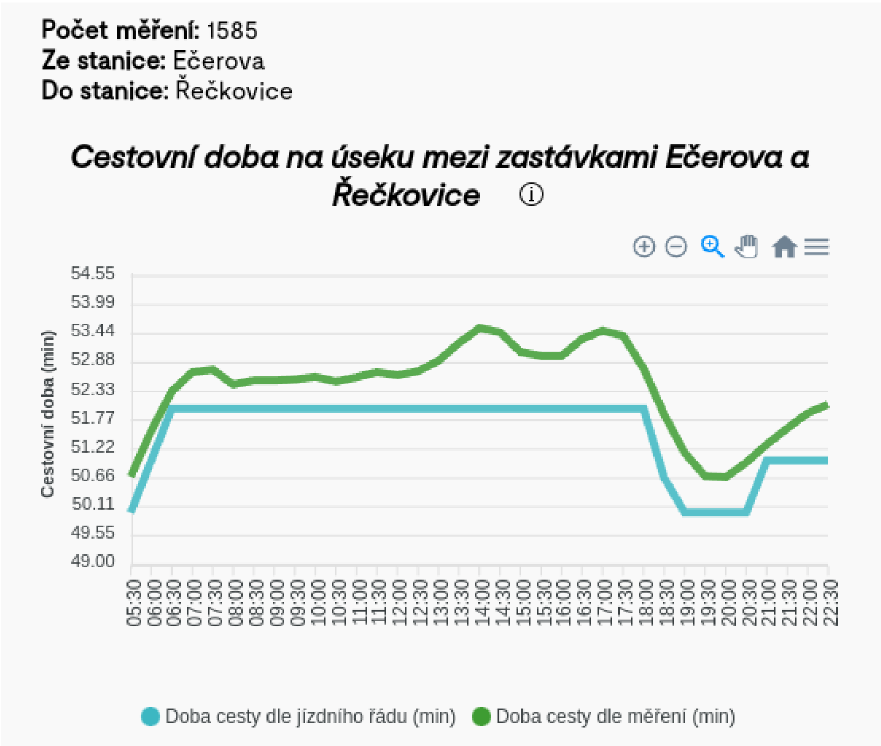


Figure 4.2: Travel duration between stops Ečerova and Řečkovice of tram line 1

Bike traffic intensity[3]

data.Brno also provides datasets about bike traffic. These datasets are used by Bike traffic intensity web application that visualizes data on bicycle accidents, data on current traffic measured by bicycle detectors, and the intensity of cyclists' and pedestrians' traffic. It shows the number of bicycle accidents that happened at specific location according to historical

[1] Ref.https://mestobrno.maps.arcgis.com/apps/MapJournal/index.html?appid=6ce3a916606e4c1996015613ef57e016

[2] Ref.https://data.brno.cz/apps/2d46c3ba8fa3440cb3d1fd8360648936/explore

data and the measured bike traffic intensity represented by the size and color of circles displayed on the map as can be seen in figure 4.3.

Figure 4.3: Visualization of Brno bike traffic intensity

4.2 Real-time positions of vehicles

There are several tools for real-time visualization of the positions of public transport vehicles. These tools can be important for the analysis and planning of traffic operations.

ODAE Public Transit Stream[4]

This tool is implemented using ArcGIS and uses live data transfer from the server. It shows the location of all public transport vehicles in the South Moravian Region and their associated attributes such as vehicle type, bearing, course or current delay. It provides an option to filter these vehicles by attributes and coordinates.

The figure 4.4 shows the visualization interface of this tool. A minor shortcoming of it is that all vehicles look the same so they are not visually categorized according to their attributes. For example, it would be appropriate to colour-code vehicles according to their current delay or vehicle type.

Monitoring of the IDS JMK operation[5]

This application showcases the real-time locations of all public transport vehicles, including regional trains, within the integrated transport network of the South Moravian Region [16]. It utilizes Mapy.cz maps and retrieves vehicle location data from the CEDRIS and RIS

[3] Ref. https://data.brno.cz/apps/c566dc900c774699a0b61e38283bebe7/explore

[4] Ref. https://gis.brno.cz/ags4/rest/services/ODAE_public_transit_stream/StreamServer?f=jsapi

traffic dispatching centers. Additionally, it provides in-depth vehicle details sourced from the BMHD.cz website. The app features filters enabling users to customize their view by selecting specific modes of transport, lines, or vehicle types [16]. The figure 4.5 shows the application's interface with an example of real-time information about specific vehicle.

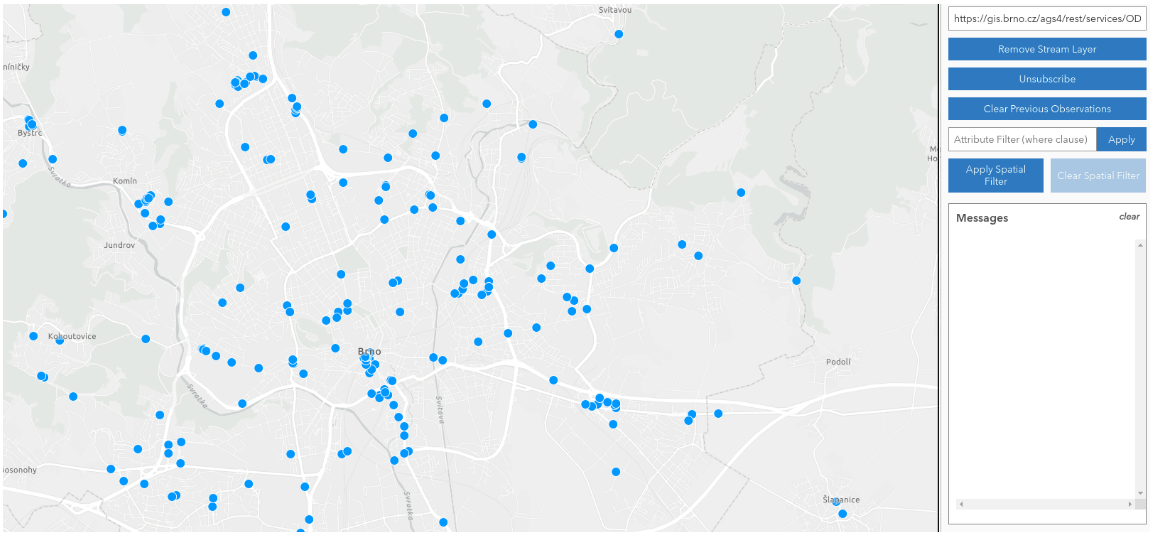


Figure 4.4: An interface of ODAE Public Transit Stream

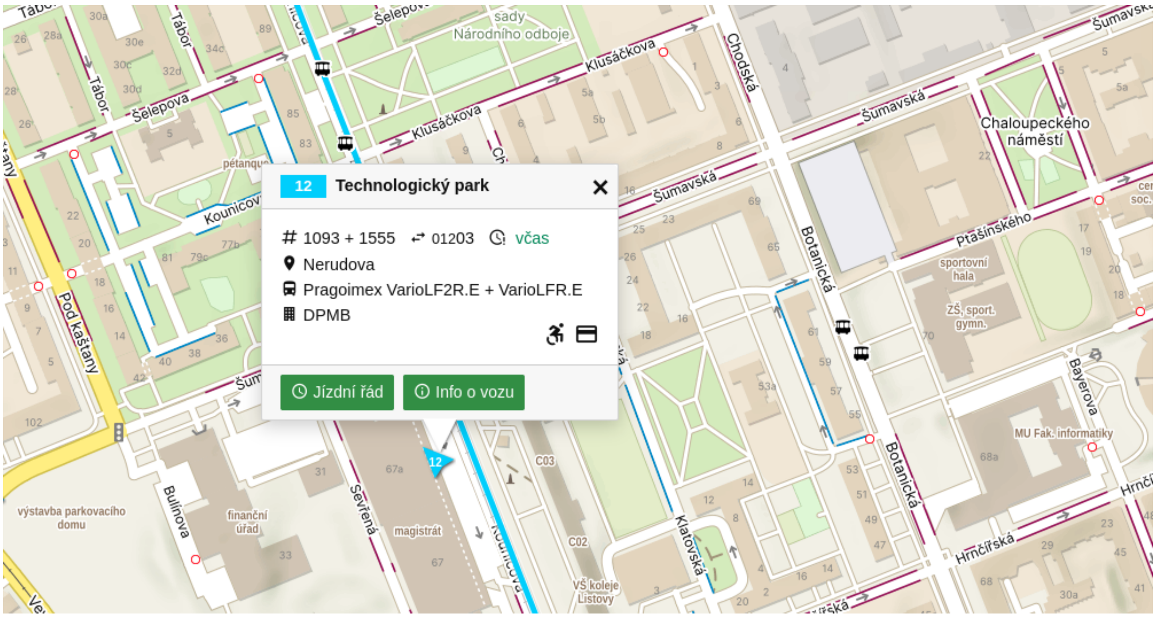


Figure 4.5: IDS JMK map with real-time location of the tram line 12

[5]Ref.https://mapa.idsjmk.cz/

Currently, there are several tools that could be useful for both static and real-time traffic data analysis in Brno. Some of these tools could be more interactive and provide more dynamic visualizations such as a timelapse of changing inputs.

One of the important problems of these tools is the inconsistency of data sources [16]. Each tool draws data in a different format and from different sources, and these tools are not able to share data with each other [16]. This causes each tool to visualize slightly different data, and users' data analysis may differ depending on which tool is used.

The solution is to develop tools based on a standardised data format. An example is the mentioned GTFS format, which is currently used by very few traffic data visualisation tools in Central and Eastern Europe [16]. Also the great advantage of this format is its reusability, i.e. if there is a data visualization application for one city that uses this format, it is very easy to rebuild it to visualize another city's data by just modifying the GTFS database.

Chapter 5

Visualization Tool Design

The main goal of this chapter is designing the visualization tool. It primarily focuses on available open data sources that will be used and also the tool architecture and its core features.

5.1 Tool architecture

The visualization tool will be implemented as a web application with a client-server approach, where the server will handle the data collection and processing as well as sending responses to client's requests, while the client will behave as the interface for user interaction, sending requests for data and displaying visualized data. The advantage of this approach is that the application will run in a web browser so it will be independent of operating systems and will be accessible to everyone. The interface for user interaction will include a map and an expandable sidebar with tabs, where each tab represents one of the tool's key features.

5.2 Core features

The tool will provide several key features, where each feature represents a visualisation of different public transport elements and the interesting data associated with them. The tool will provide visualization of routes and stops , density of trips and transfers, travel durations and real-time vehicle locations. The tool will also address the investigated shortcomings of current solutions such as adding the selection of the initial stop and time range to the functionality for the visualization of travel times or also the visualization of current delay of the vehicles in real-time.

Routes and stops

The tool should be able to visualise all public transport routes, providing the user with information about the route and the type of vehicle using that route. A suitable solution is categorizing the paths for trams, buses and trolleybuses and allowing to show and hide the desired paths by vehicle type.

Public transport stops in Brno are divided into three transit zones: 100, 101 and 510. That allows to categorize each stop into these zones and allow user to choose which zone they want to display. The figures 5.1 and 5.2 demostrate the design of routes and stops visualization.

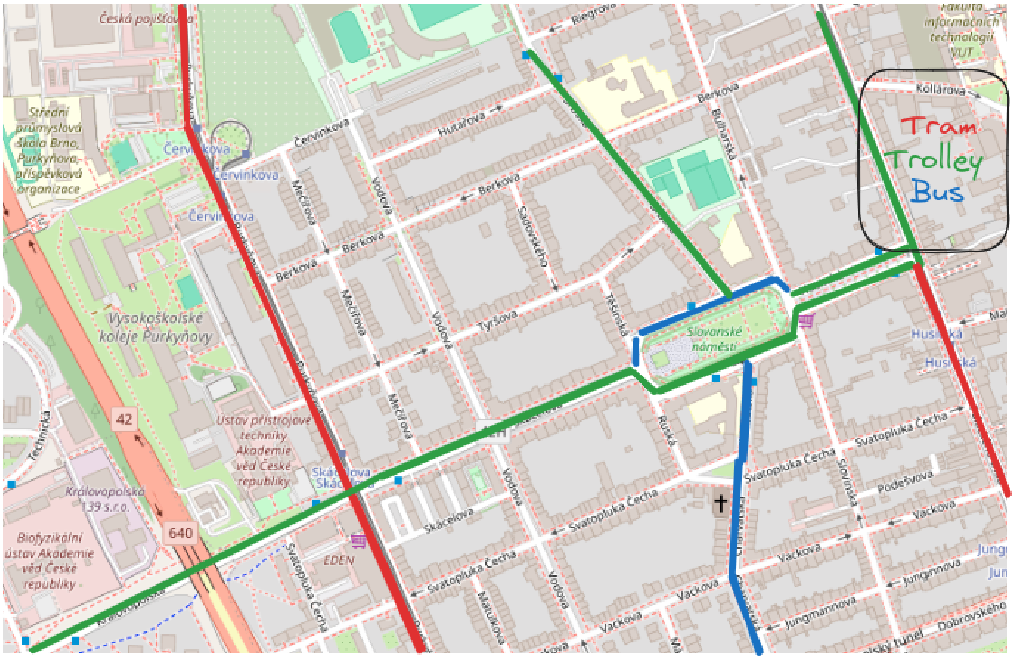


Figure 5.1: Routes visualization categorized by vehicle type

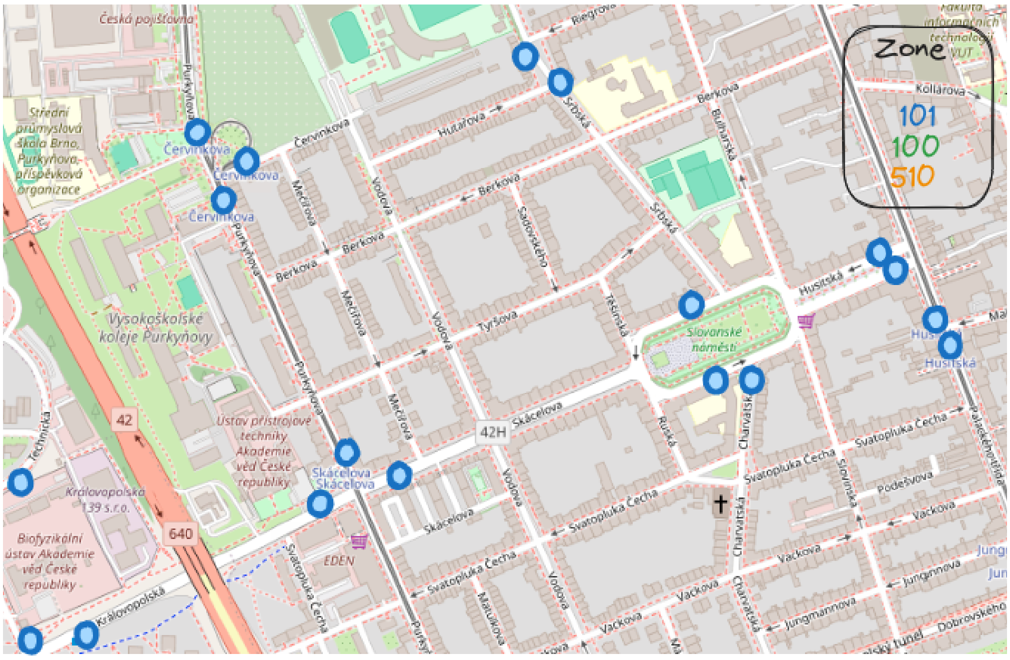


Figure 5.2: Stops visualization for transit zone 101

Trip frequency

An important measure of a public traffic accessibility is the frequency of connections in different locations. The most suitable way of visualizing location based data like in this case is a single heatmap, where the trip frequency is represented by the color intensity of each location, as can be seen in the figure 5.3. The trip frequency at each stop varies throughout the day so it would be interesting to visualize it for each time of the day and allow the user to compare the differences.

Ideally, the user should have an option to set a time range as an input and play a timelapse in which this range will periodically increment by the desired number of minutes. This will allow the user to look at differences in trip frequency in each time of the day and it adds dynamicity to the visualization.

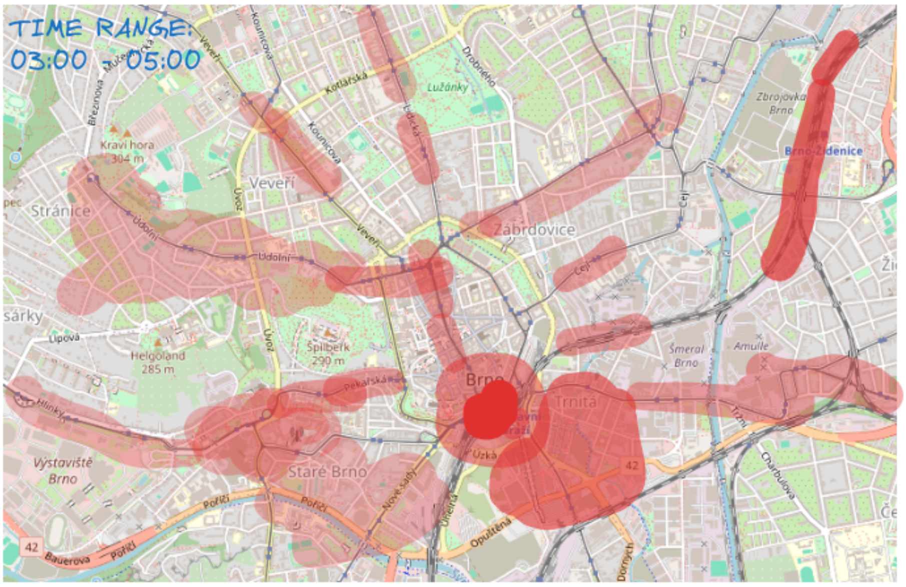


Figure 5.3: Heatmap visualization of trip frequency

Travel times

Another interesting feature would be the visualization of the travel time from the selected stop to different zones of the city. The optimal solution to this is a choropleth map where each zone would be assigned a color according to travel time to that zone, as illustrated in the figure 5.4. The duration of travel fluctuates throughout the day due to factors such as how often trips occur and the number of transfers needed at different times so again the interesting thing would be seeing the differences. Therefore, the user's input in this case would be the name of initial stop from which should the travel path begin and also the time range.

Real-time positions

Seeing the real-time locations of public transport vehicles is helpful for collecting statistics about the vehicles movement and effective journey planning. The tool will provide dynamic visualization of these real-time positions and provide the user with information about each vehicle such as vehicle's name, type and its current delay in minutes. The figure 5.5 demonstrates a map with real-time positions of vehicles, where the green color represents vehicles with no delay while the red color represents vehicles with delay two minutes or longer.

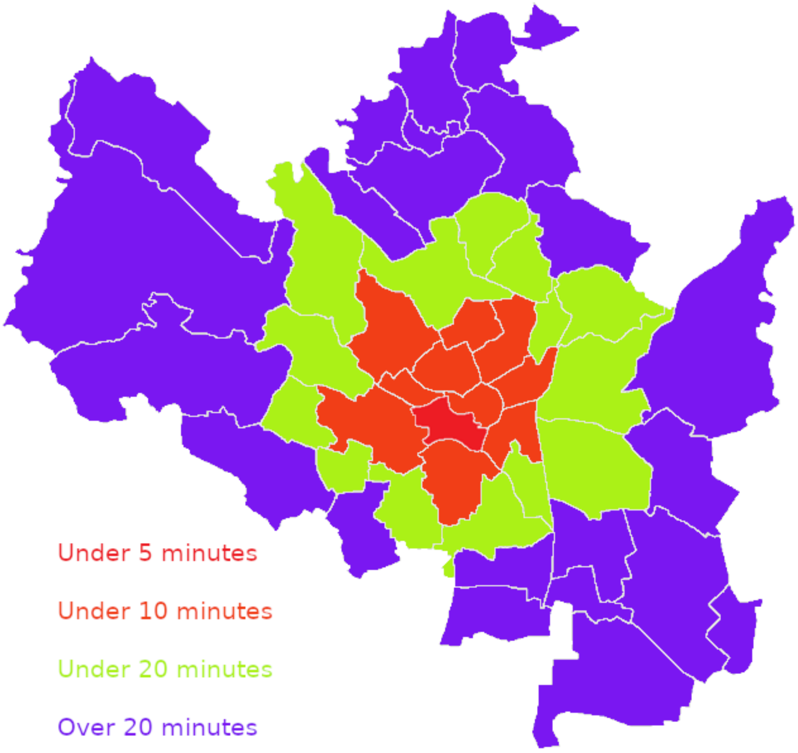


Figure 5.4: Choropleth visualization of travel times

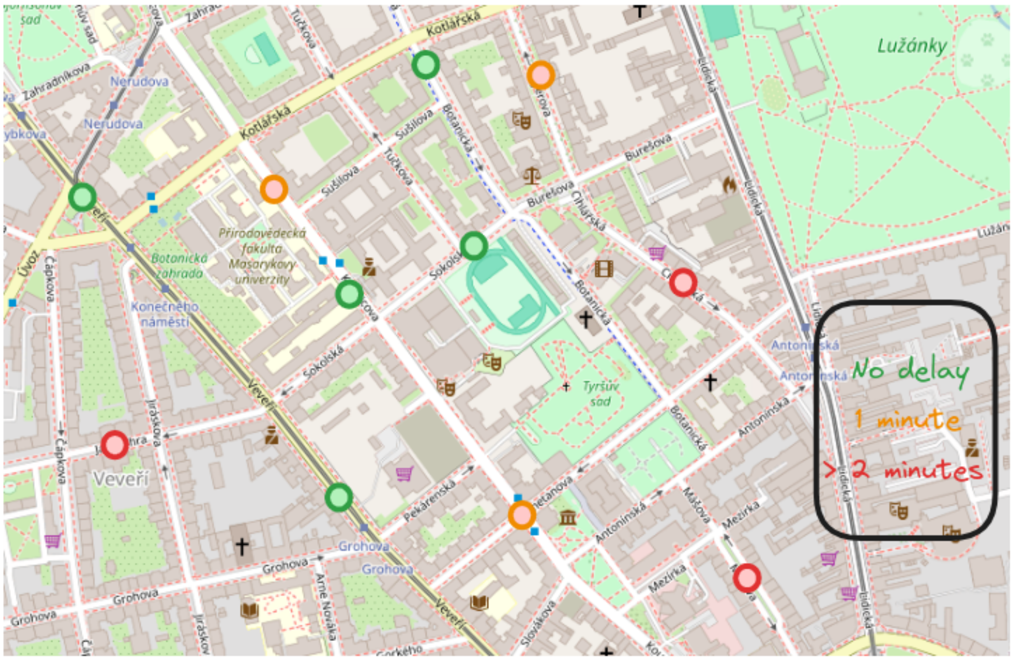


Figure 5.5: Visualization of real-time positions

Transfers

Number of transfers at public transport stops at specific time can provide valuable information about the transfer infrastructure and help designing improvements to transit network planning. Again this is location-based data so it will be appropriate to use a heatmap, where the transfer count at each stop will be represented by color intensity of heatmap points, which is illustrated in the figure 5.6. This feature will also implement a time range selection to see how transfer count changes during the day.

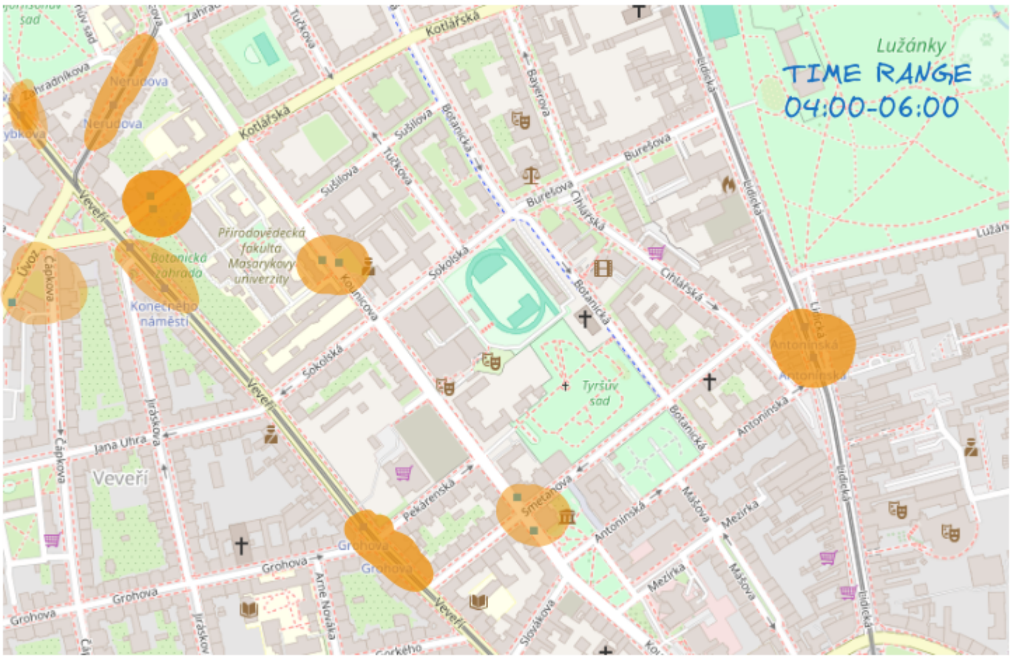


Figure 5.6: Heatmap visualization of transfers

5.3 Data model

When designing a visualization tool, data processing is one of the issues that need to be solved. The chapter 4 discussed the utilization of data.Brno datasets which is the most reliable data source for the designed tool, since it is the official public transport database of Brno and always provides up-to-date information. The application will use data from four of these datasets:

- Public transit GTFS static timetable data of the Integrated Transit System of the South Moravian Region[1]
- Public transit routes (GeoJSON)[2]
- Cadastral boundaries (GeoJSON)[3]
- Public transit real-time positional data (JSON)[4]

[1]Ref.https://data.brno.cz/datasets/379d2e9a7907460c8ca7fda1f3e84328/about
[2]Ref.https://data.brno.cz/datasets/37fc8ffb7c0b4f7fad33fc519415caa2_0/about
[3]Ref.https://data.brno.cz/datasets/8c797c879ff544359dd7798ba10e302d_0/about
[4]Ref.https://data.brno.cz/datasets/e8aa121910df41bb9a28e4ca34a263c7_0/about

GTFS static timetable data

The GTFS dataset contains a set of CSV files where each file represents one table in application's database. The data gets updated once per week every Sunday at midnight so there must be implemented a scheduler on server-side that is responsible for periodically downloading new data. In most basic scenarios, it consists of seven of the standard GTFS tables [17]:

- **agency** - contains information about one or more transit agencies.
- **stops** - contains information about all the stops.
- **routes** - contains information about the routes of the transit agency.
- **trips** - contains sequences of two or more stops that occur at a specific time.
- **stop_times** - contains vehicle arrival and departure times from individual stops for each trip.
- **calendar** – contains schedule information.
- **shapes** - the spatial representation of a route alignment so it can be accurately drawn on a map

GeoJSON data

GeoJSON is a widely used format for describing geographical features and their associated attributes [23]. It employs JSON structure and can represent various elements like points (such as addresses and locations), line strings (like streets and boundaries), polygons (including countries and land tracts), and combinations of these. Beyond physical entities, GeoJSON can also depict abstract concepts, like service coverage in mobile routing and navigation applications. [23]. Its biggest advantage is that it is very simple to visualize the data as Leaflet.js library provides GeoJSON component that can be fed with data in this format.

Public transit routes

The GTFS dataset mentioned above contains routes information, but it does not include its geographical coordinates that are required for the visualization. Therefore, the most suitable option is to use the Public transit routes dataset from data.Brno in GeoJSON format. It includes feature collection where each feature is of type LineString or MultiLineString and has an array of points (coordinates) that the route goes through.

Cadastral boundaries

Another dataset includes the information about Brno cadastral zone boundaries. This is useful in situations, where the public transport data are different for each cadastral zone. As it is mentioned later in this thesis, the application uses this dataset when calculating average travel time from one stop to each cadastral zone. It has GeoJSON format with features of type Polygon, that have an array of points which form the border of a cadastral zone.

Chapter 6

Tool Implementation

6.1 Client

The client side of the application is built upon React component hierarchy with an uni-directional data flow, which serves as the building block of application's user interface. Displaying all the geospatial data is provided by react-leaflet[1] library which is a React wrapper for Leaflet.js.

6.2 Server

Server is a Node.js application built using Express.js, which has internal WebSocket server for real-time communication. As illustrated in the figure 6.1, its core responsibility is importing GTFS timetable dataset into an SQLite3 database which happens immediately after server start and is scheduled to repeat every Sunday at midnight when the dataset gets updated. It also provides endpoints for interacting with transit data, such as retrieving transfer information, stops, stop times and transit routes and calculating travel times for cadastral zones. Furthermore, it uses external R[2] and RBQL[3] libraries to perform data analysis on travel times.

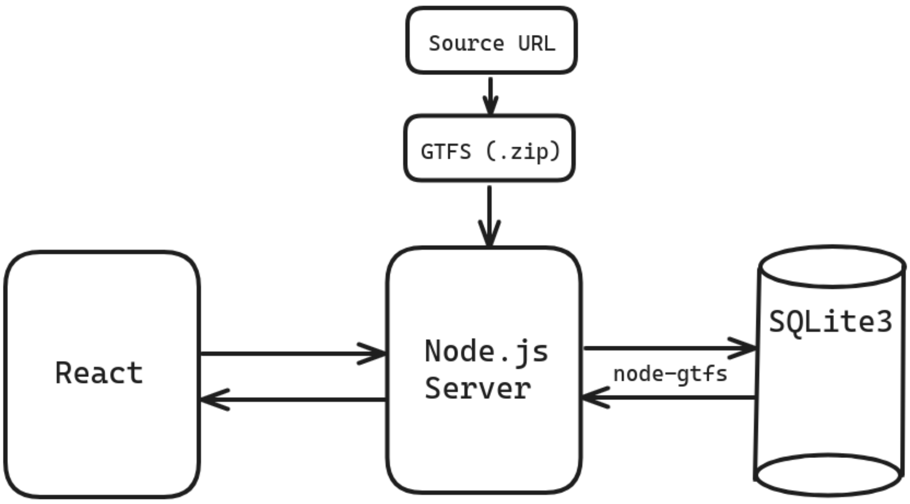


Figure 6.1: Tool architecture

6.3 Database

To create the database the server uses node-gtfs[4] library that implements downloading the GTFS feed from specified web URL and importing it into an SQLite3 in-memory database, that stores this data while the server is running and after the server connection is closed, all the data is erased. The database contains eight tables as shown in the figure 6.2 which illustrates the entity relationship diagram.

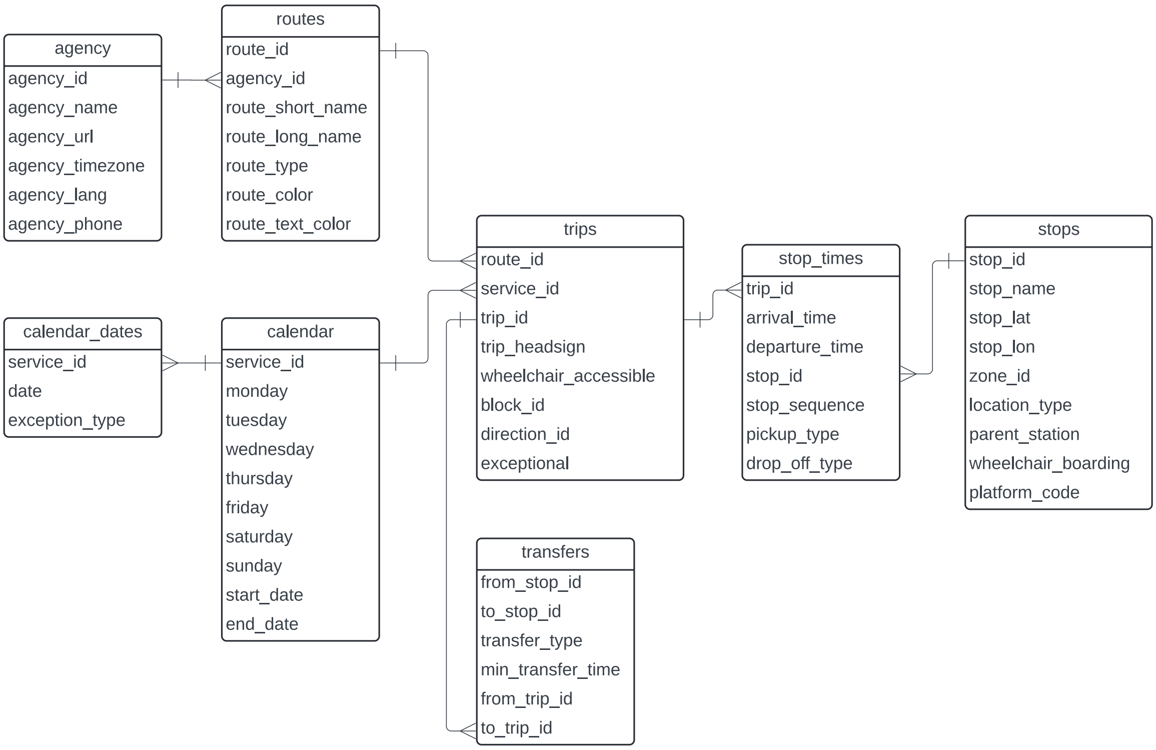


Figure 6.2: ER diagram of imported GTFS timetable data

6.4 Routes and Stops

The main data source for this feature will be the Public transit routes dataset for routes and Public transit GTFS static timetable dataset for stops.

The dataset for routes is available in many different data formats and coordinate systems. The chosen data format is GeoJSON that is using the EPSG:3857 coordinate system. This feature uses the GeoJSON component from react-leaflet library that can only be fed with geospatial data using the EPSG:4326 coordinate system, therefore it is necessary to convert the coordinate systems first before providing data to the component. To do this conversion, the server uses proj4js[5] library that transforms point coordinates from one sys-

[1] Ref.https://react-leaflet.js.org/
[2] Ref.https://www.r-project.org/about.html
[3] Ref.https://rbql.org/
[4] Ref.https://www.npmjs.com/package/gtfs
[5] Ref.http://proj4js.org/

tem to another, making it an ideal choice for solving this problem. Listings 6.1 and 6.2 demonstrate the difference between the two coordinate systems.

```
"geometry": {
  "type": "LineString",
  "coordinates": [
    [1856102.01802041,
        6297514.98589665],
    [1856850.4795963,
        6296707.53957617],
    [1856952.39161871,
        6296600.73800903],
    ...
    [1849467.69146882,
        6307039.05234927]
  ]
}
```

Listing 6.1: Route data using EPSG:3857 coordinate system

```
"geometry": {
  "type": "LineString",
  "coordinates": [
    [16.561667038437432,
        49.20217023316339],
    [16.570775529957097,
        49.19466925930823],
    [16.57193756142666,
        49.19364992546247],
    ...
    [16.491712261867863,
        49.27644107847968]
  ]
}
```

Listing 6.2: Route data using EPSG:4326 coordinate system

Table stops inside the database includes information about all the stops in South Moravian region, but the application should visualize only stops located in Brno. To filter out all stops outside Brno the database must be queried for all rows in table stops that have zone_id attribute equal to 100, 101 or 510 as these are the only transit zones inside Brno area.

As can be seen in figures 6.3 and 6.4, the routes are colour-coded by vehicle type - trams (red), trolleybuses (green), buses (blue) and the stops are colour coded by zone - 100 (maroon), 101 (purple), 510 (yellow).

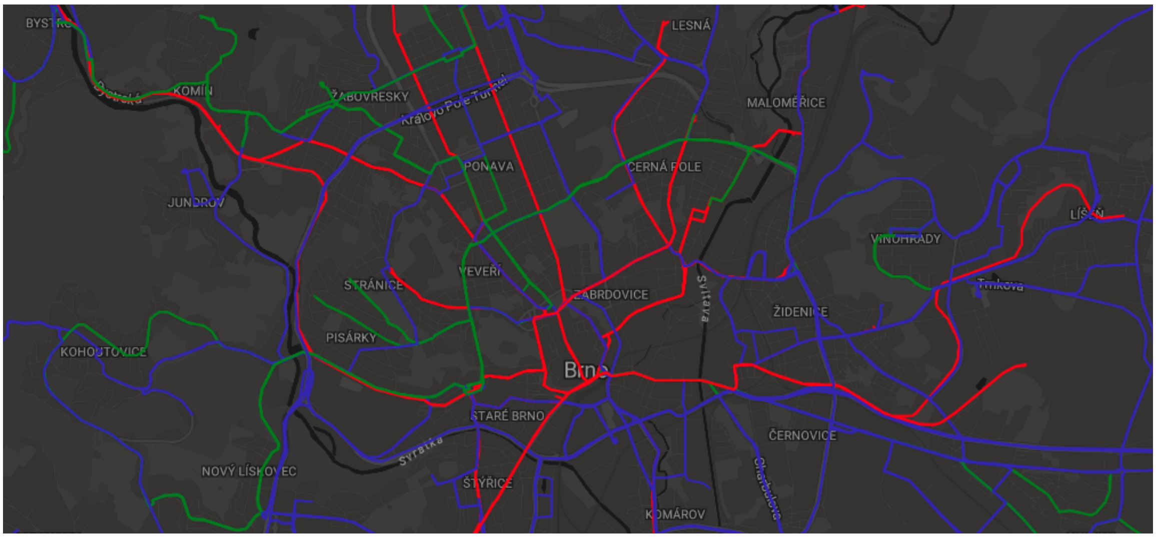


Figure 6.3: Public transit routes visualization

Figure 6.4: Public transit stops visualization

6.5 Trip frequency

To implement this feature, it is required to process data from tables stops and stop_times. The first step is joining these two tables on stop_id attribute to connect each stop time with its stop's latitude, longitude and zone_id, and filtering the table to include only stops with zone_id 100 and 101. Then it is needed to count all rows with the same value of stop_id, arrival_time and departure_time. This count represents the trip count of each stop in specific time and will be stored as trip_count attribute in resulting table.

As can be seen in the figure 6.5, table stop_times contains two records for stop_id U1436Z2 that have the same value of arrival_time, departure_time and stop_id, but each of them has unique trip_id. The same applies for stop with id U1176Z2 which occurs in three records. This means that trip count for the stop U1436Z2 (Nové sady) in time 16:38:00 is two, and for the stop U1176Z2 (Hybešova) in time 16:35:00 is three. To get the result table there needs to be performed an SQL query that aggregates these two joined tables by arrival_time, departure_time and stop_id, counts the number of records for each aggregation and stores this count as new attribute trip_count.

After performing this query, the result table records will contain all the important information about each stop, such as its name and coordinates and also its trip count for each stop_time, as illustrated in the figure 6.6.

For the visualization part, the map must show the number of trips for each stop. Since each stop is a point on the map with unique coordinates, as mentioned earlier, a heatmap is best suited for this visualization as the trip count can be represented by color intensity. Additionaly, the result table contains records for whole day, but the interesting part is being able to see how the trip frequency changes over time during the day. This is portrayed by figures 6.7 and 6.8 that illustrate the difference in trip frequency in two different time ranges.

To implement time range filtering, there has to be a slider component inside the user interface which is used to set the timeRange state variable. Changing the time range will trigger a filtering function that filters out all records with both arrival_time and departure_time outside this range.

Stop_times

trip_id	arrival_time	departure_time	stop_id
24923	16:38:00	16:38:00	U1436Z2
15221	16:38:00	16:38:00	U1436Z2
24823	16:35:00	16:35:00	U1176Z2
28123	16:35:00	16:35:00	U1176Z2
26321	16:35:00	16:35:00	U1176Z2

Stops

stop_id	stop_name	stop_lat	stop_lon	zone_id
U1436Z2	Nové sady	49.190222	16.609403	100
U1176Z2	Hybešova	49.189209	16.603364	100

Figure 6.5: Example records of tables stop_times and stops

stop_id	stop_name	stop_lat	stop_lon	arrival_time	departure_time	trip_count
U1176Z2	Hybešova	49.189209	16.603364	16:35:00	16:35:00	3
U1436Z2	Nové sady	49.190222	16.609403	16:38:00	16:38:00	2

Figure 6.6: Result data ready for visualization

To add a heatmap layer, the application uses leaflet.heat[6] library which provides a function that takes an array of points and their intensity as an input and adds a heatmap to react-leaflet map component. In this case, it is necessary to take the array of filtered records and extract values of stop_lat, stop_lon and trip_count from each record. This provides the geographical point, where stop_lat and stop_lon are its coordinates and trip_count is its intensity. Finally, an array of these points is fed into the heatmap function.

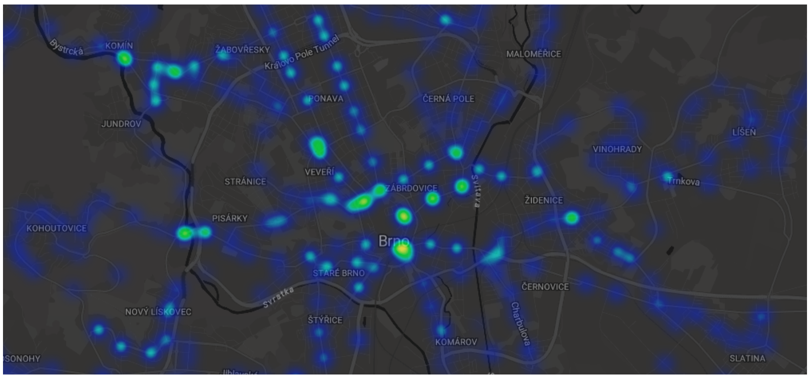


Figure 6.7: Trip frequency between 00:00 and 02:00

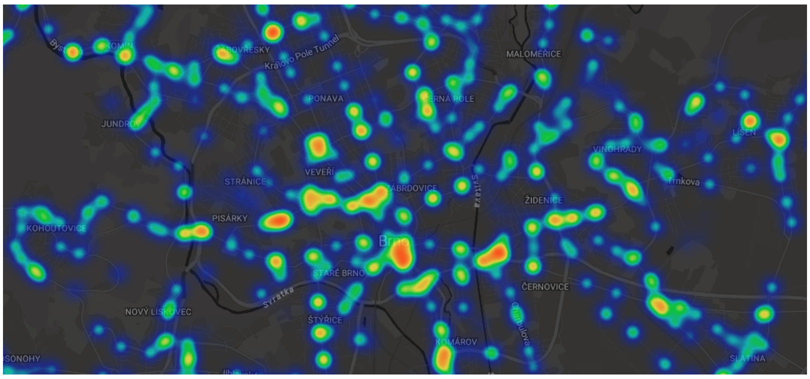


Figure 6.8: Trip frequency between 04:00 and 06:00

6.6 Travel times

This section deals with the implementation of travel times visualization. First of all, it is necessary to calculate the travel time from an initial stop to all other stops. This is taken care of by the gtfsrouter[7] library of the R language, which is able to import and process a GTFS dataset and return data containing the travel times to each destination stop. Then it is important to process GeoJSON data on cadastral boundaries, calculate the average travel time to each cadastral zone, and color-code each zone accordingly.

The start of data processing begins in feeding the function gtfs_gettraveltimes of gtfsrouter library with GTFS data and also the initial stop and time range which travel times should be calculated for. The result of this function is then written to travel_times.csv output file shown in the figure 6.9, which includes a table with travel times records. Each record in this output represents a destination stop with its geographical coordinates and the duration of travel from initial stop.

–

```
start_time,duration,ntransfers,stop_id,stop_name,stop_lon,stop_lat
15:00:00,00:01:00,0,U1436Z1,Nové sady,16.609025,49.190116
15:00:00,00:01:00,0,U1746Z1,Vlhká,16.618976,49.191847
15:04:00,00:01:00,0,U1784Z2,Zelný trh,16.609624,49.19333
15:00:00,00:01:00,0,U1436Z3,Nové sady,16.6091,49.190027
15:04:00,00:01:00,0,U1716Z1,Úzká,16.614666,49.189807
15:00:00,00:02:00,0,U1357Z3,Malinovského náměstí,16.612962,49.196072
15:04:00,00:02:00,0,U1357Z2,Malinovského náměstí,16.614794,49.195046
15:00:00,00:02:00,0,U1596Z1,Soukenická,16.60696,49.188555
15:00:00,00:02:00,0,U1637Z2,Šilingrovo náměstí,16.605234,49.192027
15:09:00,00:02:00,0,U1193Z3,Janáčkovo divadlo,16.610034,49.198165
```

Figure 6.9: Travel times records from initial stop „Hlavní nádraží"

The next step is calculating the average travel time for each cadastral zone. The solution is to load GeoJSON with the cadastral zone boundaries, traverse all the stops in the travel

[6] Ref.https://www.npmjs.com/package/leaflet.heat
[7] Ref.https://cran.r-project.org/web/packages/gtfsrouter/vignettes/gtfsrouter.html
[8] Ref.https://turfjs.org/

times records, and based on the coordinates of each stop, find out which stops belong to which cadastral zone. Then, for each zone, the median of the durations of all stops is calculated. In general, for cases like this it is better to calculate median rather than average, because each zone has a different geographical size and in large zones with big differences in duration values, an average value would give a distorted idea of the actual travel time to that zone.

To calculate the median duration, the custom function getMedianDuration is used, which takes an array of geographic points (stops) and a polygon (cadastral zone) as an input, and for each point in this array, determines whether the point is located in the cadastral zone by using the function booleanPointInPolygon from the turf[8] library, and if so, stores the duration of that stop in durations array. The median value is then calculated from this array.

If none of these stops is found within the cadastral zone, that means the zone is unreachable from the initial stop and calculating median duration is not possible. This is a mistake caused by inaccuracy of gtfs_gettraveltimes function as it did not return any records containing stops within this zone. This only happens for zones that are very far from the initial stop, so their travel time is the longest. Therefore, in this case, the function will return a constant value of 3000 (time in seconds) to represent these furthest zones.

Now, as the listing 6.3 demonstrates, it is required to call this function for each cadastral zone and assign its return value to the avgDuration property of each zone object.

```
if (Object.keys(cadastralCopy).length > 0) {
        cadastralCopy.features.forEach(feature => {
            feature['avgDuration'] = getMedianDuration(travelTimes, feature.
                geometry.coordinates);
        });
}
```

Listing 6.3: Assigning calculated travel duration to each zone

Now that each zone is assigned an average duration, each zone must be color coded according to how long its duration is using a choropleth map. The library react-leaflet provides a Choropleth component that can visualize a set of polygons and has an adjustable color scale property that gives color to each polygon based on its specified attribute. Also there has to be some way in which the user can select the initial stop and time range as the input. For this purpose there is a dropdown menu with all possible initial stops and also time range selection box included in user interface. Figures 6.11 and 6.12 show implemented choropleth map where cadastral zones that are the nearest to the initial stop have red color while the furthest are colored as blue. Also, it is possible to view the average travel time to each zone in minutes as displayed in the figure 6.10.

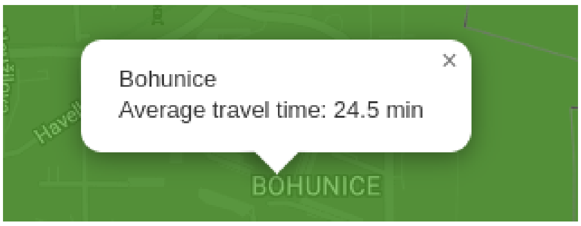


Figure 6.10: Average travel time from stop „Hlavní nádraží“ to zone „Bohunice“

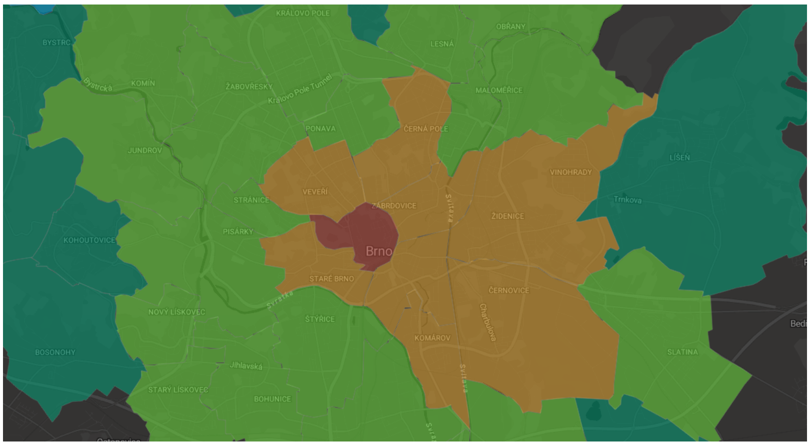


Figure 6.11: Travel durations from stop „Hlavní nádraží“

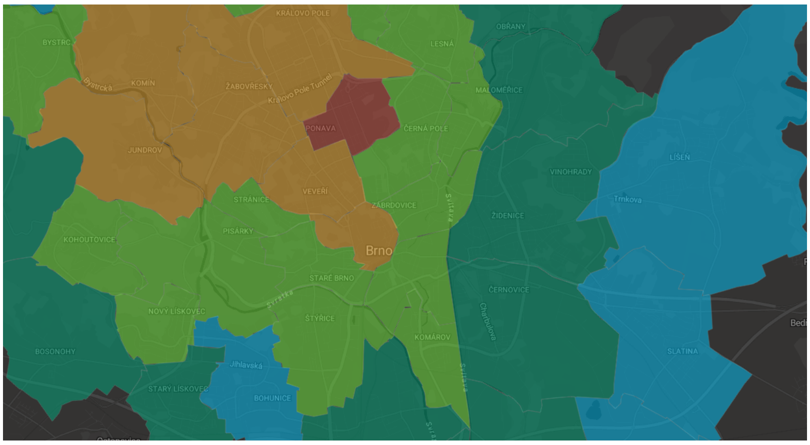


Figure 6.12: Travel durations from stop „Skácelova“

6.7 Real-Time positions

For getting public transit vehicle positions real-time data, there exists an open websocket server at gis.brno.cz. It provides live data stream of JSON objects where each object identifies a vehicle, its coordinates, delay, type and so on. This server is scheduled to send a set of updated objects every 10 seconds. The problem is, however, that each object is sent individually, but the application should be able to visualize all vehicles at once, not just one of them. To solve this, it is necessary to accumulate all objects after each update into an array and send it to the client. Another issue lies in the data format. Although, it would be possible to visualize these objects in JSON format, the better way is converting it into GeoJSON features and storing it inside a feature collection array. This will make it possible to feed these objects directly into the GeoJSON component on client side.

Data conversion

The first step is developing a function that converts each object into a GeoJSON feature. It is important to think about which attributes of the original object make sense for the visualization. It is certainly useful to know the line name and type of the vehicle. It also makes sense if the user would know the vehicle's current delay as that makes it possible to predict the arrival times and easily identify which vehicles have the longest delay. The listing 6.4 demonstrates a function that extracts these valuable attributes from an object and transforms it into previously studied and well-known GeoJSON structure.

```
    const convertToGeoJson = (data) => {
    const feature = {
        type: 'Feature',
        geometry: {
            type: 'Point',
            coordinates: [data.geometry.x, data.geometry.y]
        },
        properties: {
            id: data.attributes.id,
            name: data.attributes.linename,
            vtype: data.attributes.vtype,
            delay: data.attributes.delay
        }
    }
    return feature;
}
```

Listing 6.4: Function converting data into GeoJSON

Websocket communication

The next step is establishing websocket communication. On server side there is an internal websocket server that is subscribed to the external data stream via websocket URL. It then listens to these before-mentioned 10 seconds „heartbeats". After receiving a message, each object is converted into a GeoJSON feature and pushed into the feature collection. Finally, there is a setInterval function with a callback that executes periodically in 12 seconds intervals (after the feature collection has been filled with the data) and sends the feature collection to the client.

On client side, there must be implemented a receival of this data stream. As the listing 6.5 shows, this is solved by using a React hook useWebSocket that ensures data subscription to specified websocket URL, in this case, the URL of internal websocket server running locally. After each message is received from the server, the feature collection gets stored inside a state variable which triggers a component re-render.

```
const socketUrl = 'ws://localhost:3001';
const [geoJson, setGeoJson] = useState({});
const { sendJsonMessage, lastJsonMessage, readyState } = useWebSocket(
    socketUrl);
useEffect(() => {
    if (lastJsonMessage) {
        setGeoJson(lastJsonMessage);
    }

}, [lastJsonMessage]);
```

Listing 6.5: Client part of websocket connection

Visualization

After establishing a live data stream from server to client, it is time to take this feature collection that the client is receiving and put it on the map. This can be simply done by feeding the feature collection inside a GeoJSON component from react-leaflet the exact same way as described in routes and stops visualization, but this time features use the correct coordinate system so there is no need to deal with conversion. Every time the client receives an updated feature collection, the GeoJSON component immediately re-renders so the map always shows the latest update. Additionaly, each feature contains a property with vehicle's current delay in minutes, which provides an option to color-code vehicles according to it by passing a function to pointToLayer property of GeoJSON component.

Figures 6.13 and 6.14 portray the visualization, in which each vehicle is assigned a color based on its current delay where the green color means no delay, orange represents one minute delay and red colored vehicles are delayed by two minutes or longer.

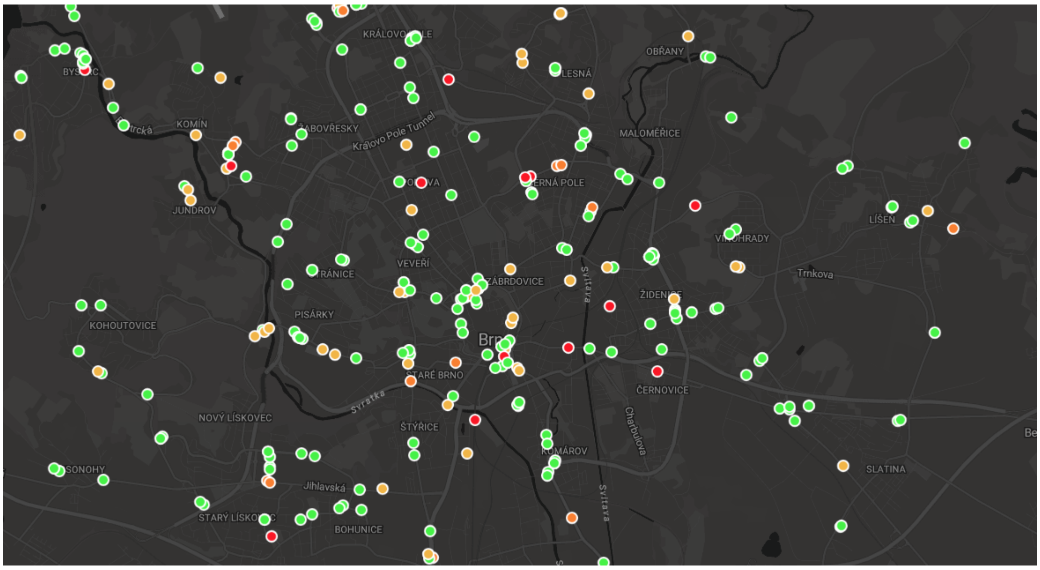


Figure 6.13: Real-time positions of vehicles during daytime

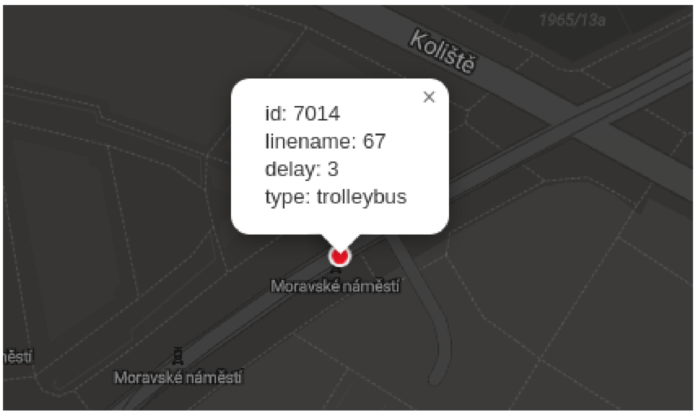


Figure 6.14: Location of trolleybus 67 at stop „Moravské náměstí“

6.8 Transfers

The final part of the tool is the visualization of transfers. It is based on similar principle as visualization of trip frequency, the only difference lies in data processing. In this case, data from tables transfers, stops and stop_times of imported GTFS feed must be processed.

At first, table transfers is linked with stop_times on both starting and destination trip_id and stop_id to get the arrival and departure times of that transfer. Secondly, it must also be joined with table stops to get the latitude and longitude of initial and destination stop.

Stop_times

trip_id	arrival_time	departure_time	stop_id
27021	21:44:00	21:55:00	U1429Z1
4497	21:46:00	21:46:00	U1429Z6
26959	22:59:00	23:03:00	U1429Z1
4453	23:01:00	23:01:00	U1429Z6
26882	22:44:00	23:00:00	U1429Z1
4685	22:49:00	22:49:00	U1429Z7
16826	22:49:00	22:49:00	U1429Z7

Stops

stop_id	stop_name	stop_lat	stop_lon	zone_id
U1429Z1	Nemocnice Bohunice	49.175849	16.567334	101
U1429Z6	Nemocnice Bohunice	49.175872	16.566969	101
U1429Z7	Nemocnice Bohunice	49.176176	16.567045	101

Transfers

from_stop_id	to_stop_id	from_trip_id	to_trip_id
U1429Z1	U1429Z6	27021	4497
U1429Z1	U1429Z6	26959	4453
U1429Z1	U1429Z7	26882	4685
U1429Z1	U1429Z7	26882	16826

Figure 6.15: Tables needed for transfers visualization

After linking the tables shown in the figure 6.15 together, the result table includes four records, each of them represented by a tuple that includes IDs and coordinates of both stops of the transfer, the arrival time to initial stop and departure time from destination stop. Figures 6.16 and 6.17 illustrate these four records of the result table.

from_arrival_time	from_stop_id	from_stop_lat	from_stop_lon
22:59:00	U1429Z1	49.175849	16.567334
21:44:00	U1429Z1	49.175849	16.567334
22:44:00	U1429Z1	49.175849	16.567334
22:44:00	U1429Z1	49.175849	16.567334

Figure 6.16: Initial stop of transfer

to_departure_time	to_stop_id	to_stop_lat	to_stop_lon
23:01:00	U1429Z6	49.175872	16.566969
21:46:00	U1429Z6	49.175872	16.566969
22:49:00	U1429Z7	49.176176	16.567045
22:49:00	U1429Z7	49.176176	16.567045

Figure 6.17: Destination stop of transfer

Now it is possible to count all unique transfers for each pair of stops by aggregating all records with the same value of from_stop_id and to_stop_id as each unique transfer is identified by these two attributes. This count is stored as attribute number_of_transfers in result table. Since the number of transfers is different for each stop, this is location based data so again a heatmap is useful for visualization. However, in this situation, each transfer is identified by two points (stops) rather than just single point, but heatmap component requires one point for each transfer. A suitable solution to this is calculating coordinates of a point located exactly in the middle between these two stops. This can be done by adding latitudes and logitudes of initial and destination stop and dividing it by two.

Finally, as can be seen in the listing 6.6, the result data involves all necessary information for the heatmap component, where latitude and longitude attributes represent the coordinates of calculated point and number_of_transfers represents the color intensity.

```
{
    from_arrival_time: '22:44:00', from_stop_id: 'U1429Z1',
    latitude: 49.1760125, longitude: 16.5671895,
    to_departure_time: '22:49:00', to_stop_id: 'U1429Z7',
    number_of_transfers: 2
}
```

Listing 6.6: Object structure of result data

As can be seen in figures 6.18 and 6.19, during the night hours the most transfers are in the surrounding areas of „Komín“, „Bohunice“ and „Líšeň“ and also around the stop „Anthropos“. From 6:00 am onwards, the number of transfers starts to increase in the city centre.

Figure 6.18: Transfers between 00:00 and 02:00

Figure 6.19: Transfers between 06:00 and 08:00

Chapter 7

Testing

Data reusability

As covered in chapter 6, several features of the tool use a data source in GTFS format containing all information about public transport timetables in Brno. This test experiment will investigate the reusability of the GTFS dataset. That means, whether the implemented tool is able to adjust to modifying the source GTFS dataset containing information about another city by visualizing its public transport data.

The experiment can be applied to two features of the tool - visualizing the frequency of trips and also transfers, because these are the only features that are completely based on GTFS data. The dataset selected for testing the data reusability is provided by The Berlin-Brandenburg Transport Association (VBB)[1]that regularly provides up-to-date transport timetable data from Berlin and Brandenburg in GTFS format. For testing, it is necessary to simply replace the source URL in the node-gtfs library configuration file to import this data into the database when the server starts.

As shown in the figure 7.1 the application is able to adapt to changes in the input data providing visualization of trip frequency in Berlin.

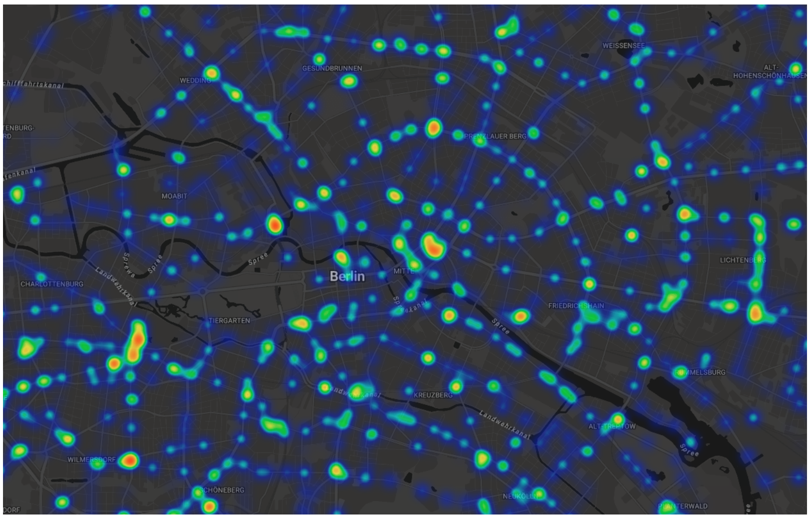


Figure 7.1: Trip frequency of Berlin between 01:00 - 05:00

Functionality

Part of the testing is to verify that the tool meets the functional requirements of the users. The primary target group is users who are involved in Brno public transport data analysis and need a visualisation tool.

Functionality testing is thus an experiment in which the tool was provided to a sample of three relevant users and their feedback was used to reveal the shortcomings of the tool and various information that may be helpful for designing future improvements.

Each user was given a set of three tasks to complete:

- Finding the stops with the highest and lowest connection frequencies between 08:30 and 08:45
- Selecting the stop „ÚAN Zvonařka“ and finding the cadastral zone with the highest and lowest average travel time between 14:00 and 16:00
- Displaying real-time visualization and identifying vehicles that are currently delayed by three minutes or longer

The feedback from users has been quite positive, but nevertheless all of these users had a problem with the selection of time range for the average travel time. The user interface contained a numeric input into which it was possible to type the time interval, but the problem was that when one boundary of the interval was changed, a request for data was immediately sent to the server and therefore there was no waiting for the user to change the other boundary. As illustrated by figure 7.2, this was quite a serious problem and had to be solved immediately by adding a button that sends the request to the server after the user has entered all the necessary inputs.

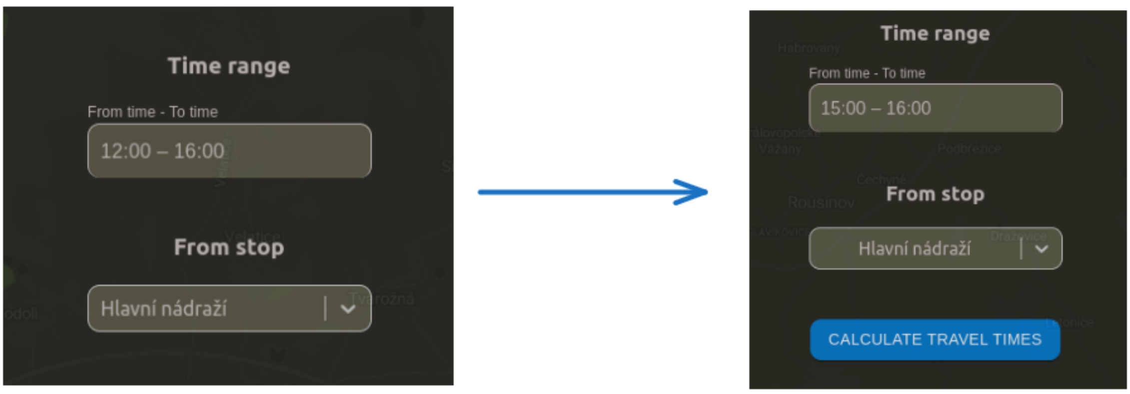


Figure 7.2: Travel times input GUI update

[1] Ref.https://daten.berlin.de/datensaetze/vbb-fahrplandaten-gtfs

Chapter 8

Conclusion

The goal of this thesis was developing a tool for dynamic visualization of public transport data in Brno. First, it was necessary to get familiar with the visualization of geospatial data in a broad context and to explore modern methods and tools. Then it was needed to study the concept of using big data in the context of public transport and to explore current solutions that are based on data from publicly available sources. The following step included investigation of the currently used tools for traffic data visualisation in Brno and finding possible improvements. Based on this knowledge, it was then possible to design the architecture and core features of the tool to meet user requirements.

The result is a tool that is able to visualize dynamic changes of interesting data about Brno's public transport elements based on user input. This tool can be used for the analysis of connection and transfer frequency, travel durations or the real-time location and delay of public transport vehicles. The tool is capable of updating transport data from official sources on a scheduled basis. The development of the tool has proved a great practical usability of GTFS data and the tool is able to easily adapt to any changes of the input data in this standardised format. Conversely, a limiting element of the tool is, for example, the source of real-time positional data, as it is a little bit distorted from the real world data due to a ten second server delay.

There are quite wide possibilities for future extensions of the tool. For example, the tool could be extended to visualise the exact number of transfers between specific stops or the time needed by passengers to take the transfer. Additionaly, there could be a feature that visualizes the maximum wait time between two consecutive connections in different city zones. The tool also provides possibilities for extensions in the area of real-time data, e.g. vehicle markers could show the direction of travel or the next stop.

Although, there are currently larger, more comprehensive applications developed by teams with greater resources, this tool focuses on things that these applications lack, such as playing a timelapse over data that changes over time or adding more interactivity to the visualization of travel times in terms of user experience. Overall, the tool could be used in practice by public transport planners allowing them to make informed decisions in a dynamic urban environment.

Bibliography

[1] (GIS) Geographic Information System. *National Geographic Education* [online]. [cit. 2024-05-04]. Available at: https://education.nationalgeographic.org/resource/geographic-information-system-gis.

[2] Visualizing Geospatial Data. *SafeGraph* [online]. [cit. 2024-05-04]. Available at: https://www.safegraph.com/guides/visualizing-geospatial-data.

[3] Explore Brno's Public Transport System. *ArcGIS* [online]. 13. june 2018 [cit. 2024-05-04]. Available at: https://www.arcgis.com/apps/MapJournal/index.html?appid=6ce3a916606e4c1996015613ef57e016.

[4] Barbeau, S. J. Quality control-lessons learned from the deployment and evaluation of GTFS-realtime feeds. In: *97th Annual Meeting of the Transportation Research Board, Washington, DC.* 2018 [cit. 2024-05-04].

[5] Bennett, J. *OpenStreetMap*. Birmingham, UK: Packt Publishing Ltd, 2010 [cit. 2024-05-04]. ISBN 978-1-847197-50-4.

[6] Breunig, M., Bradley, P. E., Jahn, M., Kuper, P., Mazroob, N. et al. Geospatial Data Management Research: Progress and Future Directions. *ISPRS International Journal of Geo-Information*. MDPI AG. february 2020, vol. 9, no. 2, p. 95, [cit. 2024-05-04]. DOI: 10.3390/ijgi9020095. ISSN 2220-9964. Available at: http://dx.doi.org/10.3390/ijgi9020095.

[7] Bulková, K. *Visualization of data.Brno datasets [online]*. 2023 [cit. 2024-05-04]. Available at: https://is.muni.cz/th/mkier/.

[8] Gill, J. K. Geospatial Visualization Tools and its Techniques. *XenonStack* [online]. 31. august 2022 [cit. 2024-05-04]. Available at: https://www.xenonstack.com/insights/what-is-geospatial-visualization.

[9] Hrubỳ, M. Geografické informační systémy (GIS). *Studijní opora, FIT VUT, Brno.* 2006, [cit. 2024-05-04].

[10] Hynek, J., Kachlík, J. and Rusňák, V. Geovisto: A Toolkit for Generic Geospatial Data Visualization. In: *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. SCITEPRESS - Science and Technology Publications, 2021 [cit. 2024-05-04]. DOI: 10.5220/0010260401010111. Available at: http://dx.doi.org/10.5220/0010260401010111.

[11] Juviler, J. Google Maps API: How to Get Started. *HubSpot Blog* [online]. 14. september 2022 [cit. 2024-05-04]. Available at: https://blog.hubspot.com/website/google-maps-api.

[12] Kraak, M.-J. and Brown, A. *Web Cartography*. 1st ed. CRC Press, 2001 [cit. 2024-05-04]. Available at: https://doi.org/10.1201/9781482289237.

[13] Kraak, M.-J. and Ormeling, F. Cartography at Work. In: *Cartography*. CRC Press, July 2020, p. 225–238 [cit. 2024-05-04]. DOI: 10.1201/9780429464195-11. ISBN 9780429464195. Available at: http://dx.doi.org/10.1201/9780429464195-11.

[14] Kujala, R., Weckström, C., Darst, R. K., Mladenović, M. N. and Saramäki, J. A collection of public transport network data sets for 25 cities. *Scientific Data*. Springer Science and Business Media LLC. may 2018, vol. 5, no. 1, [cit. 2024-05-04]. DOI: 10.1038/sdata.2018.89. ISSN 2052-4463. Available at: http://dx.doi.org/10.1038/sdata.2018.89.

[15] Kunama, N., Worapan, M., Phithakkitnukoon, S. and Demissie, M. GTFS-Viz: tool for preprocessing and visualizing GTFS data. In: *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*. ACM, September 2017 [cit. 2024-05-04]. UbiComp '17. DOI: 10.1145/3123024.3124415. Available at: http://dx.doi.org/10.1145/3123024.3124415.

[16] Lazúr, J. *Webová aplikácia pre tvorbu máp hromadnej dopravy*. Brno, CZ, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Available at: https://www.fit.vut.cz/study/thesis/23889/.

[17] Mishevska, E., Najdenov, B., Jovanovik, M. and Trajanov, D. Open Public Transport Data in Macedonia. In: *Proceedings of the 11th Conference for Informatics and Information Technology*. 2014, p. 161–166 [cit. 2024-05-04].

[18] Nedvědová, K. and Gregorová, E. Jaký druh přepravy v Brně převažuje? *Data.Brno* [online]. 20. january 2023 [cit. 2024-05-04]. Available at: https://data.brno.cz/pages/clanek-jaky-druh-dopravy-v-brne-prevazuje.

[19] Nöllenburg, M. Geographic Visualization. In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, p. 257–294 [cit. 2024-05-04]. DOI: 10.1007/978-3-540-71949-6_6. ISBN 9783540719489. Available at: http://dx.doi.org/10.1007/978-3-540-71949-6_6.

[20] Prommaharaj, P., Phithakkitnukoon, S., Demissie, M. G., Kattan, L. and Ratti, C. Visualizing public transit system operation with GTFS data: A case study of Calgary, Canada. *Heliyon*. Elsevier BV. april 2020, vol. 6, no. 4, p. e03729, [cit. 2024-05-04]. DOI: 10.1016/j.heliyon.2020.e03729. ISSN 2405-8440. Available at: http://dx.doi.org/10.1016/j.heliyon.2020.e03729.

[21] Stock, K. and Guesgen, H. Geospatial Reasoning With Open Data. In: *Automating Open Source Intelligence*. Elsevier, 2016, p. 171–204 [cit. 2024-05-04]. DOI: 10.1016/b978-0-12-802916-9.00010-5. ISBN 9780128029169. Available at: http://dx.doi.org/10.1016/b978-0-12-802916-9.00010-5.

[22] Torre-Bastida, A. I., Del Ser, J., Laña, I., Ilardia, M., Bilbao, M. N. et al. Big Data for transportation and mobility: recent advances, trends and challenges. *IET Intelligent Transport Systems*. Institution of Engineering and Technology (IET). july 2018, vol. 12, no. 8, p. 742–755, [cit. 2024-05-04]. DOI: 10.1049/iet-its.2018.5188. ISSN 1751-9578. Available at: http://dx.doi.org/10.1049/iet-its.2018.5188.

[23] Wikipedia contributors. *GeoJSON — Wikipedia, The Free Encyclopedia* [https://en.wikipedia.org/w/index.php?title=GeoJSON&oldid=1210694214]. 2024. [Online; accessed 3-May-2024].

[24] Wu, J., Du, B., Gong, Z., Wu, Q., Shen, J. et al. A GTFS data acquisition and processing framework and its application to train delay prediction. *International Journal of Transportation Science and Technology*. Elsevier BV. march 2023, vol. 12, no. 1, p. 201–216, [cit. 2024-05-04]. DOI: 10.1016/j.ijtst.2022.01.005. ISSN 2046-0430. Available at: http://dx.doi.org/10.1016/j.ijtst.2022.01.005.

[25] Zhao, K. *Network Analysis of Urban Traffic with Big Bus Data*. 2016 [cit. 2024-05-04].