Universidade de Aveiro 2017
Departamento de Electrónica, Telecomunicações e Informática,

Jorge Miguel Antunes Pereira

Dashboard para Apoio à Decisão na Análise de Tráfego e Ambiente de uma Cidade Inteligente

Decision Support Dashboard for Traffic and Environment Analysis of a Smart City

Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática,
2017

Jorge Miguel Antunes Pereira

Dashboard para Apoio à Decisão na Análise de Tráfego e Ambiente de uma Cidade Inteligente

Decision Support Dashboard for Traffic and Environment Analysis of a Smart City

"Believe you can and you're halfway there."

— Theodore Roosevelt

Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática,
2017

Jorge Miguel Antunes Pereira

Dashboard para Apoio à Decisão na Análise de Tráfego e Ambiente de uma Cidade Inteligente

Decision Support Dashboard for Traffic and Environment Analysis of a Smart City

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Informática, realizada sob a orientação científica da Professora Doutora Susana Sargento, Professora Associada com Agregação do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro e co-orientação científica do Professor Doutor José Maria Fernandes, Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

o júri / the jury

presidente / president	**Professor Doutor Paulo Miguel de Jesus Dias** Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro)
vogais / examiners committee	**Professor Doutor José Maria Fernandes** Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro
	Professor Doutor Rolando da Silva Martins Professor Auxiliar Convidado do Departamento de Ciência de Computadores da Faculdade de Ciências da Universidade do Porto

agradecimentos / acknowledgements

Gostaria de agradecer em primeiro lugar aos meus pais pelo apoio e incentivo que me deram ao longo destes anos e que sem eles não chegaria a esta fase. Às minhas irmãs por estarem presentes e enriquecerem os momentos mais positivos e à minha avó pela sua enorme força de vontade e incentivo transmitido. Um muito obrigado a todos.

Queria agradecer à minha namorada, Ana Luísa por todo o carinho e apoio que me têm dado durante o meu percurso académico, e por estar ao meu lado no melhores e piores momentos.

Gostaria de agradecer aos meus amigos e colegas de faculdade Fábio Alves, Rafael Martins, João Verdasca, Eduardo Martins, Fernando Duarte, José Duarte, Francisco Melo, Alvaro Soveral, Luís Carvalho e Miguel Tomaz pelo apoio e companheirismo ao longo deste percurso académico.
Gostava também de agradecer a todos os meus colegas do grupo de investigação do NAP pelo excelence espirito de equipa e pela amizade que me oferecerem. Queria agradecer particularmente aos meus colegas mais próximos Bruno e Leandro que me ajudaram tanto técnicamente como pessoalmente durante o desenvolvimento desta dissertação.

Gostaria de agradecer à Professora Susana Sargento pela sua orientação e o seu apoio mostrado durante o desenvolvimento da tese de mestrado, acreditando sempre nas minhas capacidades e incentivando sempre a ir mais além. Ao professor José Maria Fernandes pela sua co-orientação, apoio e pelas suas infindáveis ideias, que às vezes fora do normal mostraram-se muitas das vezes interessantes e úteis para este trabalho.

Resumo

As cidades têm estado continuamente a crescer tanto em população, como em veículos, infra-estruturas e inteligência. Ao implementar e usar tecnologias inteligentes na infra-estrutura das cidades, é possível melhorar as diversas áreas de uma cidade, como a mobilidade ao melhorar a infra-estrutura das estradas, as infra-estruturas ao melhorar o planeamento urbano e a população ao disponibilizar melhores serviços.

A cidade do Porto tem neste momento uma infra-estrutura de sensores fixos e móveis em mais de 400 autocarros, e unidades de comunicação na estrada, com GPS e sensores de mobilidade nos elementos móveis, e com sensores ambientais nas unidades fixas. Esta infra-estrutura proporciona dados valiosos baseados nos padrões de mobilidade dos autocarros. Os dados dos sensores ambientais são também disponibilizados e usados para analisar a qualidade do ar da cidade e a sua influência perante o tráfego de veículos.

O sistema desenvolvido fornece uma integração completa da informação num dashboard da cidade que mostra e correlaciona os dados gerados pelo movimento dos autocarros e do ambiente a partir dos sensores fixos, permitindo diferentes visualizações do trânsito nas estradas e do ambiente na cidade, e decisões sobre o estado atual da cidade. Um bom exemplo é a relação da variação da velocidade dos autocarros com possíveis anomalias na estrada ou engarrafamentos. Ao visualizar esta informação com um nível de detalhe superior nas anomalias encontradas na estrada, o gestor da cidade pode beneficiar do dashboard quando precisa de tomar decisões relacionadas com o planeamento urbano e assim melhorar de uma maneira inteligente a mobilidade da cidade.

Abstract

Cities are continuously growing in population, vehicles, infrastructures and intelligence. Using and deploying smart technologies in the cities infrastructure can improve the multiple existing areas of a city, such as mobility by improving the road network, infrastructure by improving the urban planning and population by contributing with better services.

Porto city has an in-place infrastructure of fixed and moving sensors in more than 400 buses and roadside units, with both GPS and mobility sensors in moving elements, and with environmental sensors in fixed units. This infrastructure can provide valuable data that can extract information to better understand the city and, eventually, support actions to improve the city mobility, urban planning, and environment.

This work has the objective of using the information generated by the sensors placed in the buses of Porto, and using it to analyze the road traffic information based on the mobility patterns of the buses. The data from the environmental sensors deployed in Porto is also provided and used to analyze the air quality of the city and its influence by the traffic.

The developed system provides a full stack integration of the information into a city dashboard that displays and correlates the data generated from the buses movement and the environment from the fixed sensors, allowing different visualizations over the road traffic and the environment in the city, and decisions over the current status of the city. A good example is the relation of bus speed variation with possible anomalies on the road or traffic jams. Visualizing such information with a superior level of detail on the road traffic, more anomalies can be found, adding more value to a city manager when taking urban planning decisions to improve the city mobility in a smart way.

Contents

Contents **i**

List of Figures **v**

List of Tables **vii**

Acronyms **ix**

1 Introduction **1**
- 1.1 Motivation . . . 1
- 1.2 Objectives and Contributions . . . 2
- 1.3 Document Structure . . . 2

2 State of the art **5**
- 2.1 Introduction . . . 5
- 2.2 Porto . . . 5
- 2.3 Smart Cities . . . 9
  - 2.3.1 City Big Data . . . 12
  - 2.3.2 Mobility on decision support systems . . . 12
  - 2.3.3 City wide dashboards . . . 13
- 2.4 Data Warehouse . . . 19
  - 2.4.1 Extract, Transform and Load . . . 19
  - 2.4.2 Star schema . . . 20
- 2.5 GIS - Geographic Information System . . . 21
- 2.6 Related Work on Mobility and Environment Analysis . . . 27
- 2.7 Discussion . . . 29
- 2.8 Summary . . . 30

3 Data analysis **31**
- 3.1 Introduction . . . 31
- 3.2 Vehicle Data processing . . . 31
  - 3.2.1 Vehicle Position and Time Instant Fetch . . . 32
  - 3.2.2 Position to Road Snap Operations . . . 33
  - 3.2.3 Calculation of Vehicle and Road Variables . . . 36

3.2.4 Persistence of Information . . . 38
3.3 Environment Data processing . . . 41
3.3.1 Environment Metrics Fetch . . . 41
3.3.2 Parsing and Preparation of Data . . . 42
3.3.3 Persistence of information . . . 42
3.4 Building the Traffic Profile . . . 43
3.5 Traffic and Environment Correlation . . . 44
3.6 Road Cleaning and Transformation . . . 45
3.7 Summary . . . 47

4 The System **49**
4.1 Introduction . . . 49
4.2 The architecture . . . 49
4.3 Data Information Services . . . 52
4.4 The System Web based Dashboard . . . 53
4.4.1 Mapping of roads and locations . . . 53
4.4.2 Charts Representation of Vehicle and Sensors Variables . . . 54
4.4.3 Traffic Profile Comparison . . . 55
4.4.4 Representing environment information . . . 55
4.4.5 Road Definition and Transformation . . . 56
4.4.6 Road Data Management View . . . 57
4.5 Summary . . . 60

5 Implementation **61**
5.1 Introduction . . . 61
5.2 Data Fetching . . . 61
5.3 Data Analysis . . . 62
5.4 Databases and Data-warehouse . . . 62
5.4.1 Structure . . . 62
5.4.2 Functions . . . 64
5.5 Data Visualization . . . 67
5.6 Summary . . . 68

6 Case Studies **71**
6.1 Introduction . . . 71
6.2 Traffic in different hours of weekday . . . 71
6.3 Discovering a Critical Zone . . . 72
6.4 A special event in Porto . . . 75
6.5 How the road traffic affects the environment . . . 77
6.6 Summary . . . 79

7 Conclusions and Future Work **81**
7.1 Conclusions 81
7.2 Future Work 81

Bibliography **83**

List of Figures

2.1 VANET network in a smart city. 6
2.2 The different layers in a smart city network [7]. 11
2.3 Dublin mobility dashboard [13]. 15
2.4 London smart city dashboard [14]. 16
2.5 Boston city dashboard [15]. 17
2.6 Moncloa city dashboard [16]. 18
2.7 Sydney city dashboard [18]. 18
2.8 A generic structure of a database star schema. 21
2.9 Ellipsoid and geoid comparison [35]. 24
2.10 Database Spatial Operations. 27
2.11 Traffic mining approach in TrailMarker work [42]. 28

3.1 Vehicle Position gathering to a database server and fetching cycle for data analysis of position metrics. 32
3.2 Vehicle trajectory on multiple phases on different roads. The light grey roads represent unknown roads, and the dark grey roads represent the ones available on the database. 34
3.3 Comparison of the vehicle position road match without a buffer versus with a implemented road buffer on the same geometry road. 35
3.4 The Haversine Formula [48]. 37
3.5 Four positions from a unique bus trajectory during its trip represented in two different roads. t = time, d1,d2 = distance, start = fist position in the road, last = last position in the road. 38
3.6 Representation of the process of Gathering environment information from sensors, the transmission of the metrics to the Fiware server and the request of environment information from the local server. 41
3.7 Distance transformation approach. Each one of these points represent a vehicle position on the road, with a known distance between them that is accumulated to know the real distance of each point on the road. 43

3.8 Example of the road speed profile with an aggregation of points from 3 different vehicles through the distance of the street. The metrics represented in the figure are from one point and represents the speed of 33 km/h from the vehicle 2811 at the distance 400 meters of the street, which is also the 0.5 fraction of the road, in the middle. 44
3.9 Approach used when choosing roads to correlate the speed information with environment information. Here is chosen the roads that are closer to the sensor that provides the environment metrics. 45
3.10 Example of the Avenida Carlos Oliveira Campos street initially composed by four different segments, being cleaned and transformed into one segment. 47

4.1 Proposed system flow architecture. 50
4.2 Interactions of database components and its intrinsic support. 51
4.3 Global road traffic. 54
4.4 Traffic profile view of the Rua de Costa Cabral street in the morning of the weekdays. 55
4.5 Inputs Selection for Different Searches. 56
4.6 Traffic profile comparison of Avenida da Boavista street from different time instants. 57
4.7 Visualization of multiple Environment Sensors in Porto city and the CO vs speed correlation chart. 58
4.8 Dashboard interface to select and transform Road Segments. 58
4.9 Dashboard interface for giving instructions and getting the status of the analysis of road information. 59

5.1 Data warehouse structure. 65
5.2 Transformation of date-time to dimension table structure of data warehouse. 66
5.3 Main visualization components from the dashboard implementation. 68

6.1 Rua da Constituição street in Porto spatial representation and its speed profile comparison of mid-day and end-day hours of a typical week. 72
6.2 The Porto street of Avenida da Boavista with a speed profile during a typical week, in the morning, between 8 and 9 o'clock. 74
6.3 Speed profile of a section of the street of Avenida da Boavista in Porto during a typical week in the morning, between 8 and 9 o'clock. 74
6.4 Street view of the intersection of Avenida Antunes Guimarães with Avenida da Boavista road. 75
6.5 Area of Red Bull Air race event in Porto with the selected Avenida da República road to analyze. 76
6.6 Avenida da República road traffic profile comparison in two different time instants, one of the events days and the other in the weekend after the event. 77
6.7 Correlation of traffic speed vs CO values through the day from the environment sensor deployed in Casa da Musica, Porto. 78

List of Tables

2.1 Description of the vehicular information available from the bus mobility network of Porto city. . . . 7
2.2 Description of the environment information available from the deployed sensors in Porto city. . . . 9
2.3 Comparison of features from different smart city dashboards. NO2 = Nitrogen dioxide; SO2 = Sulfur dioxide; PM = Particulate Matter; O3 = ozone. 14
2.4 Comparison of a list of geographic information system applications. 23
2.5 List of OpenStreeMap road information attributes. 25

3.1 Available information used for the data analysis. 32
3.2 Example of the allocation of values to the attributes of each point in the four steps from the figure 3.5. 40

Acronyms

Ajax Asynchronous JavaScript and XML.

AP Access Point.

API Application Programming Interface.

BT Bluetooth.

CO Carbon Monoxide.

CPU Central Processing Unit.

CSS Cascading Style Sheets.

CSV Comma-Separated Values.

GIS Geographic Information System.

GPRS General Packet Ratio Service.

GPS Global Positioning System.

GSM Global System for Mobile Communications.

HTML HyperText Markup Language.

HTTP Hypertext Transfer Protocol.

IEEE Institute of Electrical and Electronics Engineers.

IoT Internet of Things.

JDBC Java Database Connectivity.

JSON JavaScript Object Notation.

LTE Long-Term Evolution.

LTS Long Term Support.

OBU On-Board Unit.

ODBC Open Database Connectivity.

OSM Open Street Map.

REST Representational State Transfer.

RSU Road-Side Unit.

SQL Structured Query Language.

VANET Vehicular Ad-hoc Network.

WAVE Wireless Access in Vehicular Environments.

WGS World Geodetic System.

WIFI Wireless Local Area Networking.

WLAN Wireless Local Area Network.

WSN Wireless Sensor Network.

XML eXtensible Markup Language.

Chapter 1

Introduction

1.1 Motivation

Nowadays it is more and more demanding being connected everywhere, from an individual standpoint using smart-phones or similar devices, and also from a city point of view where different infrastructures need to be connected using sensors and communication devices.

A smart city has a vision to integrate and evolve different areas such as environment and transportation in a way that it can optimize their resources, plan its preventive maintenance activities through monitoring systems and smart sensors to collect data in real time and improve the city management decision making [1].

The cities are growing every day and with that, comes an increasing on the vehicles that circulate through its streets. As the city grows, the traffic congestion increases and becomes more important to manage the road infrastructure. Optimizing road traffic is necessary but demanding, so using a different and better approach from the traditional human-centric methods to the road management of the city is ideal.

As making changes in the road infrastructure of the city is very expensive, their construction or modifications need to be very well studied and their outcomes need to be predictable. Using smart technologies in the city to complement a decision making support system will help with a certain level of accuracy to identify problems that often cannot be detected by the human eye. The solution of building or transforming the city infrastructure can be less expensive than what is currently available, with data and a decision support system. In theory, it is possible to perform more informed and conscientious decisions regarding city traffic network, namely from traffic light timing to changes in network through roundabouts or road traffic direction changes.

Adding sensors to vehicles that circulate through the city roads allows to grab useful information from the city's mobility, that can be processed and analyzed to aid the management of the city roads and the decision making when changing or implementing a road infrastructure.

This dissertation will make use of this information to develop a decision support system to understand and improve the city management.

1.2 Objectives and Contributions

The main objective of this work is to provide an end to end solution to monitor changes on road traffic from information obtained from more than 400 buses and environment sensors available in Porto city. To reach that goal, the following objectives are taken into account:

- Extract traffic related data from buses and fixed sensors.
- Select and define the main statistics and indicators for traffic and extract them.
- Relate the traffic information with environment information.
- Provide city wide dashboards for different visualizations over the traffic and environment performance in the city over time, namely compare the same road in different times of day/week/month.

Given the size and specificity of the traffic related data, there are some challenges that can already be identified:

- To map the bus information into spatial map of the city, namely handling GPS jitters and map GPS positions into actual road location.
- To summarize the bus information - options like data warehouse to support a decision support system to obtain traffic estimations for the city roads based on the bus dynamics around the city.
- To transform the data into information that can be used to support decisions on the mobility of the city.
- To represent the bus traffic information in a comprehensible way, using a user friendly interface.

Following this study development, one article has been submitted, with the title "Decision Support Dashboard for Traffic and Environment Analysis of a Smart City", to VEHITS 2018, in October 30th and it is waiting for approval.

1.3 Document Structure

This document is organized as follows:

- **Chapter 1** contains the motivation, context and objectives of the work.

- **Chapter 2** presents the state of the art, which contains concepts of smart cities, data warehouse, spatial information and data visualization.
- **Chapter 3** contains the logic and analysis of the vehicular data.
- **Chapter 4** presents the overall architecture of the proposed solution, as well the developed dashboard and the integration services.
- **Chapter 5** presents in more detail the processes used in the implementation of the system.
- **Chapter 6** shows the results obtained from the implemented solution using some case studies.
- **Chapter 7** presents the main conclusions of this work and suggestions for future work.

Chapter 2

State of the art

2.1 Introduction

In order to better understand the scope of this work and the reasoning behind several options it is essential to understand the context of Smart cities, more specifically the case of Porto. This includes understanding the information that is available and common strategies to handle the data either at a conceptual integration level and at a technical level (e.g. handling the information streams). During the extent of this chapter it will be presented and contextualized the current status of the Porto city where it is described the implemented smart vehicle network in the city and the available data sources; the general concepts of a smart city, the city layers, the transportation and big data insight with some examples of existing dashboards; the concepts of data warehouse used in this work, the processes and generic structures; the Geographic Information Systems presents the multiple geographic systems and the related concepts.

In the end of the chapter it will also be presented related work that shares similar objectives, or similar types of data sources that are based on vehicular and environment information.

2.2 Porto

The current work will focus on Porto city and it will depart from its existing IT / smart city infrastructure.

Porto is one of the cities that is implementing in its infrastructures smart solutions to improve city's dynamic and intelligence in a wide variety of fields, from the environment to the mobility where distinct data of the city is captured using an in place base of fixed and moving sensors in more than 400 buses and road side units with environment sensors.

Vehicular Information

The smart mobility platform of the city is built around the public transport bus service, where a vehicular ad-hoc network through on-board units (OBU) are implemented in the buses to provide free WIFI access for the users of the bus transportation service. It is also deployed in the city roadside units (RSU) in multiple locations to provide a hotspot for the bus OBU to connect and exchange big amounts of information, as the example in figure 2.1.

The Porto's vehicular network also serves as a gateway of the information generated from the environment sensors deployed in different locations. After capturing the environment variables, it uses different hops from the network to reach their end destination to then be analyzed.

All these infrastructures provide continuous data from different areas: from the mobility side, the information captured from the buses can provide data from their users, status of the bus and their real time position, and network information. From the information captured from the environmental sensors, it is provided weather, pollution and environment information. All these sources of information generate a huge amount of information, since there is a high density of vehicles with more than 400 buses circulating in a wide area around the city with a continuously availability of information, generating data every 15 seconds for each bus.

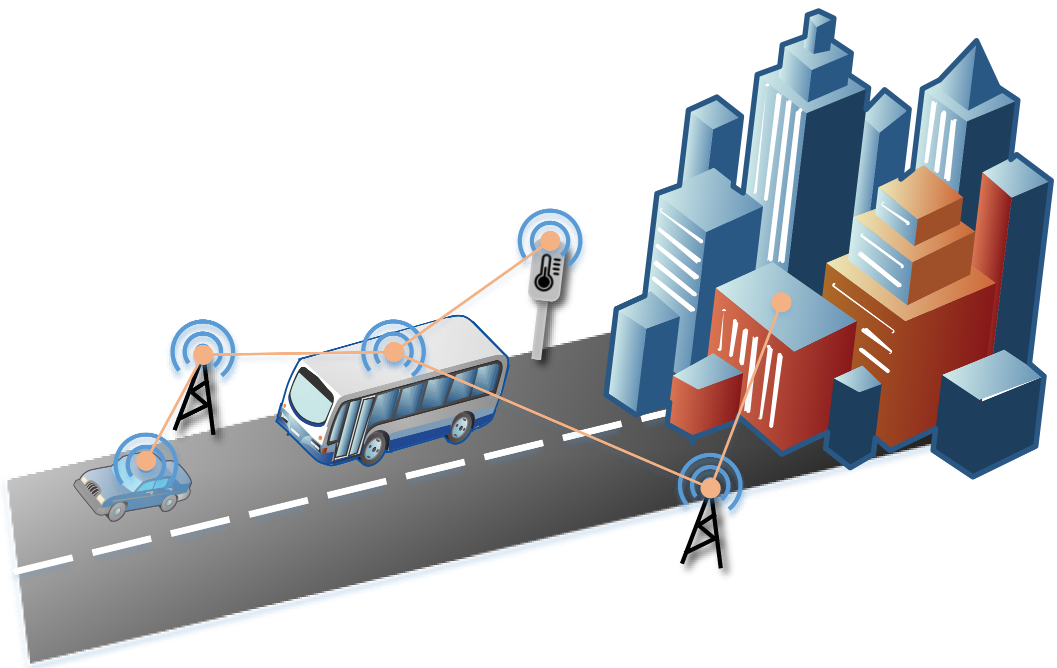


Figure 2.1: VANET network in a smart city.

All the generated data from the vehicular network gives new possibilities of using this data to build different applications for the transportation systems namely for improvement of the services related with the public transportation, and also for the enhancement of the road infrastructure for the population that lives in the city.

The vehicular data available from the captured variables from the buses around the city contains many important information to study. These variables are divided by the ones that provide the *timestamps* and others that provide the position and location of the vehicle, all during multiple captures in their trip. The following table provides detail of the existing attributes from this data-set of vehicular data.

attribute	designation	example
node id	unique identifier of the vehicle	2529
system time	date and time attribute of the hardware in the vehicle	2017-07-18 00:25:10
gps time	date and time instant of the gps in the vehicle	2017-07-18 00:25:10
server time	date and time instant of when the data reaches the server	2017-07-18 00:25:10
latitude	the latitude of the vehicle position in decimal degrees	41.23123221
longitude	the longitude of the vehicle position in decimal degrees	-8.13121312
altitude	the height of the vehicle above sea level in meters	96
speed	the instant speed of the vehicle at a certain time	35
heading	heading/direction of the vehicle in decimal degrees	58

Table 2.1: Description of the vehicular information available from the bus mobility network of Porto city.

Although there are many variables available from the vehicular network as shown in table 2.1, not all are necessary to make the necessary statistics and analysis on the road. Here the ones that are being used are the ”node id”, the ”gps time”, the ”latitude” , the ”longitude” and ”speed”. There are several reasons to choose these 5 variables, and some of these reasons are the following:

- `node id`: the unique identifier of the vehicle is used to make individual statistics for each vehicle. With that addition, it will be possible to select and query a specific vehicle and make a traffic analysis for that vehicle.

- `gps time`: In order to get a date and time reference associated with the data, the gps time is the variable used. Although there are other time variables, the gps time is the one associated with the captured time of the gps coordinates.

- `latitude`: the latitude of the vehicle is one of the essential attributes, so it gives the position of the vehicle to make the base of the road statistics.

- `longitude`: like the latitude of the vehicle, the longitude serves to give the position of the vehicle.

- `speed`: the instant speed of the vehicle is a complement of data used to know the speed at a certain coordinate. It is used to add precision to the traffic analysis.

These attributes provide the essential to work with geo-localization of vehicles, which is what is necessary to study the speed and traffic on the roads in different instants and events. This vehicular data is from about 400 different vehicles and has a good time granularity as it adds a new log entry every 15 seconds for each vehicle, which means that if all vehicles are active at the same time, there are 24000 logs/minute.

Environmental information

Multiple environment sensors deployed in the Porto city provide valuable data about the environment conditions in the areas that are nearby the sensors. In the captured data, it is possible to obtain information about the weather and the air or noise pollution values of the city.

The environment information is used in this work in conjunction with the vehicular data to add a new layer of information on the mobility of the city.

All the environment sensors information have a range of attributes available, but not all the environment variables are contextualized in this work, so only the used ones will be focused and are the following:

- `id`: this unique identifier is needed to associate the data to a specific sensor. With this information, an anomaly in the sensor can be detected and solved.

- `latitude` and `longitude`: the position of the sensors are an important metric, due to the fact that it gives environment values based on a location, so that can be related with the vehicles position.

- `time`: it is necessary to know when a certain environment value occurs.

- `temperature`: this variable is useful to provide the temperature values throughout the day.

- `noise`: the noise levels allow us to study how the road traffic affects the noise pollution in the city.

- `co`: the carbon monoxide is important to measure the air quality in the city and how the traffic affects it.

attribute	designation	example
id	unique identifier of the sensor	213
latitude	the latitude of the position of the sensor in decimal degrees	41.2312
longitude	the longitude of the position of the sensor in decimal degrees	-8.3232
time	the instant date and time when the data was captured	2017-07-18 00:25:10
temperature	the captured temperature represented in degrees Celsius	25
noise	the noise levels at a certain moment in decibels	71
co	the carbon monoxide value at a given time in volts	2.12

Table 2.2: Description of the environment information available from the deployed sensors in Porto city.

2.3 Smart Cities

Smart cities is a recent adopted concept in the 21st century to the development of information systems in the cities, although the definition of smart cities may be varied for different persons, Hollands describes different meanings to best define it in [2]. One that he expresses is that smart cities are the application of electronic and digital applications in cities; the use of information technology to transform the way of living in a region; the variety of embedded Information and Communication Technologies deployed in the city; the spatial territories that bring information and communications technologies and people to improve innovation, learning and knowledge to the city. It is a way to integrate urban infrastructures and services in large cities in different areas such as buildings, transportation, electrical and water distribution, and public safety, contributing for the planning, development, and operation of the cities [3].

As the global population continues to grow and the population that lives in the cities increases, new ways are needed to follow this growing demand that comes with the increase of vehicles and the overload of public services, in a way to continuously provide quality of life of the general population that lives in the cities. It is here where the use of smart cities term has emerged as a urban based technological innovation to add a new perception of the variables of the city, integrating sensors and infrastructures with intelligent networks, simply making it smart, digital and innovative. This will enable to introduce new challenges and transformations on the management and behaviour of the city people, mobility and infrastructure.

As figure 2.2 shows, a smart city network is composed by different layers: the population, the transportation, the information and the energy. These layers will be detailed in the following text.

Population

The citizens are the ones that can make the city smarter because the they are the main actors in the daily movement of the city. A smart city should also focus on the higher education of the individuals, so more skilled workers will be available. The more knowledge, the more value and creativity people can bring to the city and be more attractive for smart people impulsing the urban development [4].

The urbanization or growth of population in the cities also comes with positive aspects as the workforce will increase, the taxation base will expand, and the consumption on the local business. On the downside, this also adds some challenges: the pressure on the public services and resources to continue with the population growth rate, the way to control bigger crowds, the scaling of public transportation and management of waste. This could be an opportunity for integrating IoT devices for better management of services and resources of the city to help make the cities and populations smarter [5].

The implementation of smart technologies will benefit the population in different areas: in education it will improve quality and access, in health care it will lead to faster and accurate diagnosis, in transportation by offering a faster and convenient way to move using public transports and in public safety to respond to emergencies and threats in real-time [6].

Transportation

One of the main service presented in the smart cities is the transportation service, where the optimization on the use of buses, metro, trains, garbage collectors and emergency services can be examples of what a smart city can provide. Focusing on the bus transportation system, this is a service that has a connection or interactions with the multiple layers of a city, from the people that uses it, to the necessary fuel resources to move the vehicles, the bus stops and road infrastructure to move on.

The bus transportation service can be used in a smart way to better understand the mobility of the city. Because buses are one of the few types of vehicles that move in a wide area of the city every day, they are also the ones that can give us precious information of the traffic in different locations and different times. By adding sensors in the buses, we can analyze their mobility and use this information to detect traffic or accidents in the city, improve the road network infrastructure in the city, making the transportation service moving faster and providing new types of information to the people.

One of the challenges of the bus transportation is to improve its mobility in the city, moving faster and efficiently and arrive on time at the bus stops. All these challenges have an impact on the way that city moves, because there will be less traffic, people will move faster and arrive on time at their works and the population will be more satisfied and will

Figure 2.2: The different layers in a smart city network [7].

use more the service. This will also have an impact on the environment, as less fuel will be used and CO2 emissions will decrease.

Information

Information is the base for the city innovation and decision support when improving the overall management, services and dynamics in the city. In a smart city this information can be obtained from different sources, the city elements, the assembled infrastructure of sensors in different areas of the city and the information from the citizens digital footprint and social media [8]. All this daily activity information from assets like buses, building, environment can be collected and interpreted to be used as key performance indicators to report incidents, sending event alerts, measuring city changes over time and real-time monitoring for anomalies detection.

Energy

Another challenge of an expanding city is to supply utilities that meet growing demands. Using a smart infrastructure for energy and water will improve its efficiency to produce as much energy required while reducing waste and consequently costs. It is important to have a good management in resources as they are limited, so using sensors to supervise and manage the production and distribution of resources will help to prevent losses. Also giving people smart energy monitors can reduce the city consumption [6].

Although all these layers may seem to be independent, most of them have connections and dependencies between them. As an example, the people uses services like electricity, or water that came from the energy resources, uses public transportation to move around the city, and uses information from smart devices.

2.3.1 City Big Data

One sentence that better describes the advantages of big data in a city is the following: "Data-driven decisions are better decisions—it's as simple as that. Using big data enables managers to decide on the basis of evidence rather than intuition. For that reason it has the potential to revolutionize management" [9].

Data is what characterizes a smart city, from the data where problems can be detected and analyzed but also can help to perform more intelligent / cost effective decisions at a city level, by combining more efficiently the available information and support decision system. The cities have been producing data from census, surveys or city audits to analyze the city variables. These techniques have multiple problems and limitations: they are often generated on a non-continuous basis, focusing on a specific time and space, low variables count, limited to access, and expensive to generate and analyze [10].

A difference that big data brings is the volume of information which is very big. Gigabytes of data are captured in smart city from different sources continuously every day, more than ever before. Data from multiple sensors deployed in the city, vehicles or people generates a great variety of information from GPS signals from vehicles, environment information from deployed sensors and city infrastructures status and normal functioning, leading to new discoveries when using in conjunction from different sources, as an example using environment and traffic information to analyze their relationship. All this information can be stored with a fast speed, making it possible to be analyzed in real time, reducing costs and taking faster actions.

2.3.2 Mobility on decision support systems

Decision support systems are interactive computer systems for use by managers to access to data and giving them access to analytic models, in a way to improve the decision process using the suitable technology for the problem [11].

This area can be extended to other subjects: the mobility of a city will benefit of a decision support system to take wise choices on the actions over the city mobility and infrastructure.

Using a decision support system on a vehicular data approach, there are some requirements that can be taken into account:

- Get a specific type of vehicle - categorization of data based on the type of vehicle.

- Compare transformations over time - decision based on the comparison of different instances of time.
- Analyze vehicle data with multiple granularity's of time - analyzing information with different aggregations of time.
- Segment information for different roads - segmentation of information based on the road.
- Summarized information for a day or week - summarizing the information for different granularities.
- Information from big amounts of data - structuring and indexing of data.

All these requirements will be the base to build the necessary infrastructure for the provisioning of mobility information to be analyzed and provided through the decision support systems in this document.

2.3.3 City wide dashboards

Many cities are starting to make changes in their intelligence in different areas by adding sensors and providing information through dashboards. In table 2.3 it is compared the functionality and features presented in different dashboards currently available in several cities.

Figure 2.3 contains an example of a mobility dashboard of the city of **Dublin** where the traffic can be seen in a global view in the main roads, but also parking lot locations using sensors and traffic cameras. Often the data is sourced through annual reports and in the case of the real-time data, it is readable through an API [12]. Our proposed work can extend the information presented by Dublin dashboard by analyzing the roads with greater detail to improve anomaly detection, with information in all streets.

Figure 2.4 presents a dashboard of **London** where many city variables from points of interest, public transportation (metro status), environment (weather and pollution data) and cameras (street view) are viewed in real-time. Although there is a lot of real time information, this is only being captured and presented for a general consumer, lacking on a more detailed analysis on traffic congestion and pollution data.

Another example is the **Boston** city dashboard in figure 2.5 where there is information about the trains stops arrival time estimations in real time, a general indicator about the weather, the real time train locations and directions and bike stations. All these information is obtained from video cameras and sensors in different locations of the city. The dashboard is mainly focused in public transportation, and it could have other indicators on the city mobility as the road traffic.

The **Moncloa** district from Madrid provides a dashboard in figure 2.6 focused on WIFI connections from the local population and environment indicators. In the WIFI area it displays multiple metrics from the connection times, number of connections and their

city	traffic	public transport	weather	pollution	map	map overlays
London	no	metro status	wind, temperature, humidity, precipitation	O3, NO2, SO2, PM	yes	points of interest
Dublin	yes	bus, metro, air mobile apps	wind, temperature, precipitation	noise, O3, NO2	yes	traffic density, parking lots
Boston	no	bus, Train	temperature	no	yes	bike, train stations
Moncloa	no	no	temperature, humidity, light	noise, CO, NO2	yes	wifi heamap
Sydney	no	train, ferry, bus	wind, temperature, humidity, precipitation	O3, NO2, SO2, PM	yes	real time transportation

Table 2.3: Comparison of features from different smart city dashboards. NO2 = Nitrogen dioxide; SO2 = Sulfur dioxide; PM = Particulate Matter; O3 = ozone.

location with a heat-map. On the environment side, it has different charts of weather and pollution metrics within a customized time range obtained from outdoor environmental sensors. It lacks information about the mobility and public transportation of the city.

Sydney city dashboard in figure 2.7 provides information about the environment and public transportation. From the different views we can identify real time information from weather and pollution indicators and also the status of public transportation on a map overlay with their positions in real time. For the public transportation information display, it receives data from the transportation API that uses the Google GTFS [1] (General Transit Feed Specification) technology [17]. It also shows real time traffic cameras but it lacks on a more analytic traffic approach.

[1]https://developers.google.com/transit/gtfs/

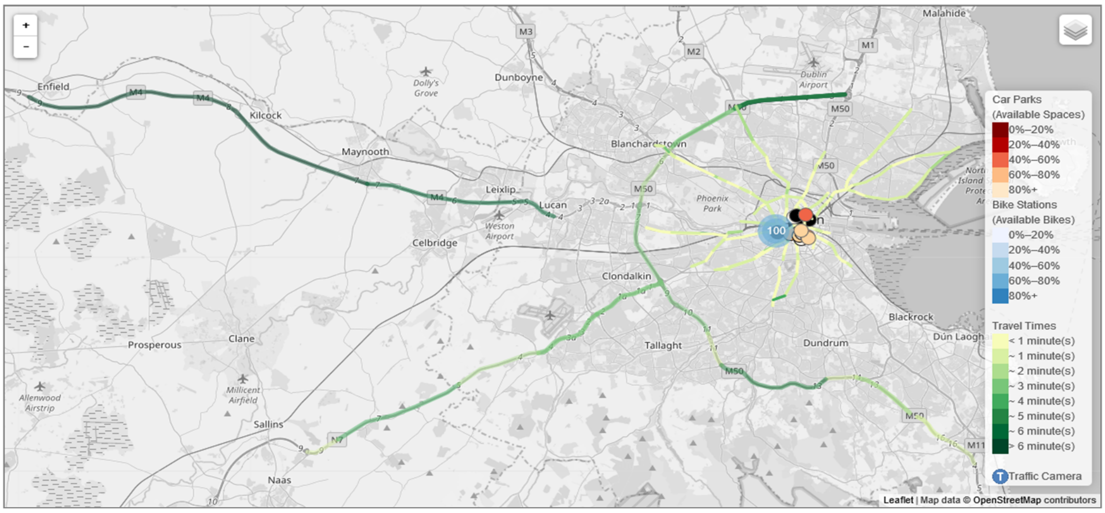


Figure 2.3: Dublin mobility dashboard [13].

Smart City Developments

The city of Barcelona [8] is an example where smart initiatives are implemented in the city infrastructure from different areas in order to deliver better public services. It uses the fibre optical network, Wi-Fi mesh network, sensors network and public Wi-Fi network to provide intercommunication for the city transportation, education and healthcare services. The main sources of information come from the city sensors, Open Data, and information from the citizens digital footprint. The Open Data project aims also to give public access of government information for the society to use it and create more value for the city.

The Helsinki [19] smart city is an example of the development of the city services and Open Data to all with interest in developing new products and business opportunities through available API. It aims for a mobile application cluster and uses a Living Lab as a platform for idealization and product development of different applications, institutions and services, being the orchestrator that provides support and feedback for collaborative projects. In the public transportation network composed by buses, trains, trams, metro and boats, it monitors these vehicles with equipped GPS locators to collect their location and providing it for the Open Data applications.

In Oulu city [20], in Finland, outdoor smart sensor nodes are a part of a wireless network infrastructure in the urban space. The infrastructure is made of: a WLAN network with approximately 1200 APs (Access Points) for public access; a Bluetooth (BT) network composed by a cluster of Bluetooth APs for free BT access for context-aware services; a cluster of WSN (Wireless Sensor Network) APs across Oulu downtown for a free connectivity of low power WSN sensors; large public displays on the streets for displaying city information.

The project from the University of Toronto [21] presents a connected vehicle infrastructure using intelligent transportation systems network architecture for the road vehicle mobility and a stable cluster of vehicles made of cars and trucks with Home agents

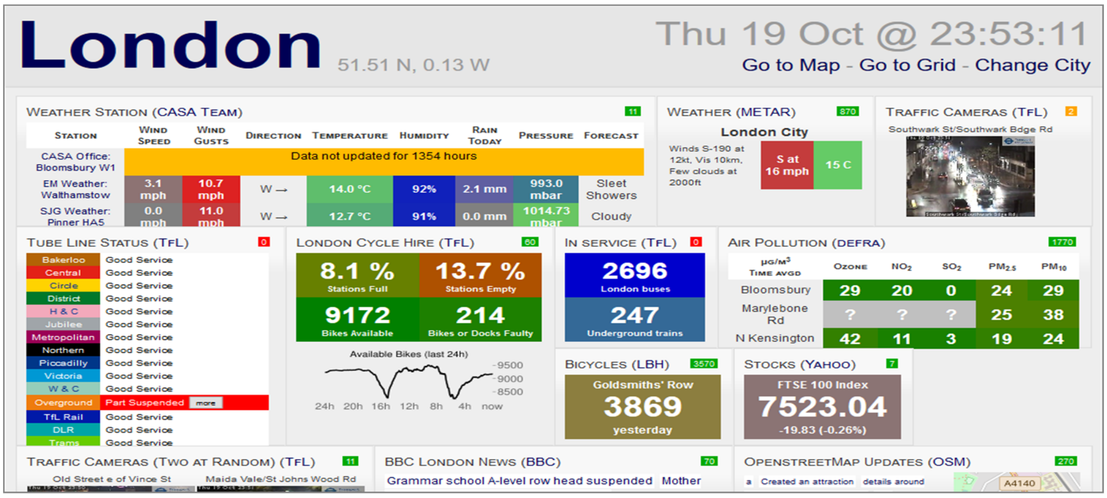


Figure 2.4: London smart city dashboard [14].

and multiple servers in the fixed network. This architecture offers Vehicle-to-Vehicle and Vehicle-to-Infrastructure communication, using GSM/LTE and IEEE 802.11. It has the objective of high network connectivity for large cluster of vehicles for a capable aggregation of the vehicular data.

The city of Santander [22] explores IoT devices and solutions deployed in several urban scenarios around the city to be aggregated into one testbed. The different scenarios include: environment sensors at fixed locations that provide measurements of environment pollution with alarms generated by the system; ferromagnetic wireless parking sensors buried under the asphalt for the occupancy monitoring of the outdoor parking spaces, its usage and accounting; sensor nodes installed on buses uses the General Packet Ratio Service (GPRS) for network communications to transmit air pollution, position and speed of the vehicles in a way to provide mobility patterns around the city; mobile phone sensing using application for user event report in the city and to exploit the users position, direction, noise and temperature.

The testbed presented for the city of Cambridge [23] consists of deployed sensor nodes with WIFI antennas for network connectivity deployed on buildings and street lights, and gateway nodes to provide internet access to the wireless network. The sensor nodes have the objective of data collection and real time monitoring using multiple environment sensors and IEEE 802.11 radios for wireless communication.

A work with the association of the city of Melbourne [24] was the noise mapping case study containing a fixed and mobile infrastructure on public spaces. This work uses a Wireless Sensor Network (WSN) with fixed sensor nodes deployed in street lights or buildings that continuously monitor the sound levels in a given area, mobile sensors deployed on vehicles that serve as relay nodes for the data buffering and transportation from the sensor nodes through multiple hops to the gateway, that sends the noise measurements via the internet to the cloud.

A vehicular infrastructure deployed in the city of Madison [25] uses cellular and WIFI

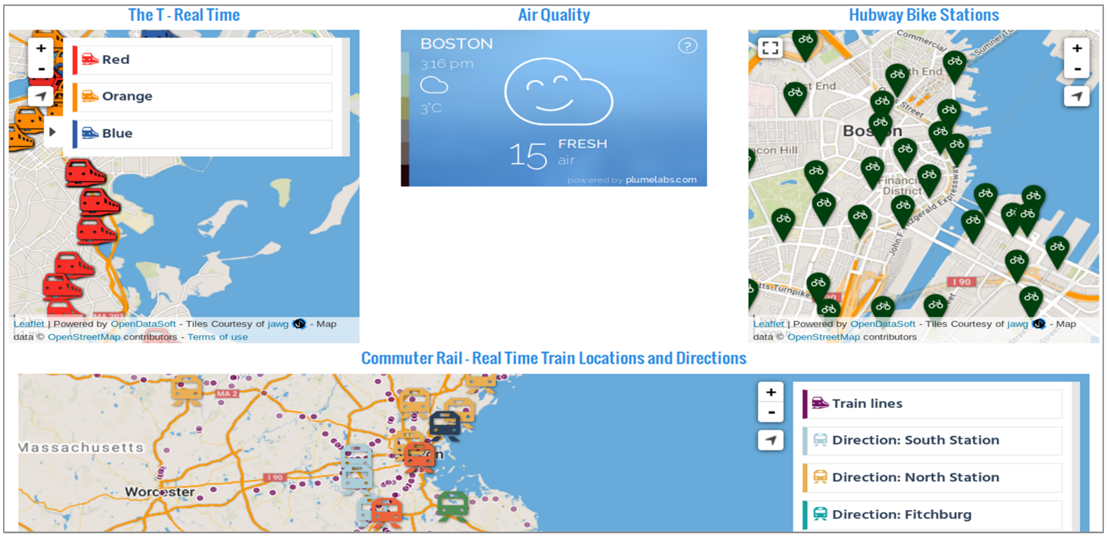


Figure 2.5: Boston city dashboard [15].

---

interfaces on units assembled in the city buses. The testbed is composed by wireless nodes in the vehicles that are equipped with a GPS receiver and measures the vehicle position and network performance such as wireless signal strength, network latency and throughput speeds, and a software platform that provides a way of monitoring and make analysis on the performance characteristics of the network. The final objective is to associate the network data measurements with the vehicle position in a way to quantify the impact of mobility on network performance.

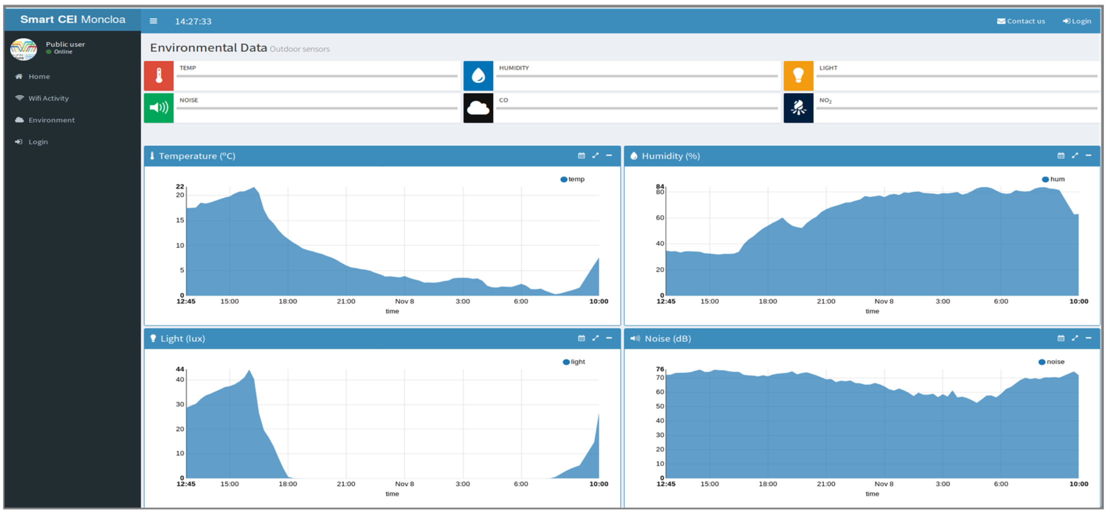


Figure 2.6: Moncloa city dashboard [16].

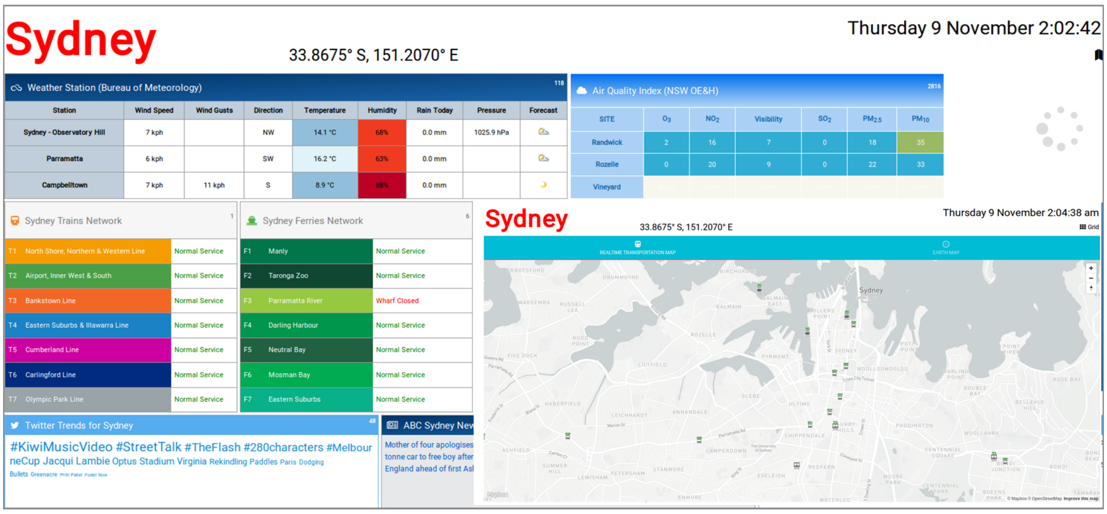


Figure 2.7: Sydney city dashboard [18].

2.4 Data Warehouse

In smart cities the big amount of information that comes from different sources to be processed and analyzed needs a solution like a data warehouse to support the decision support systems and improve the decision making in the city.

"The modern way to build systems is to separate the operational from the informational or analytical processing and data" [26]. The traditional databases systems are focused on operational and transitional operations that, for their operational purpose suits well, but for informational or analytic needs it is necessary another approach, the data warehouses. The analytics of informational processing are designed for the improvement of decision making in a organization looking at different vistas of data to detect trends.

By moving from an operational database where the data is in bulk to a data warehouse, we can fill most of these requirements because the data will be more granular, time variant, subject-oriented and have some summaries. Also the data can be corrected by removing errors or imprecision present in data, to be reusable, and, restructured by segmenting by subject and time and indexed to be accessed faster and in a efficient way.

The ETL (Extract, Transform and load) process is a commonly used procedure in data warehouse systems for the data manipulation before being stored in the data warehouse.

2.4.1 Extract, Transform and Load

Before reaching the data warehouse, the data passes through ETL (Extract, Transform and load) steps, where the errors can be removed, the missing data corrected and can be structured and analyzed from multiple sources for the data warehouse and end user tool requirements [27]: it is the transformation of raw data into information where it is important to improve the quality of data.

Extract

Extract is the first step where the data can be retrieved from different data sources or platforms, using different formats and characteristics, for example having vehicular data from a relational database and environment data from a web service in JSON format.

For the extracting part it is needed to be aware of the different drivers and connection protocols to request data and know the data structure and their different formats. For example, it could be necessary to use JDBC drivers to connect to databases or a HTTP requester API to request data from a web service.

There are two phases on the extraction process [28]: the initial extraction is when a big amount of data is extracted from different sources for the first time to populate the data warehouse, and the incremental extraction is when the data warehouse is updated or added with new data since the last extraction. As an example, the initial extraction can be when the data warehouse is constructed, but we already have vehicular information in a data source of 6 months, so that big amount of data is extracted to populate the data warehouse for the first time. After that, while the vehicles move, their data can be periodically extracted to update the data warehouse with more recent information.

Transform

The next step in the ETL cycle is the transformation of data. Normally, in this step it is made the cleaning and correctness of the data to make it more consistent and complete. Other transformations and calculations can be made in this process to add new layers of dimensions. In this process it is also made the integration with the data warehouse schema, where the data needs to be sliced in the defined granularities of the fact and dimension tables, and prepared for the structure in each table, making the necessary relationships.

Load

Lastly the data is loaded to the multidimensional data warehouse, where the prepared data from the previous extraction and transformation steps is written in the data warehouse or overwritten to update existing information into both fact and dimension tables.

2.4.2 Star schema

In dimensional modeling, the simpler form of block structure is the star schema which consists of one fact table that has large amounts of data, and the smaller dimension tables that link to the central fact table. In figure 2.8 it is represented a generic structure of the star schema that may have any number of dimensions [29]. Normally the fact table supplies numeric values, while the dimension tables supply labels or descriptions of those values.

A fact table is the primary table in the dimensional model where the performance measurement values are stored, as the term fact represents a business measure [30]. We can imagine a moving vehicle retrieving measurements, intersecting different dimensions of space, time and vehicle, so each measure will add a new row in the fact table. In the fact table there are two or more foreign keys that connect to the primary keys of dimension tables. For example, a vehicle ID in the fact table will correspond to the ID in the vehicle dimension table, and the same applies to other dimensions, expressing many-to-many relationships between dimensions.

The Dimension Tables are the complement of information of the fact tables, containing as many as possible attributes that describe the rows of the fact table [30]. Normally the dimension tables have a relatively low number of rows, but have a large number of columns. It is normally used the dimension tables for the constraints of the query; for example, getting a measurement of the vehicles with the brand Ford will first see the brand in the dimension table vehicle to then mach the ID in the fact table.

As referenced by Kimball, "Dimension tables are the entry points into the fact table. Robust dimension attributes deliver robust analytic slicing and dicing capabilities. The dimensions implement the user interface to the data warehouse." [30]. The better the quality

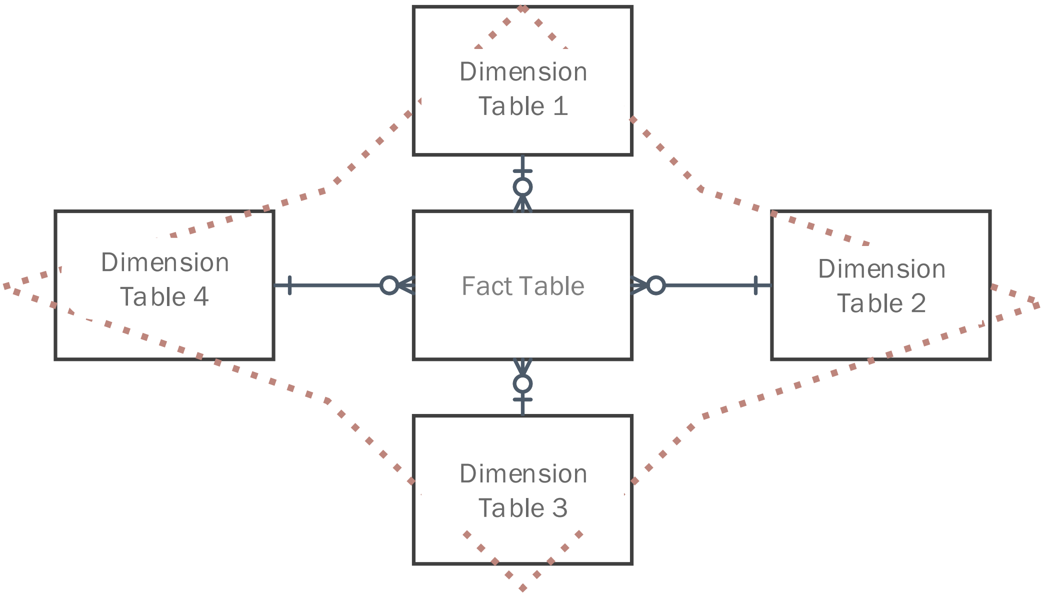


Figure 2.8: A generic structure of a database star schema.

of the attributes and the more they are in the dimension tables, the better description we have about the measurements.

The simplicity of the star schema gives a better understanding of the data by reducing the number of tables, and using meaningful descriptors help to reduce the number of errors and improve the performance as it needs to make less joins while requesting data using a query.

The more granularity that we have in our data, the better the expression and the dimensionality in the data warehouse as it adds different new dimensions to the schema. Even if a certain level of granularity is not needed, it is a good policy to have presented in the dimension tables, as it adds low space and very low impact on performance; on the other hand, the granularity is there for unexpected needed queries. For example, in the beginning we only need to represent the speed of the vehicles with a daily granularity, but for some reason we now need a hour or minute granularity.

2.5 GIS - Geographic Information System

As a lot of information captured from the smart cities devices is geographic namely vehicle and sensor position, the roads and buildings, it would be useful to use it effectively for transformations, calculations or representation, so handling this information using GIS is the natural approach [31]. Statistical geography has emerged during the last decades for the aid of finding patterns or similarities in spatial data sets in a way to provide new spatial theories. Geographic information systems are very good at facilitating the job of spatial analysis to provide a new way of manipulation, visualization and perception of spatial information to extract additional meaning and improving the understanding of the attributes being examined. GIS can provide techniques that ease the analytic research on spatial information for the user by improving the approach when exploring and presenting spatial data over the traditional methods.

Spatial data is becoming more available in most application fields such as health, education, commerce and transportation to solve real location problems using statistical and mathematical geography methods. Also, private and public sectors are starting to recognize the benefits of applying their models with geographic techniques to solve social and economic problems [32].

The GIS is used for spatial analysis to answer some questions/problems [33]:

- The simultaneous difficulty and importance of finding spatial, temporal, or space-time patterns in large spatial databases.
- The production of large databases raises a number of issues: How can the barriers to users generated by the complexity and size of many databases be reduced? How can the quality of the data be assessed? How can the data be used, even in light of inaccuracies? How might novel approaches to visualization be useful in addressing some of these questions?
- The recent improvements in computational and GIS technology generate a renaissance for particular methods of spatial analysis, such as point pattern analysis.

Geography and cartography must be used together to use GIS and build informative maps. To represent something geographically, it must be either a static geographic information, for example, a road, or an event that is dynamic and that occurred, for example, at a temporal change.

When representing things and events on a map, it is many times complex, as we want to inundate the map with information. The representation should have a certain level of abstraction without reducing complexity to highlight the essential components and ensure that maps and geographic information makes sense to people. The reliability of geographic information depends on the level of abstraction that is showed on a map. For example, a parking lot location on a map is useful for driving a car and not so useful when riding a bicycle.

Below it can be seen a list of Geographic Information System software, and table 2.4 shows a comparison of these geographic information systems applications.

[2] http://www.qgis.org
[3] https://grass.osgeo.org/
[4] https://www.esri.com/
[5] http://geoserver.org/
[6] http://mapnik.org/
[7] http://postgis.net/
[8] https://openlayers.org/
[9] http://leafletjs.com/

System	type	open source	observation
QGIS [2]	desktop	yes	open source; supports viewing, editing, and analysis of data
GRASS GIS [3]	desktop	yes	open source; supports raster and vector data
Esri [4]	desktop, server and mobile	no	GIS software, web GIS and geo-database management applications
GeoServer [5]	web server	no	edit and share geospatial data
Mapnik [6]	desktop and web server	yes	C++/Python library for rendering
PostGIS [7]	spatial database	yes	adds support for geographic objects to the PostgreSQL database
OpenLayers [8]	frameworks and libraries	yes	JavaScript library for displaying map data in web browsers
Leafletjs [9]	frameworks and libraries	yes	JavaScript library for displaying map data in web applications

Table 2.4: Comparison of a list of geographic information system applications.

Projections

Projections are the transformation of spherical coordinates into planar coordinates [34]. As the geographic information most of the time is collected as latitude and longitude coordinates, it needs to be projected to make it easier to relate with maps that people are familiar with to easily locate things and events.

Projections are based on different models of the earth. The most simple is the spheroid which is most used for the entire earth projections. Figure 2.9 presents the comparison of two commonly used models: the ellipsoid model is made for more detailed or smaller areas and fits the shape of the earth as it is used for horizontal coordinates (point in earth surface) in geodetic networks; the geoid model has the most accuracy, being very optimized for a particular area of the earth and more suitable surface for vertical coordinates (elevation and depths). The smaller the area, the more accurate the projection can be. Although there are these this different models, there is not one case that fits all, and so there is no standard earth model.

For the spatial measurements it is used a geodetic datum which is a set of reference points that take place on the earth's surface, defining the size and shape to map the earth [36]. Many different datum have been created and, while some are designed specifically for a certain country or region as it will be more precise in that area, others define better the

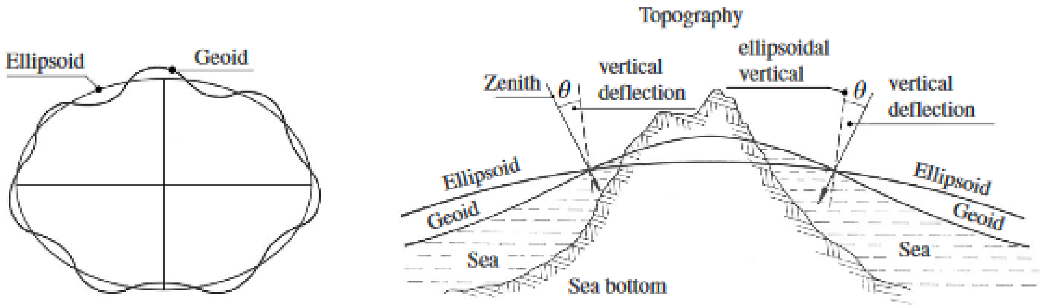


Figure 2.9: Ellipsoid and geoid comparison [35].

global earth surface.

One of the existing datums, the WGS 84, is a standard for use in cartography, geodesy, and navigation [37] designed to cover the entire world in its scope. This is the reference model that is used worldwide by the GPS systems and the one used in this work when using GIS operations.

OpenStreetMap data

OpenStreetMap[10] is a collection of map related data that is free and editable, built in by contributors that supply data about roads, trails, cafes, railway stations, and others, all over the world. This was impulsed with the wide range and affordable GPS receivers with a high quality location information that anyone can obtain. This project can enable people or small organizations that cannot afford digital geographic information that is multiple times considered as expensive [38].

For GIS applications, the OSM data is the base to work with geometric operations by letting the users import, edit, and tag OpenStreetMap data offline, providing a light structure that provides a reliable and scalable GIS. For example, the OSM street data can be imported to a PostgreSQL database to be used by the GIS application, the PostGIS to perform spatial transformation on the roads from OpenStreetMap.

The OSM data is available in OSM-XML format and Shapefile (SHP) format which can be obtained as a package of a continent, a country or a region from services such as GeoFabrik[11] [39].

This OpenStreetMap dataset comes with a specific format, as represented in the following table 2.5. The table contains the attribute type, the designation of the attribute and an example of the value of the attribute.

From all the attributes that come with the open street map dataset, only three are being used to provide the relevant statistics in this work:

[10] http://www.openstreetmap.org
[11] https://www.geofabrik.de/

attribute	designation	example
gid	unique identifier of the segment	24241
osm_id	open street map identifier	39587108
name	name of the segment	Rua do gerês
type	the type of the segment	residential
oneway	Boolean to check if the segment is oneway direction	0 or 1
bridge	Boolean to check if the segment is a bridge	0 or 1
tunnel	Boolean to check if the segment is a tunnel	0 or 1
maxspeed	maximum speed allowed in the segment	50
geom	geometry type of spatial reference of the segment	Multilinestring

Table 2.5: List of OpenStreeMap road information attributes.

- `gid`: the gid is used to provide an unique identifier of the segment.
- `name`: the name attribute is used so we can have the real name of the street and therefore the possibility to search roads by name.
- `geom`: the geom attribute is the most important one. With this geometry type variable, it is possible to make multiple spatial operations and transformations on the segments.

PostGIS

The PostGIS module is an open source spatial database extender that adds support for geographic objects for PostgreSQL object-relational database. It adds support of spatial data types for geographic objects to allow geographic and location queries to be run in SQL language [40].

PostGIS, as a geographic information system, enables a regular database to store geometric types of data in their tables, and also provides geometric functions and indexes that enable the manipulation of spatial data directly from the database and answer questions about space and objects in space.

In PostGIS there are three main geometry data types: the point used to represent locations, the linestring that is a sequence of points can represent a street, and the polygon that is similar to linestring, but where the start point is the same as the end point, to represent a region. From these data types we can calculate if a point is in the intersection with a road or if a road is inserted within a certain region, or calculate the distance of a road. For the geometries data types it is specified the spatial reference system that is the way of denoting the coordinate system defined for the geometry points. Different geometries using the same spatial reference system can then be overlaid without distortion. The reference for working with latitude and longitude data from different regions of the globe is the common used WGS 84 Lon Lat spatial reference system [41].

To analyze data from a data set of spatial information loaded into the database, PostGIS enables to query and do spatial operations that create new geometric objects that can be rendered on a map. For example, with a latitude longitude position of a bus, be able to know the road where the bus is and the distance to a certain bus station. To make this, it will be necessary to create a geometric point and intersect the point with a linestring representation of the road, and then calculate the distance from the geometric bus point to the geometric bus stop. All these geometric representations will need to be in the same spatial reference system.

Spatial Operations

To support the processing and analysis of the vehicles locations, it is used several spatial operations inside the database for road transformations with the help of PostGis specific functions. The main operations used are the following and can be followed with Figure 2.10. These functions enable multiple creations or transformations of spatial data, geometric operations and mathematical calculations with geometry types using spatial database queries. All these operations and more can be found in PostGIS documentation [40].

- `Intersection`: If two geometries share any portion of space, then a spatial intersection is taking place. This operation is used to check if a certain vehicle position is inside of a road from the existing ones from the database.
- `Line Locate Point`: Locates a point inside a line geometry from a range between 0 and 1. The point here is the coordinates of the vehicle, while the line will be the road segment. It is used to know exactly where the vehicle is located in the road segment.
- `Union`: Merge two line geometries into one geometry. This creates a unique segment by merging two different ones if they are nearby. This operation is helpful to create a customized road segment from the existing ones in the database.
- `Distance`: Gets the lowest distance in meters between two geometries. Useful to find the closest road to a vehicle location.
- `Make point`: Create a geometry point from latitude and longitude coordinates. This operation permits to use the spatial operations from the decimal coordinates. When making a point it is a good practise to specify the spacial reference system.
- `Buffer`: Modifies the road geometry, creating a buffer with a specified radius.

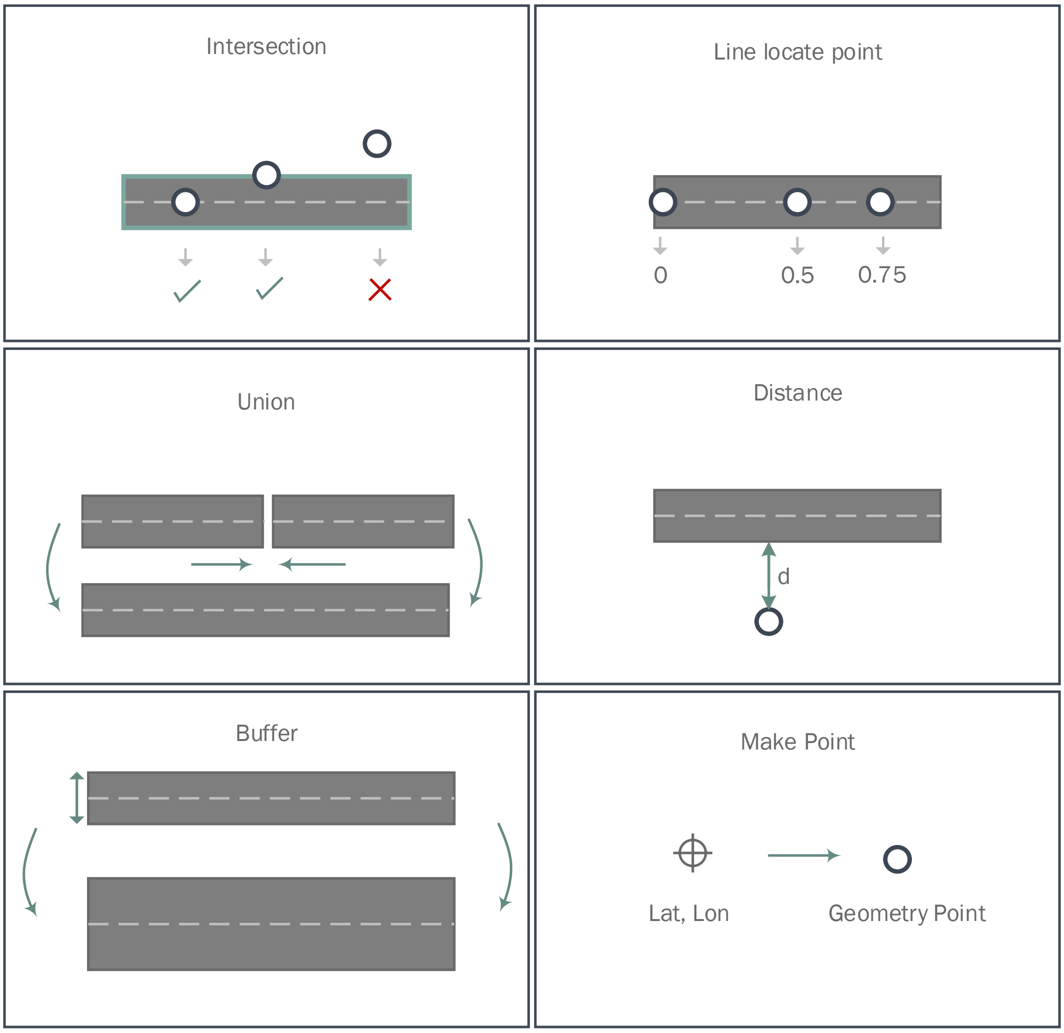


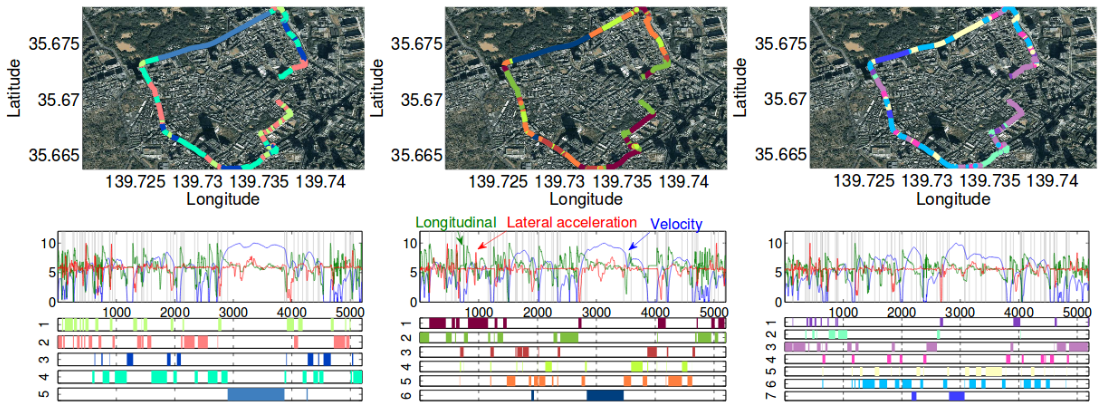
Figure 2.10: Database Spatial Operations.

2.6 Related Work on Mobility and Environment Analysis

This section presents the related work in the area of mobility and environment analysis.

TrailMarker [42] is an example of a work where different approaches are used to analyze real data sets of vehicular sensor data to find outliers, effectively finding typical patterns and points of variation on the road. This work relates to the present work by having similar objectives of finding road patterns, but is more centralized on the driver side patterns where acceleration and speed are used to predict the type of driver (inexperienced, aggressive, careful). Our proposed work is more centered on the street traffic profile as a whole and not so much on a driver profile as an individual.

Erdogan [43] proposes a geographical information system that has the objective of



(a) H-regime #1 "stable" (21/31 *trips*) (b) H-regime #2 "careful" (9/31 *trips*) (c) H-regime #3 "inexperienced" (1/31 *trip*)

Figure 2.11: Traffic mining approach in TrailMarker work [42].

---

assisting the analysis of road accidents based on accident reports. In the system, the accident locations are geo-referenced into the highways so it is easier to detect the areas with high rate accidents, in a way to take precautionary measures and improve the safety in critical areas. Our system, by making an analysis of the road traffic profile, can visually detect anomalies against an expected profile; although we cannot effectively predict if it is an accident or other event, it helps to improve that zone by detecting the exact location of the problem.

Another important topic is the study of pollution levels related with the road traffic. In an analysis and evaluation made on the main urban roads in Beijing [44], it was used an in place noise sensors along the road for multiple sampling points in order to identify the factors influencing the noise levels. From their results, where more than 3000 vehicles pass every hour, above their designed capacity, the noise levels exceeded the national standard during daytime hours. Other factors were also pointed, the road width, surface texture and the types of vehicles in the traffic composition. Our work tries to follow a similar objective, by correlating the pollution levels with road traffic, but with a more detailed approach on road traffic analysis in a more extended number of roads for an improved detection on the source of the problem.

The work proposed by Keay [45] aims to make a correlation of the road traffic volume in Melbourne city with the weather conditions in the same place. In the study, the focus is on how the weather affects the road traffic. It is compared different seasons of the year, daytime and nighttime instants, and wet against dry days. It concludes that the traffic volume has the greatest impact during the rainfall in winter and spring, where it is reduced on wet days in the daytime and all over the seasons during nighttime. Our work on the other hand tries to focus on how the road traffic affects the environment and pollution indicators.

Tripzoom [46] is a mobile application that collects mobility data and patterns from the users of the application to improve their personal mobility. It senses the mobility of the users, creating a profile about their trips, places, mobility patterns and transport type and motivating the users to change their mobility behaviour. It is made incentive measures for the users to change their type of transportation in a way to reduce the road traffic in the city. By gathering data from multiple devices, it is possible to see aggregated statistics of the users to check the impact of the measures over time.

Wu [47] proposes a decision support system for route choice with minimal costs based on dynamic traffic congestion pattern analysis. It uses GIS to manage the input data (travel time from two points) with a spatio-temporal database model, and queries and visualizes the result models in a graphical user interface. The result is a route planner support system that uses the departure time and destination to calculate the minimum cost of a trip to reach its destination taking into account the congestion and shortest path.

2.7 Discussion

The development made in multiple cities shows the progress that has been made towards the implementation of smart sensors and network. The works presented in this chapter, from the different cities, show the various infrastructures implemented: the environment sensors fixed in multiple places of the city to measure the pollution levels; location sensors equipped in vehicles to provide mobility patterns; sensors in parking spaces for public parking management; public WIFI network for the general use of the population. Also, another challenge is to connect these sensors and services to the internet through the appropriated network communication technologies for implementation issues or economic reasons, as some cities focus on fibre optical network, the Wireless Sensor Network communication, cellular network and WIFI. Some of the cities also adopted the living lab solutions for the application and services development, and also the Open Data project in a way to provide public access of the information captured in the city. By comparison, the base of this work, which is implemented in the Porto city, uses in a similar way fixed environment sensors deployed in the city, moving location sensors and WIFI APs assembled in the buses that provide public internet access for the users. The network implemented in Porto city connects the fixed and moving sensors and high range wireless technology, the WAVE/IEEE 802.11p that communicates with fixed base stations spread across the city, and uses cellular network (LTE) while the wireless APs are not in the range of the sensors.

The city dashboards presented in the current chapter show that the majority of cities collects and shows information from the public transportation and environment sensors, and the Dublin city adds another layer of information, the traffic information in the main city roads. Most of these dashboards show the real time or the last available information metrics from the sensors, but they generally miss on the data analysis and different indicators. The present work tries to take another approach by analyzing mobility and environment information in the city, based on the bus generated information, to display on the dashboard a detailed traffic profile on multiple streets and correlate the traffic and pollution data to find undiscovered issues in the city.

The presented related work on the mobility and environment data analysis shows that multiple studies were made using vehicular information for driver profile analysis, road accident location analysis, route choice based on traffic and the correlation with noise pollution and the traffic. In a similar data approach but with different objectives, our system uses bus position information to calculate the traffic information profile for the desired roads with information available, giving the possibility to see critical speed zones where road construction or accidents can be detected. It also uses the captured pollution metrics to make correlations with the traffic speed.

2.8 Summary

This chapter has provided an introductory explanation on the topics within vehicle mobility and geographic information, but also existing works on the same context.

The city of **Porto** is presented here as a city with an implemented vehicular network infrastructure and as the base of this work with the provided dataset of vehicular and environment information. The vehicular and environment information are detailed, and their captured attributes are presented.

It was presented the **smart cities** context where the work fits, describing the different layers that composed their network and the information that can be captured and used in decision support systems. It is presented a list of existing dashboards from different cities where it is compared their features and functionality. Moreover multiple city initiatives show the work that has been done in the smart cities context and IoT applications in their infrastructures.

The topic of **data warehouse** is presented here as it is the base of the storage solution of the system and generally used as data storage for decision support systems. It is explained the ETL (Extract, Transform and Load) procedure for the data transformation before reaching the data warehouse, and also the star schema structure generally used in the data warehouse modeling.

In the spatial and geographic context, it is presented **GIS** concepts and existing applications, where it is compared their purpose and functionality. The OpenStreetMap road dataset is also presented and their attributes explained, since it is generally used in GIS applications, and also in our work as the base of road information system. One of the existing GIS applications is the PostGIS spatial database extension also used in our work as the tool for making the described spatial operations.

Finally, in the end of the chapter it is given examples of **related works** in the area. Solutions to find the vehicle drivers behaviours, accident detection and prevention, the correlation of road traffic and pollution levels, and also with the weather conditions on the road. It also presents applications used to get the mobility profiles from users in the city and decision support systems using GIS for analysis and presentation of traffic information. All the presented related works have some related context, but our system approaches the road traffic in a different way, by getting profiles in each road individually, improving the level of focus and detail.

Chapter 3

Data analysis

3.1 Introduction

This chapter presents the techniques and algorithms used for the analysis of the bus and sensor network related data. It is showed how the vehicular data is gathered from the data sources, the steps made to position a vehicle on a road, the calculation of road and vehicle variables to be used in the decision support system and the persistence of the information into the data warehouse. From the environment information side, similar steps are made, the fetching of information from the source, the parsing and preparation of the data and finally the persistence of the information.

3.2 Vehicle Data processing

Using real vehicle data from the buses around Porto city adds a big amount of information to analyze mainly because of their number, 400 active buses. Although this amount of information will add more processing requirements, it gives a greater city wide analysis of vehicular information that will lead to more statistics options and roads coverage. This problem of the large amount of information is partially solved selecting only the data required to analyze, by restricting the data with a specific time window. Moreover the city manager will want to analyze a specific road at a time, so by restricting the data for a specific road will speed up the processing time.

Real time information also leads to some inconsistency's on the captured data mainly because the sensors are used in an environment where almost nothing is predictable, so some of the information will have some noise from the sensor readings against real data, like the GPS sensor readings having greater noise when nearby of tall buildings. To pass over this problem some methods using spatial operations were used to add precision to the analysis algorithm, in more detail in section 3.2.2.

The available information from Porto buses and sensors is summarized in the table 3.1.

origin	name	type	sampling	description
Bus	GPS	lat lon	every 15s	Bus position
Bus	time	datetime	every 15s	Bus instant
Sensor	environment	numeric	every 60s	environment metrics
Sensor	weather	numeric	every 60s	weather metrics
Sensor	pollution	numeric	every 60s	pollution metrics

Table 3.1: Available information used for the data analysis.

3.2.1 Vehicle Position and Time Instant Fetch

The vehicular data described in section 2.2 passes through multiple steps before reaching its way to be analyzed. From the sensor readings of the buses and passing through a deployed network infrastructure in the city, the information reaches the server where it is saved in a MySQL[1] database. It is from this database that the analysis starts, where the information is queried and fetched for the input of the traffic analysis algorithm.

The information is restricted by the GPS time and it is fetched portions of data with a range of an hour in each script cycle, so when the SQL query is executed, every entry in the vehicle database table from all vehicles that have available information at a certain time range is returned; for example, fetching every log data from all vehicles with a time range between 1pm and 2pm if the range specified is of 1 hour.

Moreover the returned information is ordered by node ID and GPS time respectively in the query. Ordering the result by the node ID, gives the possibility, in the next process step, to make an analysis of each vehicle sequentially until the next one without interruptions; ordering by GPS time, makes it possible to trace the progress linear in time.

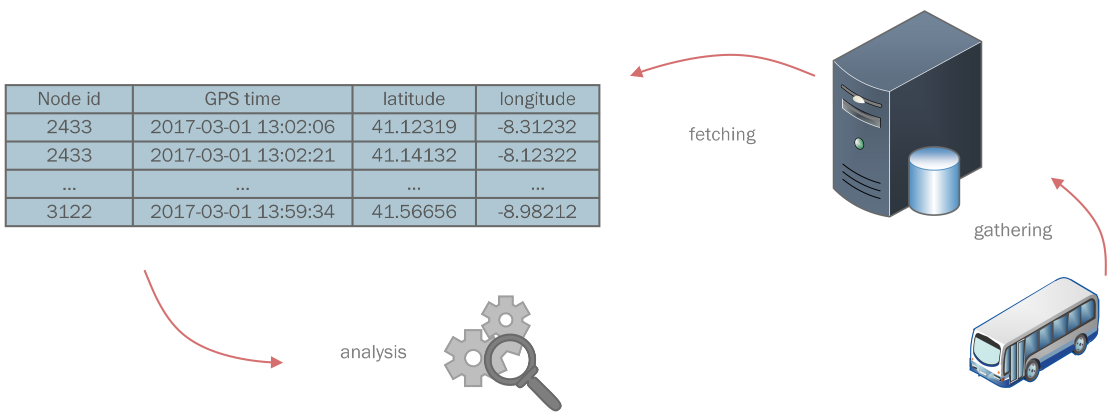


Figure 3.1: Vehicle Position gathering to a database server and fetching cycle for data analysis of position metrics.

[1]https://www.mysql.com/

3.2.2 Position to Road Snap Operations

In this section we describe how we map GPS location (e.g. bus , sensors) into roads given the GPS location jitter. This is essential to identify with precision the road from the bus position, to then make the necessary calculations to generate a speed profile.

After the information from the position of the vehicles is fetched from the remote SQL database and locally available to be processed, the first thing to do is to check if the vehicle is inside a road and, if that is true, locate the point on the road.

At each vehicle position, it is checked if there is an intersection from the vehicle position and the road geometry, previously presented in section 2.5. The latitude and longitude coordinates of the vehicle position are converted in a geometry type, with the help of spacial functions, to match the road type, which is also geometry. With both variables at the same type, it is then possible to execute multiple spatial functions. There are two main spatial functions used: the intersection that intersects two geometry objects and the line Locate Point to locate a point on a road. This process is described as follows and in figure 3.2.

1. A vehicle has a certain position, but there are no matching roads on the database. The next position of the vehicle is still off-road, so this position is also discarded. To check this, the list of roads in the database are tested with the vehicle position using the intersection operation.

2. At this point, the vehicle starts to enter an existing road. Here just a piece of the car enters the road, but because it intersects the road, it is assumed that it is on the road, and the ID of the road and fraction are fetched. The fraction is calculated using the line Locate Point spatial function, which gives the fraction of the current road geometry where the vehicle position is.

3. Next the vehicle continues its trajectory, it remains on the same road, so the same road ID is returned, but because it is on a different position, the fraction will be different.

4. The vehicle then changes its direction and enters in a new road. Now the road ID is different from the previous one.

5. Finally the vehicle leaves the previous road, and its position is not recognized from the available roads in the database. The position is discarded.

This vehicle location method will determine the precise location if the vehicle position data is perfect, but that does not always happen. In a real world, it is normal to sometimes have offsets of the coordinate values from the vehicle's real position. This precision problem also occurs in the working data set, since it is made of data from real vehicles. To correct or surpass this lack of GPS precision, it was implemented on the database a buffer on the roads geometry, adding a new customized geometry for each road available in the database.

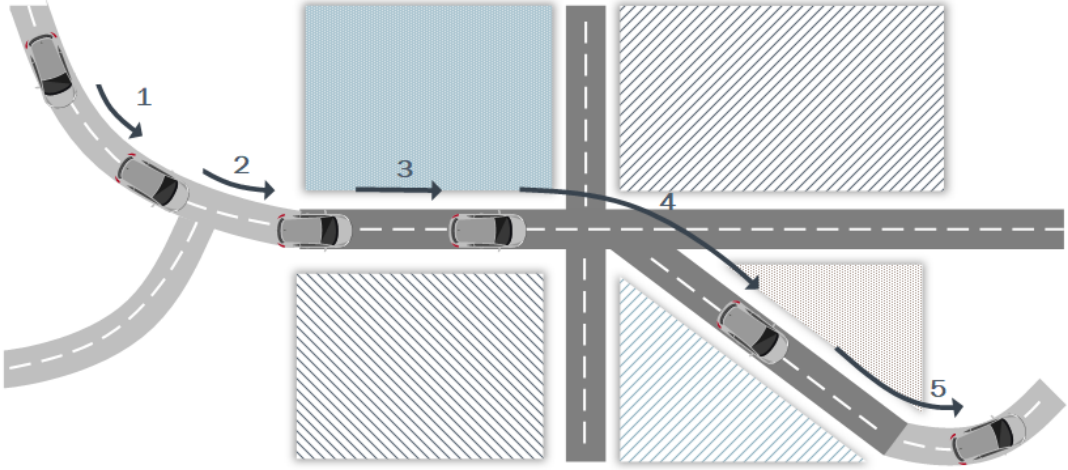


Figure 3.2: Vehicle trajectory on multiple phases on different roads. The light grey roads represent unknown roads, and the dark grey roads represent the ones available on the database.

With this method, it will be easier to snap the vehicles position onto their real road, and relevant points are not discarded.

Figure 3.3 depicts two different cases with the same vehicle positions in the same road. In one case there is a normal road geometry without a buffer, and below there is one with the buffer implemented. The vehicle in the real world follows its normal route, in its lane without problems, but the GPS reading shows sometimes some offset from the real data. Following the same figure 3.3, next it is showed with steps the way this method works and the difference it makes when a road buffer is implemented:

1. A vehicle enters a road, and in this step the GPS data and the real position matches. In both cases, with or without buffer the result is the same, there is an intersection and this point is not discarded.

2. The vehicle continues its route in the same road on its lane. The vehicle positions locate the vehicle a bit outside of the road, but still lives inside the road, so again there is an intersection with and without the buffer.

3. Next, the vehicle continues its trajectory normally, but now the GPS coordinates locate the vehicle outside the road. While in the road, without buffer, will not have an intersection and therefore this useful data will be discarded, with the buffer the point will have a match, and can be used to provided added precision of the algorithm.

4. Here the vehicle appears to have changed its lane because of the precision error. This will not affect the algorithm, and the direction will be the same as the previous one.

5. The vehicle leaves the road, counting in this step a hit in both cases, because it still intersects the road.

In the end, this example will work in both cases, but with different results. Without a buffer we will have a 4 in 5 matches while with the road buffer we have all the matches.

This method will not always be accurate. For example, if there is a loss of precision longer than the road buffer, that point will lie outside of the buffer and will not count, but the buffer should also not be too big, so it will overlap other roads.

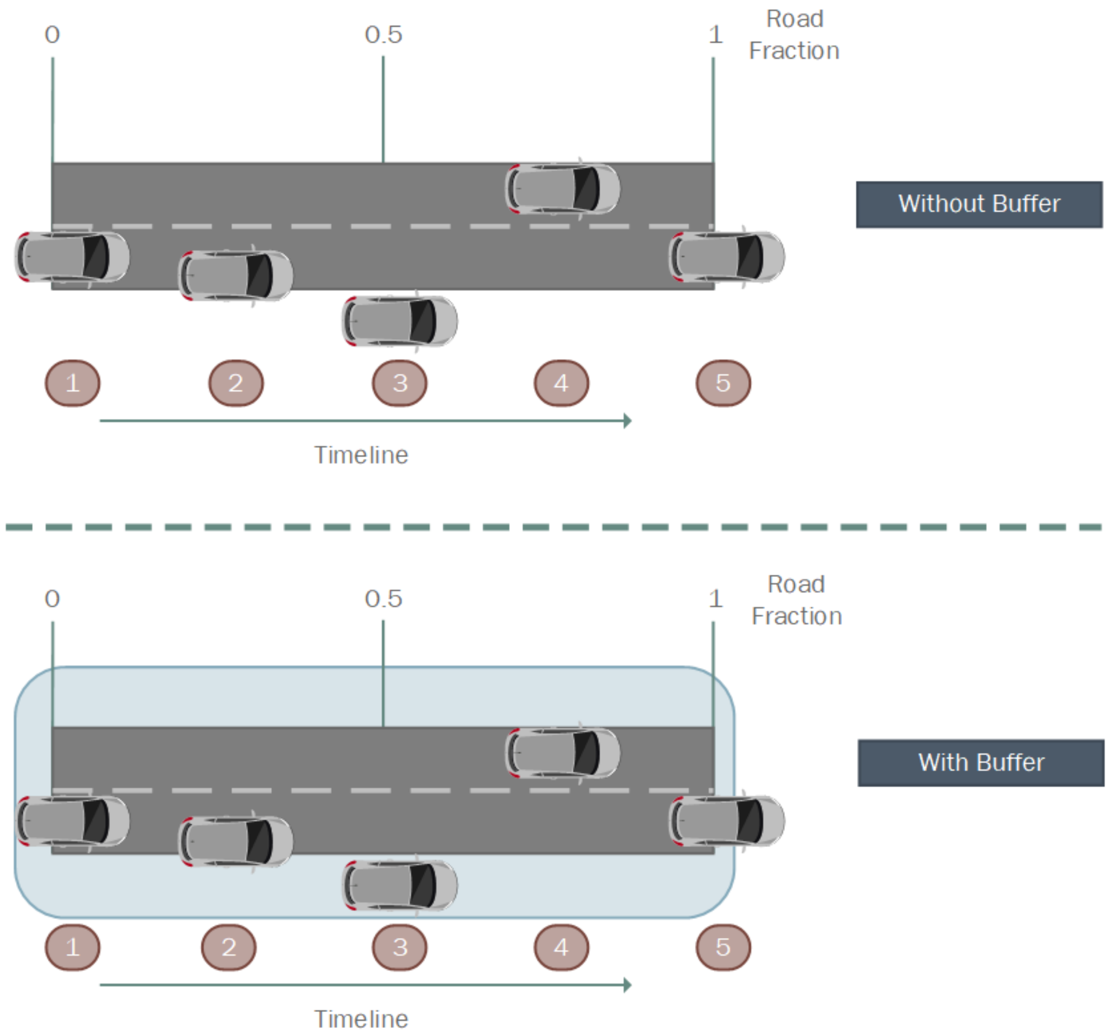


Figure 3.3: Comparison of the vehicle position road match without a buffer versus with a implemented road buffer on the same geometry road.

Using the same example of the figure 3.3, we can also observe how the fraction calculation works. This is independent of the road buffer, so it is valid for both cases. Each road has a beginning and an end, so in the beginning of the road there is a fraction of 0 and in the end a fraction of 1. All points inside the road are between these two values. With this method it is possible to attribute a fraction to a point, making it possible to locate it inside the road. In point 3, the car has a fraction of 0.5, which indicates that the car is in the middle of that road.

Because the beginning and the end of the road are not interchangeable, their fractions will always be the same. With this assumption, it is possible to calculate the direction of the vehicle based on two points. If the first point has a fraction larger than the next one, it has one direction; if is lower, it will have the opposite direction. This differentiation helps to cut the data, and select only a specific direction if needed.

3.2.3 Calculation of Vehicle and Road Variables

After the position of the vehicle is located within a road, the next step is to calculate the delay and speed variables of the points along the road. At each iteration there is a new coordinate and a new timestamp of a vehicle.

In figure 3.5 it is shown a trajectory of a vehicle. Each vehicle presented represents a new coordinate of the vehicle. Multiple steps are made during the trip of a vehicle as demonstrated in the algorithm 1, and it works as follows:

- After finding a match of the position to a road, it gets the identification of the road in order to be able to associate with the point object, but also the fraction of the road where the point is located; for example, a fraction of 0.5, which is the middle of the road.

- While the vehicle is inside the road, all the data points are associated with the road and used to calculate each individual **speed and delay** in different sections of the road. For this calculation, we will follow the figure 3.5:
  - Each bus in the figure is an entry log in the database.
  - After entering in a road, the first position marks the beginning of the iteration, and the latitude-longitude and time variables are saved on a list. At this point, no calculations are made.
  - For the following positions, the procedure is different. Using the *Haversive* formula (figure 3.4), it is calculated the distance from the previous point using the latitude-longitude coordinates, and it is also subtracted the time from the same point to be able to get the speed ($distance \div time$) in that section.
  - In the following step it is calculated the direction of the vehicle in that road. The first and third/last bus points are used to calculate the direction of the calculated variables between them. Here it is subtracted the fraction from both, and if the result is negative, it gives one direction, and positive otherwise.
  - During this interaction, the following variables are saved for each point in a list: ID of the bus, the ID of the road, the sequence of the point, the time, the speed, the distance, the direction and the fraction of the road. This also serves as a staging area before the data being stored in the database.

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

Figure 3.4: The Haversine Formula [48].

- The vehicle **leaves the road**, and the spatial intersection matches a different road, finalizing the calculations from the point before. All this list of calculated variables that belong to the previous road is persisted to the database.
- The cycle is restarted after being mapped to a new road.

During the algorithm execution, it is created a new object for each point to save in memory specific information, and for the calculations that are needed either for some processing operations, and also to persist those attributes into the database. The object attributes created are the following:

The **node id** is an object attribute created to store the unique identification of the vehicle.

The **road id** is the unique identification of the matched road, and it is used to make a connection between the point position and a road.

The **sequence** number attribute is a sequential counter to save the order of the points in each road.

The **gps time** attribute holds the date and time of a point.

The **delay** attribute is the time difference in seconds from the previous point in the same road.

The **fraction** variable is the point position in a road in a 0 to 1 range, where 0 is the beginning of the road and 1 the end of the road.

The **distance** attribute is the distance in meters between the previous point in the same road.

The **latitude** is the latitude coordinate of a point.

The **longitude** is the longitude coordinate of a point.

To follow and better understand the algorithm, in the table 3.2, it is represented an example of the allocation of values to point variables during a bus trip, using the same example in Figure 3.5.

Analyzing the example of the allocation of values to the point variables in table 3.2, we can obtain the following information. The **node id** is always the same, which means that it points from the same bus. The **road id** identified is the same during the first three steps and different on the fourth, when the bus changes the road. The **sequence** number is incremental in the same road, restarting on the new road. The **gps time** is incremental for the same bus, as it was ordered during the fetching of information. The **delay** and **distance** do not have a value in the first point of a road, because it is the first one and does not have a point before to compare in the same road. The **fraction** gives the increasing position along the movement of the point in the first road and on the new road (its respective position is between 0 to 1). The **latitude and longitude** are the actual coordinates of the vehicle, used for the distance calculations and road mapping.

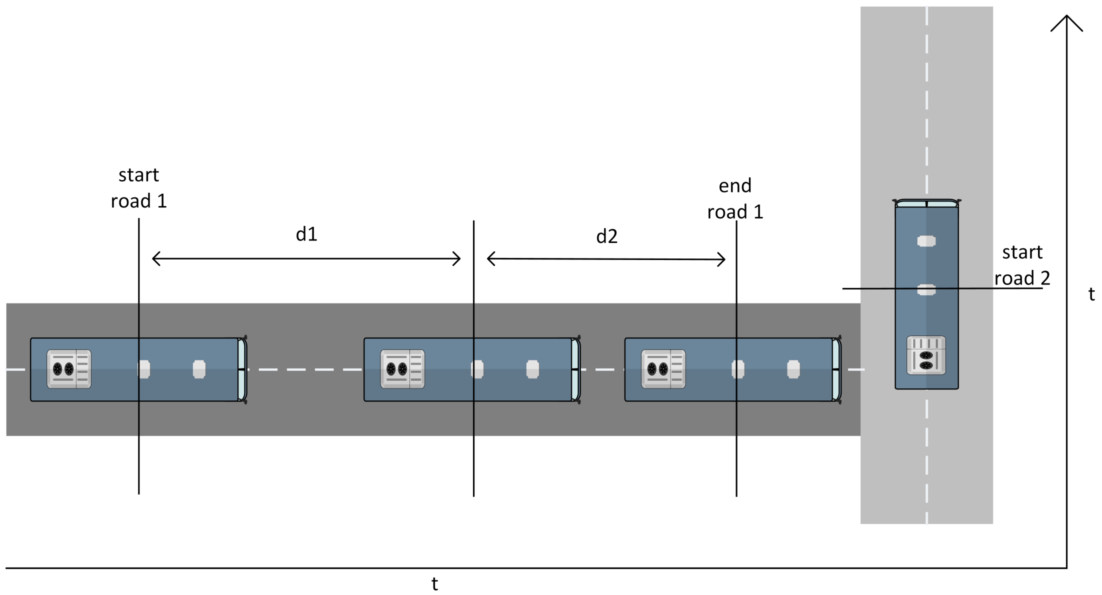


Figure 3.5: Four positions from a unique bus trajectory during its trip represented in two different roads. t = time, d1,d2 = distance, start = fist position in the road, last = last position in the road.

3.2.4 Persistence of Information

When a vehicle leaves a certain road, the processed information (Point Object) stored in an array is ready to be persisted in the database. Because the database has a data warehouse structure, a few steps are needed to be taken.

One of the steps is to make a link between the dimension tables and the fact table. Here the dimension tables used are the date which holds all the metrics related with the date and time, and the road table that has geometric information about the roads. The fact table is the main table that holds the majority of the environment data. It is called a customized database function to insert the date in its specific dimension table and get the respective ID.

The second step is to get the course number. The course number is a sequential number allocated to each array of points, to identify each sequence of points in a road. It is made a SQL request to find the last one available in the database to then increment by one.

To finalize, all the list point object values belonging to the road, plus the date ID and course number are used as input in another database function that persists the information in the fact table.

Algorithm 1: Road points variables calculation.

```
input  : Records: A number of vehicle data records
input  : Query: object for querying the database
output: a list of road segment variables
points_list ← ∅;
off_points ← 0;
last_road_id ← 0;
foreach entry ∈ records do
    node_id ← entry[0];
    gps_time ← entry[1];
    latitude ← entry[2];
    longitude ← entry[3];
    road_id, road_fraction ← Query.function.insideRoad(latitude, longitude);
    if road_id = 0 or road_id != last_road_id then
        if off_points = 3 and points_list is not empty then
            if points_list[0].fraction − points_list[−1].fraction < 0 then
                direction ← 0;
            end
            else
                direction ← 1;
            end
            if points_list 2 then
                Query.persist(points_list, direction);
            end
            points_list ← ∅;
            last_road_id ← road_id;
        end
        continue;
    end
    off_points ← 0;
    if points_list is empty then
        road_point ← createObject();
        road_point.setNode(node_id);
        road_point.setRoad(road_id);
        road_point.setSequence(1);
        road_point.setTime(gps_time);
        road_point.setDelay(0);
        road_point.setFraction(road_fraction);
        road_point.setDistance(0);
        road_point.setLat(latitude);
        road_point.setLon(longitude);
        points_list.append(road_point);
    end
    else
        last_point ← points_list[−1];
        sequence ← last_point.sequence + 1;
        delay ← gps_time − last_point.datetime;
        distance ← havesineDistance(last_point, lat, lon);
        road_point ← createObject();
        road_point.setNode(node_id);
        road_point.setRoad(road_id);
        road_point.setSequence(sequence);
        road_point.setTime(gps_time);
        road_point.setDelay(delay);
        road_point.setFraction(road_fraction);
        road_point.setDistance(distance);
        road_point.setLat(latitude);
        road_point.setLon(longitude);
        points_list.append(road_point);
    end
end
```

attribute	Step 1	Step 2	Step 3	Step 4
node id	241	241 (=)	241 (=)	241 (=)
road id	12314	12314 (=)	12314 (=)	321 (≠)
sequence	1	2 (+1)	3 (+1)	1
gps time	2017-07-18 00:25:10	2017-07-18 00:25:30 (+20s)	2017-07-18 00:25:45 (+15s)	2017-07-18 00:26:15 (+30s)
delay	—	20	15	—
fraction	0.2	0.5	0.8	0.4
distance	—	35	23	—
latitude	41.2312	41.2433	41.2787	41.1231
longitude	-8.8643	-8.8433	-8.8475	-8.6324

Table 3.2: Example of the allocation of values to the attributes of each point in the four steps from the figure 3.5.

3.3 Environment Data processing

The processing and analysis of environment data is made with multiple steps, which first needs to be fetched from the remote data source and, being available locally it can be parsed, cleaned and structured by the existing data types. The last process is the preparation of the data to be stored in the database specific format to then be persisted.

3.3.1 Environment Metrics Fetch

The environment information is captured by sensors deployed in the Porto city, that send their collected metrics through the implemented vehicular network. This information, after moving through the network in the city, reaches a server that provides the last environment information for each sensor. It is a Fiware[2] server that stores the environment information and provides its access through a RESTfull web service interface. Figure 3.6 depicts this process.

In order to get the sensors data, from the local server script it is made a request of the session token with the Fiware login credentials, that are necessary to request information from the server. For security reasons, this token has a validity of one hour, so it needs to be renewed periodically. The server checks the credentials and succeeds the authentication with a token response value.

After having the token, it is made a HTTP GET request with the token in the parameters to get the information from the sensor's. The Fiware server provides the last sensors environment information of the multiple sensors in an unique JSON message. After receiving the message, it can now pass to the parsing and processing part.

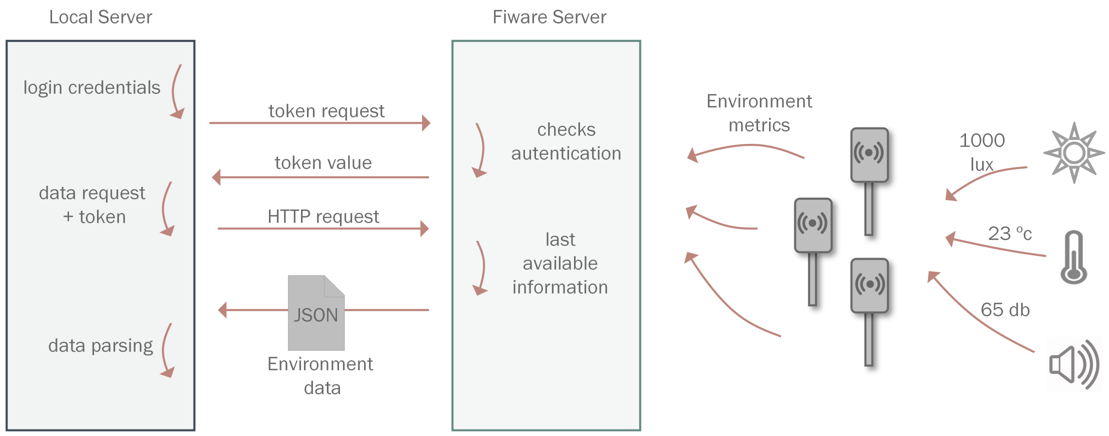


Figure 3.6: Representation of the process of Gathering environment information from sensors, the transmission of the metrics to the Fiware server and the request of environment information from the local server.

[2]https://www.fiware.org/

3.3.2 Parsing and Preparation of Data

The environment information gathered from the sensors is at this step locally available and ready to be processed. The first step to make is to parse the message which is in JSON format.

In the parsing process it is iterated the list of sensors to process each one separately, and then in each one, get the environment variables by their type:

- The **weather** that contains data from the wind, precipitation, and solar radiation.
- The **basic environment** contains data from the temperature, humidity, and luminosity.
- The **noise** that contains data from the environment noise.
- The **air quality** that contains data from the o3, no2, CO and particles.
- The **general** contains information from the sensor, like the ID, coordinates and type.

Each type of sensor data has its own associated *timestamp* of the captured metrics. The data type metrics are prepared and cleaned by checking if the values fall in their expected range and also if the values exist, which most of the time sensors do not always provide their information. All these metrics, with their associated *timestamp*, are coupled into an object of their specific sensor and added to a list. This is made sequentially to all sensors in the JSON message. When all sensor metrics are finished in the processing and added to the list, they can now be persisted.

3.3.3 Persistence of information

The environment objects stored in the list can, at this step, be stored in the database. Because it has a data warehouse structure, the information needs to be prepared for the specific format. The list of objects are iterated and a few operations with the database are made for each sensor:

- It is called a database function to send sensor general data type. This function checks the ID received to search in the sensor dimension table if it already exists, creating a new entry with the ID, latitude, longitude and type if it does not exist. It also adds a geometric point to a table column based on the input coordinates.
- Next it is made a request to another database function to get the ID of the associated date from the time dimension table. The function receives the *timestamp* as input, and creates a new date entry if it does not exist, returning always the associated ID.
- Finally, the information of the environment information is persisted in the sensors data fact table, adding a new entry in the table with the associated sensor ID and time ID.

3.4 Building the Traffic Profile

The traffic profile of a road is a way of observing the speed of the vehicles through the distance of the road. The process of building the traffic profile is presented in the traffic profile API which creates the profile when requested from the dashboard. A user from a selection of inputs form a request from the dashboard, providing as inputs the road name, the time range, the direction, and the distance range.

The road information from different vehicles that were already analyzed is requested from the database using the inputs received as conditions to segment the information to a specific road, left or right direction, two different dates and two distances of the road. The distance ranges in the area between the initial and end distance of the road that the user wants to analyze. For example, a road that has 800 meters, but just wants to analyze the first 400 to have more detail, so it will be specified a range of 0 to 400 meters.

The information that is resulted from the request contains points as objects, which includes the ID and speed of the vehicle, distance from the previous point and fraction of the road. This distance is then accumulated at each point to give the real distance of the point on the road for each vehicle route. For example, in figure 3.7, having 3 points returned from the database of the same bus gives initially the distance between then, but to map the points to the real road distance, their distances are accumulated.

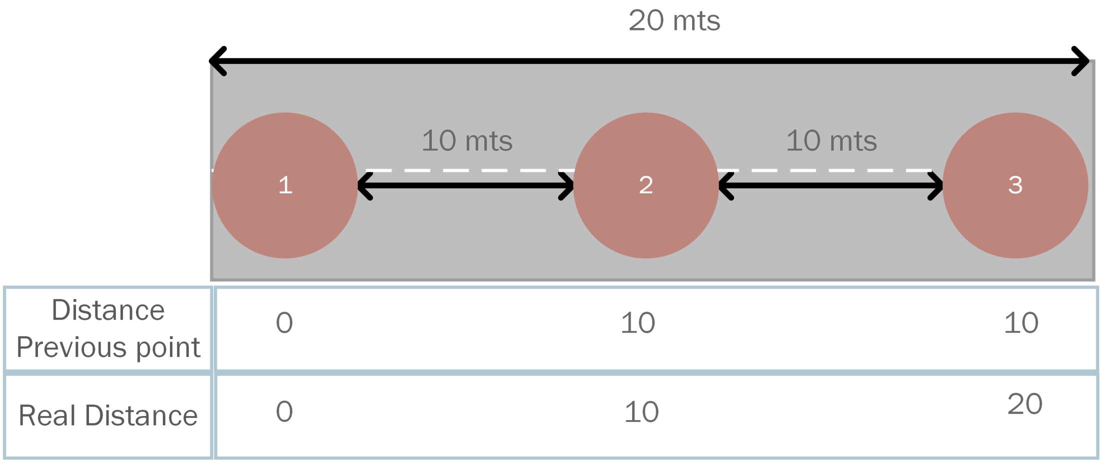


Figure 3.7: Distance transformation approach. Each one of these points represent a vehicle position on the road, with a known distance between them that is accumulated to know the real distance of each point on the road.

After the points be mapped to the real distance, they now are not aggregated by vehicle, but by distance. As figure 3.8 shows, multiple points from different vehicles create a profile through the distance of the road.

By having the points aggregated by distance, it is now possible to select multiple sections of the road, and calculate for each one the 25%, 50% (median) and 75% quartile of the speed values to give an approximated curve of the speed in the road where the majority of the vehicles are.

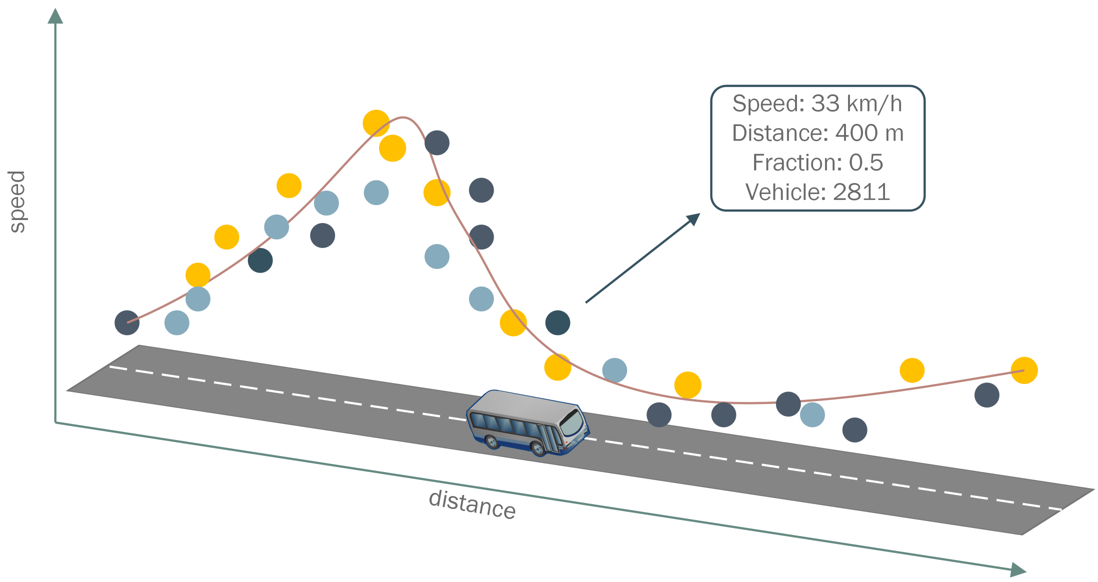


Figure 3.8: Example of the road speed profile with an aggregation of points from 3 different vehicles through the distance of the street. The metrics represented in the figure are from one point and represents the speed of 33 km/h from the vehicle 2811 at the distance 400 meters of the street, which is also the 0.5 fraction of the road, in the middle.

3.5 Traffic and Environment Correlation

The impact of city traffic in the environment is the main use case selected. It allows relating information that could be useful to city planning and city quality improvement; this can be achieved by relating road traffic profiles with co-located environment sensors along time. Relating peaks in both traffic and environment readings can help to identify time and space patterns that may be worth attention and/or localized intervention because they are affected by themselves.

The calculation of the correlation is made in the sensors correlation API, which uses the following steps to make the calculations in real time:

- The service receives a specific sensor and time range from dashboard inputs selected by the user.
- It is fetched from the database the average of the environment **measurements** (noise, CO) aggregated in the different hours of the day for the specific sensor and date range.
- As shown in figure 3.9, from a geographic position of the environment sensor, it is made a radius of 100 meters to **find the roads** that lie inside that area. To do this, it is used PostGIS spatial functions to create a spatial circle area, to then make a spatial intersection operation with the list of available roads, returning the ID of the valid roads from the database.

- The IDs of the valid/nearest roads are used to **Fetch the average speed** values from the multiple sections that lie inside the circle, but now aggregating by hours of the day from the same date range as the data from the sensors.

- The variables from environment metrics and average speed are **associated** to the same hour as a point of connection, so both profiles can be visually analyzed and correlated.

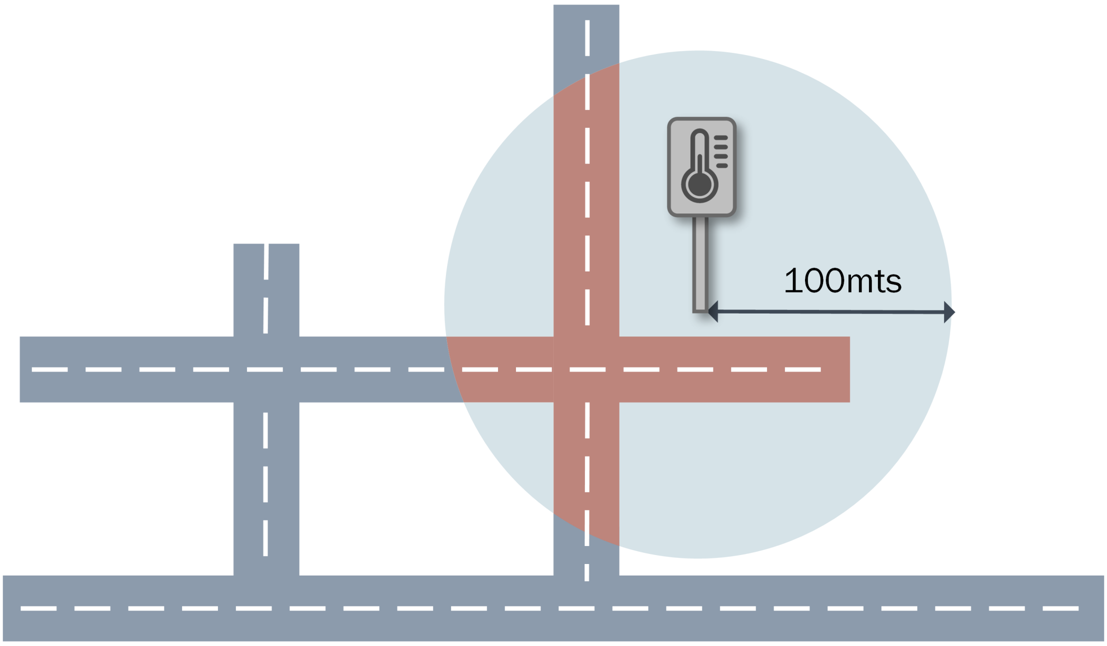


Figure 3.9: Approach used when choosing roads to correlate the speed information with environment information. Here is chosen the roads that are closer to the sensor that provides the environment metrics.

3.6 Road Cleaning and Transformation

The list of road segments imported from the OpenStreetMaps dataset provides a good source of road geometry information, but multiple times a single road is too much segmented to be used with the granularity of the vehicular information. To reduce the segmentation of the roads it is created an algorithm that joins the multiple segments into one using different spatial operations. Figure 3.10 represents the result of the road transformation made with the algorithm 2, which is part of the road building service and works as follows:

- Receives as input the list of segments ID to union, the road name and an object for making queries with the database.

- First it checks in the database if the segment union was already performed by getting the gid of the segment with the provided road name.
- In case the gid value is a valid number, it means that the road already exists and it does nothing more and returns the gid.
- If the road name is not found, which means that the gid assigned value is null or empty or 0, it will attempt to create a new one.
- It executes a segment union spatial query from the list of segment IDs provided and gets the assigned gid after creation.
- After the union is performed, it creates the road buffer spatial operation for the new geometry.
- Next step is a method to check if the road union was well performed to support all spatial operations. To check if the built segment is valid, it is executed a spatial operation which locates the vehicle on the road. It returns a value between 0 to 1 if the operation is performed with success or an exception if the segment is not valid for this operation.
- The result of the point location is checked and, if it is null or empty, the operation is not successful and the segment created is deleted from the database. It is assigned the gid variable with -1 value as an indication of road creation failure.
- Lastly, the gid is returned.

Algorithm 2: Join of multiple existing segments to create a road.

input : id_list: A list of the ids of segments
input : road_name: The road name
input : Query: object for querying the database
output: gid: the id of the created road

```
gid ← 0;
gid ← Query.GetSegmentGID(road_name);
if gid is NULL OR gid = ∅ OR gid = 0 then
    gid ← Query.UnionSegments(id_list);
    Query.CreateSegmentBuffer(gid);
    result ← Query.lineLocatePoint(gid);
    if result is NULL OR result = ∅ then
        Query.deleteSegment(gid);
        gid ← −1;
    end
end
```

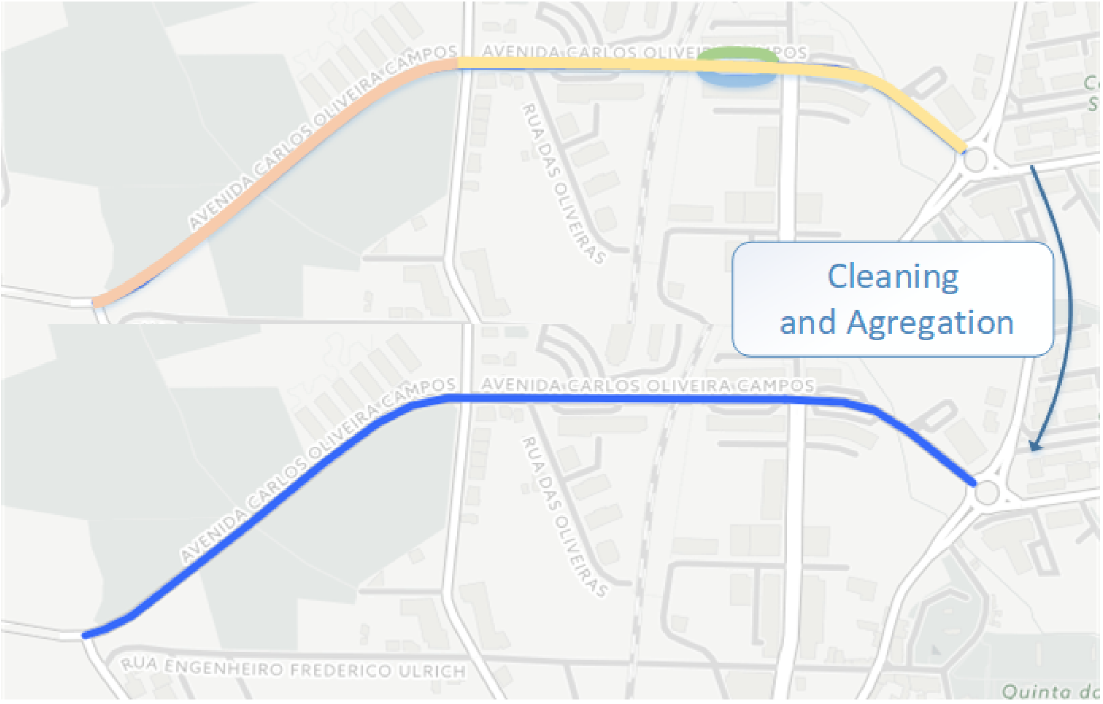


Figure 3.10: Example of the Avenida Carlos Oliveira Campos street initially composed by four different segments, being cleaned and transformed into one segment.

3.7 Summary

In this chapter we explained the techniques and algorithms developed when handling with raw data. On the vehicular data gathered from the buses, we have seen the cycle of the data before reaching the analysis of the algorithm and how important it is to be ordered by their GPS time.

Multiple spatial operations were made to find a road that matches the position of the bus, where the intersection operation is the main solution used to locate the vehicle in the road. A buffer implemented on the road segments is a way to overcome with the lack of GPS precision in some areas.

During the movement of the bus inside a road, multiple variables are calculated to each section of the road: the speed and delay in each point inside the road using the calculated distance to the previous point, and also the time difference for the same previous point. This calculated variables of the point object are then prepared and persisted into the data warehouse structure.

For the solution to handle and process the environment information from the sensors deployed in the Porto city, it is demonstrated the steps made when requesting the information from the remove RESTFull server and the type of variables that come with it: how it is parsed and prepared before being persisted in the data warehouse structure.

For the road traffic profile presentation, the techniques of the aggregation of multiple points from a selected road were made in a way that the median speed curve of the vehicles along the road can be calculated.

The correlation of traffic and environment information is made by selecting the environment metrics from a specific sensor to then find the closest roads with road traffic information available at the same space of time, and aggregate in the same hours of the day.

The Road Cleaning and Transformation methods are a way to transform the roads from the OpenStreetMap dataset, where a single road can have multiple segments and branches to a unified segment, to be compatible with the road analysis algorithm.

Chapter 4

The System

4.1 Introduction

In this chapter we describe the overall architecture and implementation of our system, with a description of its overall components.

The chapter starts by presenting the global system flow architecture, its components, interactions between the different pieces and the data flow from the source to the application. It is focused in the storage components as the central part of the system that links the processing component to the application component.

It also describes the provisioning of the information for the dashboard by fetching it from the database, which is made by web services. Multiple services provide the necessary endpoints to make data available while transforming it for the necessary requirements of the data presentation. The existing services are provided for road traffic information, environment and traffic correlation, and road definition and management.

The final part of the chapter presents the web based dashboard, where the multiple visualizations of the different interfaces are displayed and explained.

4.2 The architecture

Figure 4.1 depicts the overall system architecture. The data management is where all the data gathering (fetching available data from different inputs), data processing (the processing of information and storage) and data provisioning (provide processed information through web services) occurs.

The information needed to be processed is available from different sources of data. The vehicular data is stored in a relational database and it is queried and fetched from a python script. The environment data is available in a REST web service and the requests are also made from a python script. The Open Street Map data set is imported from a file directly to the local database.

The processing part of the system is composed by two separate components: the one that processes information from the buses and the one that makes transformations on environment sensors data. During the execution of the processing of the vehicular information, interactions exist with the database, which are required to aid in some spatial calculations, but also for the persisting of information. For the sensors processing component, the connection with the database is used mainly for storing the prepared information in a data warehouse format.

For the storage of both mobility and environment information, a relational database type is used, using a generic data warehouse structure adapted for the specific types of data and granularity.

The information stored in the database is made available to the outside with interfaces from web services. These web services provide information from the database and, in some cases, they require transformations on the data structure. One of the available APIs, the Road Management API, is present to provide a link to the management and transformation of data. The Road Traffic API is responsible for making the bus spatial information accessible to be used in the web application by requesting it from the database. The Sensor Data API serves the web application with environment information correlated with mobility information.

The web application is the component that requests vehicular information from the web services to fill its interface with valuable data. The other endpoint of the application is the user, partially responsible for the requests made.

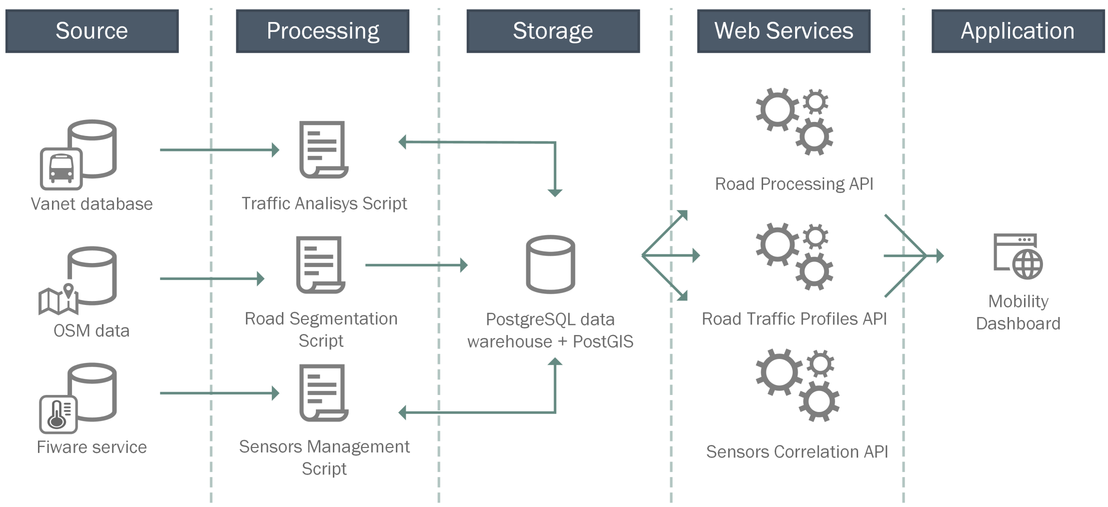


Figure 4.1: Proposed system flow architecture.

The components of the processing part are hosted in the same machine, but separated in different containers, so each component has its own independent environment, and therefore, it is not affected by other components. The database, the scripts and the web services have their own container.

Data Storage

The data storage component consists of a database that stores all the processed information used from the different data types and also backing the multiple components in the overall architecture. The main focus of the database is to support the web based application with information in an acceptable time frame.

The database used in the project is based on the PostgreSQL relational database as it supports multiple spatial extensions and it has a good documentation. The relational databases support functions that allow to have logic in the database side, and also supports extensions to add a new functionality. Another important aspect is its flexible implementation options, so it is simple to make a data warehouse structure.

The database structure is a simple form of the data warehouse architecture. The Star-Schema is the style used, as it is composed with dimension and fact tables. The reason to choose the Star-Schema structure is because it supports multiple granularities of different dimensions allocated in the dimension tables, which is needed for the flexible data requests from the dashboard. Another advantage of this architecture is that it can reduce the space occupied in disk by not repeating the same data in the database (each dimension entry can be attributed to multiple metrics), which is useful to save space for the large vehicular data-set.

Several functions in PostgreSQL[1] database are implemented to shift many operations from the application side to the database. This shift has several advantages: first, it promotes re-usability of code because these functions can be used from different applications without the replication of code; second, it reduces the overhead from application and network as it can make multiple operations inside the function with just one request; and finally, the SQL operations made are closer to the database which increases the speed of these operations. On the downside the functions will be strongly coupled to the database vendor.

Figure 4.2 depicts the global view of the structure of the database.

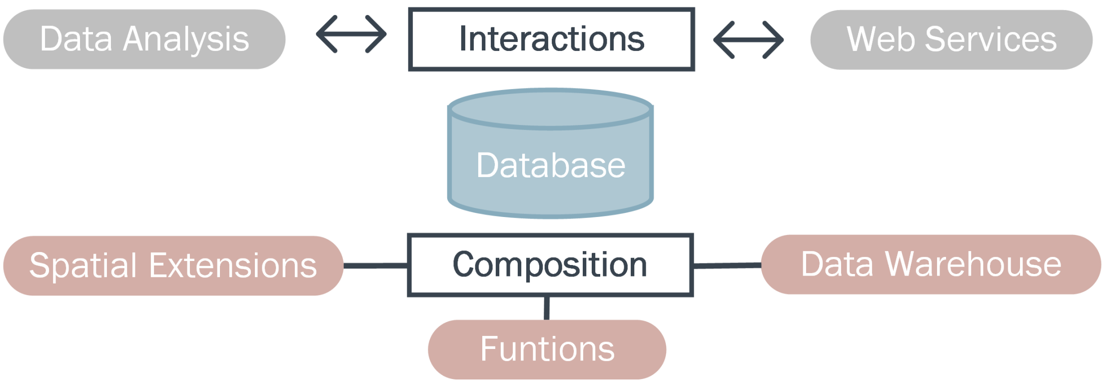


Figure 4.2: Interactions of database components and its intrinsic support.

[1]https://www.postgresql.org/

4.3 Data Information Services

Web services are an interface implemented between the local data warehouse and the outside applications. These services provide the necessary data used by the statistics by requesting the database, and also manipulates and format the data in a specific data format for the applications. Some of the services have more logic and will perform the needed calculations to support the statistics.

In this work the services mainly support the dashboard with information for the road traffic statistics, the sensor data and the road management interface. Although the end-point of the web services is used only by the dashboard, other applications, if needed, can also use this endpoint, to access processed data or reusing some of the services logic.

The road traffic, road processing and sensors correlation API contain some of the services that provide information for the section of the dashboard where it is showed statistics for the traffic in the Porto city. The services are the following.

- `Available Roads`: this service provides the list of available road segments that exist in the database and were previously created. From the request, it returns the ID and name of the road segments in a JSON format. This service is normally used in the dashboard for the user to be able to choose the specific road to analyze or process information.

- `Road Comparison`: provides the speed profile on a road for two different roads at different time frames. It uses the median of the speed profile of the road for two different dates to provide a comparison at the same level of the road, the distance. The correlation is also calculated and provided, as it gives the relationship between the two profiles.

- `Dashboard Inputs`: saves the selected user inputs on the dashboard from the detailed view (figure 4.5), receiving from the dashboard the selected road, direction, date, hour, quarter and weekday range to be saved. It also provides the saved inputs list that can be used as a fast way to view and request previous searches saved by the user.

- `Road Processing`: This service serves as an REST endpoint for the road processing analysis referred in the Vehicle Data processing algorithm, receiving as input the date range and road ID to start a new data processing phase. It also gives the processing status, providing the road being processed, and also the completeness percentage of the processing.

- `Traffic Profile`: The traffic profile service provides information about the speed profile of the vehicles in a specific road. As previously described in the Traffic Profile algorithm, it gathers the multiple road points from the vehicles from a specific road to generate the 25%, 50% and 75% speed of vehicles in the different distances of the road, providing the results in a JSON message.

- `Traffic and Environment Correlation`: provides the information about the both environment sensors and road information in the same perspective. The previously explained Traffic and Environment Correlation algorithm is the base of the service, where the calculated environment metrics and the average speed of the streets near the sensors are associated to each hour of the day, to then build a JSON message that the dashboard will consume.

- `Road transformation`: service that, based on the input of multiple segments ID from a road, makes the cleaning and aggregation of the segments into one, returning the ID of the new segment if the process was made with success. This process is described previously in more detail in the Road Cleaning and Transformation algorithm.

4.4 The System Web based Dashboard

The main user interface of the system is the web based dashboard as the final endpoint of the system where the user can finally see the meaning of the data aquired from the vehicular network. The dashboard is composed by multiple types of visualizations to represent the vehicular data transformed in traffic information and environment sensors metrics aggregated with traffic information.

The dashboard contains different views to show the data in different perspectives: the global traffic view that shows the traffic in a less detailed top view; the traffic profile view that shows a more detailed view of the traffic profile in a specific road; the traffic comparison view gives a possibility to analyze the traffic profile in two different dates at the same time; and the sensor correlation view that compares the traffic information with the environment data. The dashboard also presents a management area, which provides a way to manage road segments, and another page that manages the processing of information with instructions and status.

4.4.1 Mapping of roads and locations

One of the visualizations of the dashboard is the global view of the traffic in the city as represented in figure 4.3.

The main inputs from this window are temporal. The user can first choose a week to then select a specific weekday to automatically update the chart with traffic information, if available.

From this visualization there is a chart that presents the average speed and number of vehicles through the hours of the day after the user selected the previous inputs. In the chart, by hovering in a specific hour, the user can then get a visualization in a map of the traffic in different sections of the multiple roads that have processed traffic information for that specific time. The legend of the map helps to interpret the colors displayed on the road segments, by attributing a specific speed range to each of the colors.

The main objective of this visualization is to find the most critical hours of the day where the average speed is low or the vehicle count is high, from all the available roads, by using the chart on the left and the most congested roads by visualizing the color mapping of the road segments on the map.

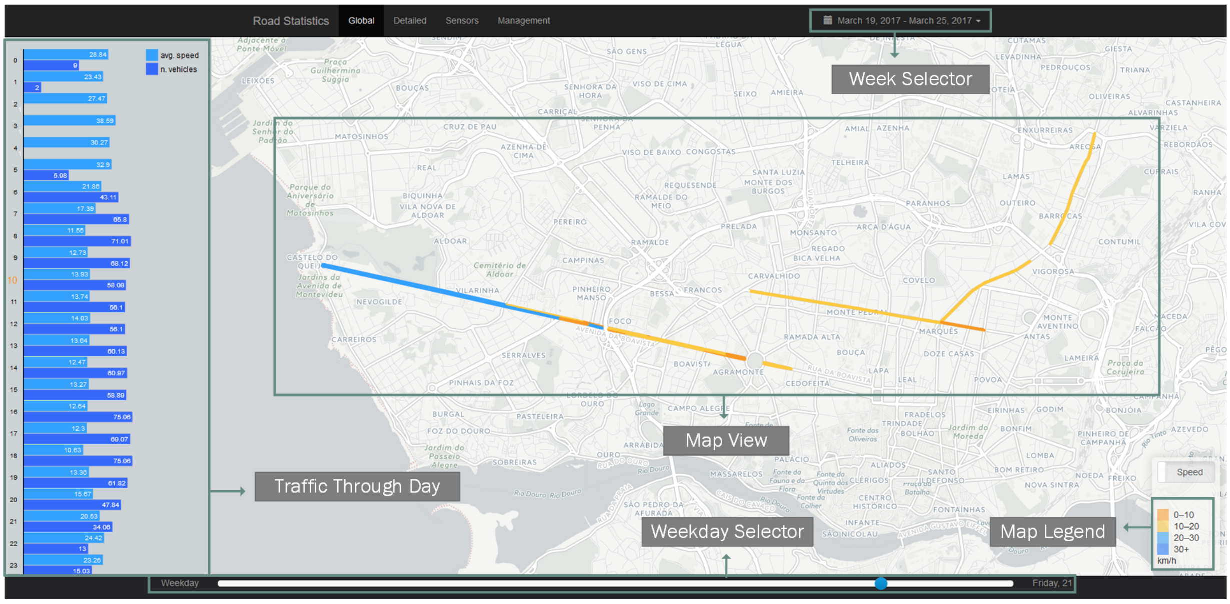


Figure 4.3: Global road traffic.

4.4.2 Charts Representation of Vehicle and Sensors Variables

The speed profile of the streets is very useful for a city manager to analyze with some detail the critical zones of a street with the help of a chart that shows the 25%, 50% (median) and 75% quarter speed of the vehicles (y axis) through the road distance (x axis). To complement the chart, there is a visual representation of the analyzed street in the map that represents a point position when hovering over a region of the chart (Figure 4.4). This type of interaction is helpful to show exactly on the map a critical speed zone observed in the chart traffic profile.

To show the statistics demonstrated before, the user can use some input selector to specify and restrict the data to show. Figure 4.5 shows the inputs the user can make to visualize the speed profile. From the user interface, it can be selected the road to analyze from a selection of available roads, the direction to view, the range of the hours of the day, the range of the minutes of an hour and the range of weekdays. After specifying all these inputs, the charts can be updated, added to comparison or saved.

After the user saved the selection made, he can later visualize the exact same visualization by selecting the desired visualization from the saved inputs. This is useful when the user finds a good visualization where an anomaly was founded. This anomaly can be later analyzed.

Figure 4.4: Traffic profile view of the Rua de Costa Cabral street in the morning of the weekdays.

4.4.3 Traffic Profile Comparison

Another helpful visualization is the comparison of the speed profile of a street in different moments (Figure 4.6). For example, a city manager may want to analyze the traffic in the morning versus afternoon or weekdays versus weekends to understand the traffic in the city in different occasions. Another example of the visualization is to compare the traffic before and after some public road works or traffic mobilization, and check if the change was good for the traffic or became worse.

After adding different moments of the same road to comparison, the user can visualize the speed profile of the street in two different dates and the correlation of this comparison. Again, the map complements this visualization with a point on the street when hovering over the comparison chart.

4.4.4 Representing environment information

An important aspect of a city is the possibility to analyze the pollution in the city and discover the main sources of that problem. One of the known sources, that is the major contributor of the pollution in the city, is the road traffic. To better understand the causes that traffic makes in the environment, it is created in the dashboard a visualization where some environment variables from multiple deployed sensors in the city can be compared and correlated to the average speed of the vehicles that go through nearby roads during the different hours of the day.

In figure 4.7 it is represented on the map several points that represent the different locations of the sensors deployed in the city. By selecting a specific sensor, the user can see

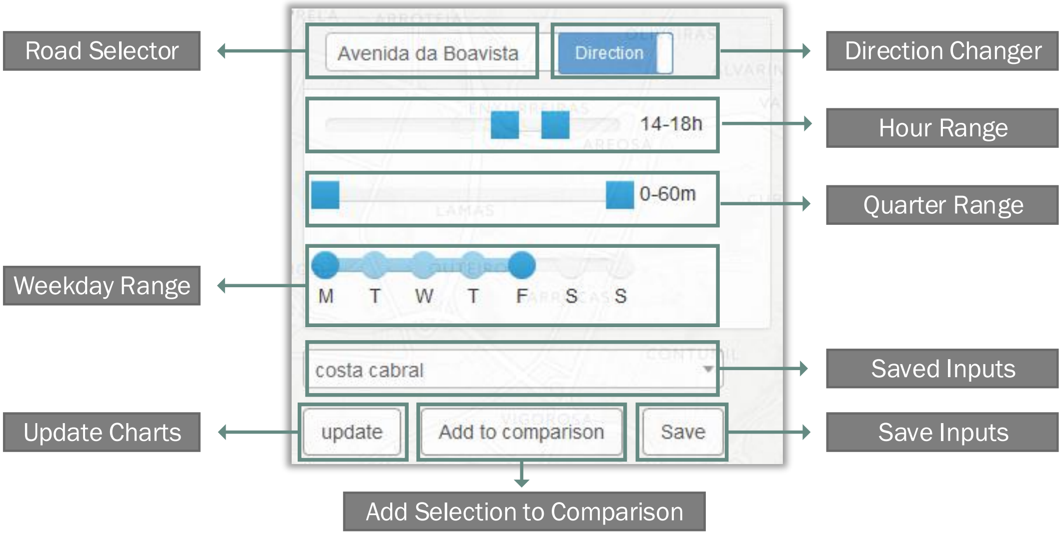


Figure 4.5: Inputs Selection for Different Searches.

---

details of the selected sensor. By choosing a specific week to analyze and the weekdays, we can visualize in a chart the levels of carbon monoxide of a selected sensor against the average speed of the nearby streets during the different hours of the day.

This type of visualization enables to recognize some patterns in the traffic and their effect on the pollution pattern to then understand what the critical hours of the day for a specific zone. By using this visualization we can see how the road traffic affects the environment, and also how the environment affects the traffic.

4.4.5 Road Definition and Transformation

It is also possible from the dashboard to manage the information to show on the statistics. Figure 4.8 shows the user interface used to add spatial road segments to be processed.

One of the problems with the OpenStreetMap dataset is that a single road can have multiple segments associated, which can be a problem to analyze small segments when the granularity of the data is low. It is created a visualization that can be used to associate all the segments of a specific road into one.

The user can select a road that can be composed by multiple segments, and select the ones that are needed for the road building. To help the process, a map visualization of the segments is shown, and the segments deselected from the list are also removed from the map. After choosing the ideal segments for the road, the user can the click to build the road.

After ending the building process, the dashboard presents a feedback to the user that refers if the building process is successful or not. Subsequently successfully road building, the selected road can now be used for the statistics processing and visualization.

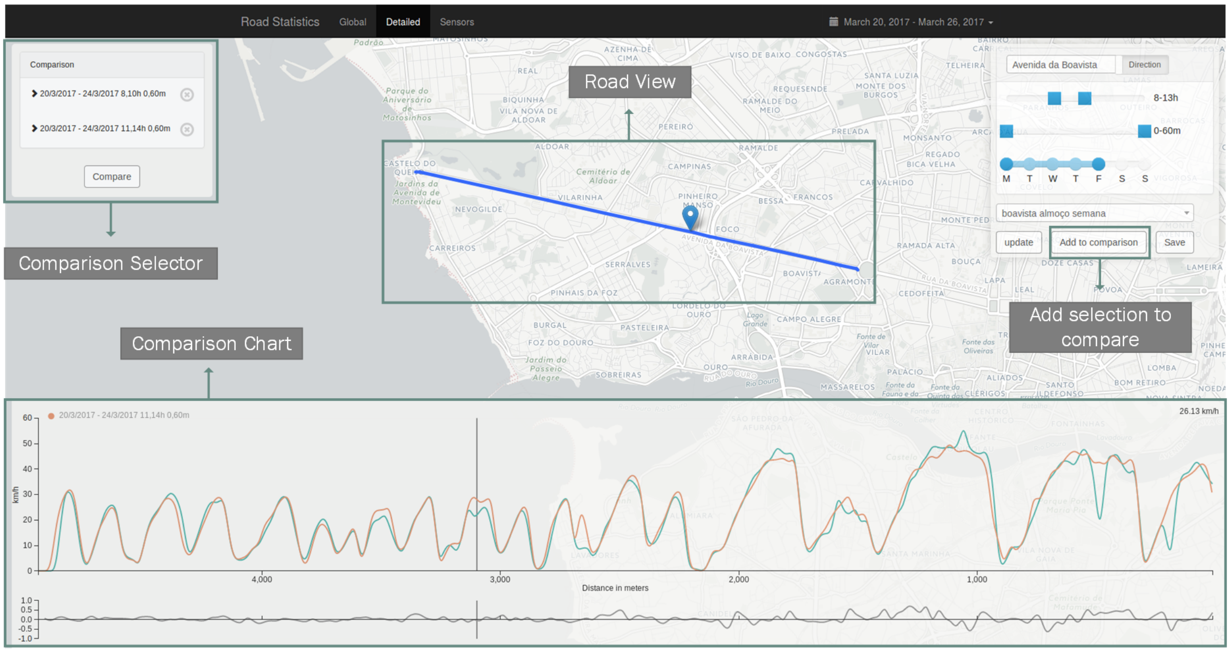


Figure 4.6: Traffic profile comparison of Avenida da Boavista street from different time instants.

4.4.6 Road Data Management View

To show statistics for the end user, the data needs first to go through processing algorithms as demonstrated in section 3.2, to then be ready to be presented in the dashboard. To simplify this process for a city manager, it was created in the dashboard an interface as shown in figure 4.9, where the user can hover by a list of roads that appear on the map, so the user can spatially see which road it is, to then select a specific road from that list of available ones that were created before. After selecting a road, the user can select a specific week to process the information, giving a command to start the road traffic algorithm where the road and date range are provided.

After starting the process, the script where the algorithm lies in provides the processing status that can be followed with the information of the week processing completeness percentage, the road name and the actual processing date.

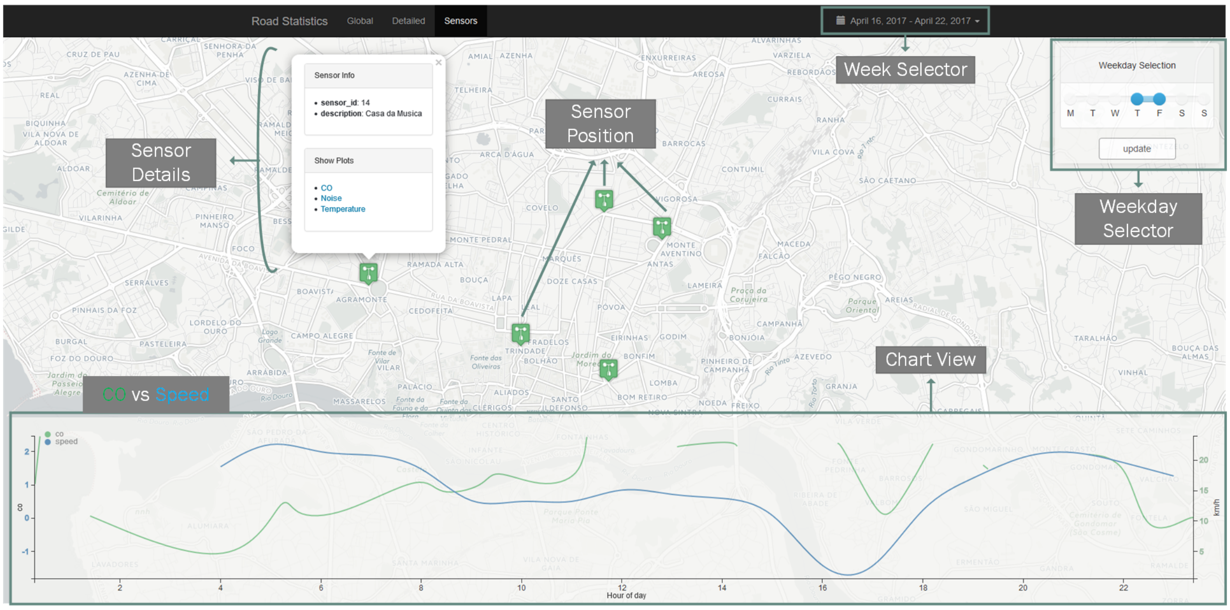


Figure 4.7: Visualization of multiple Environment Sensors in Porto city and the CO vs speed correlation chart.

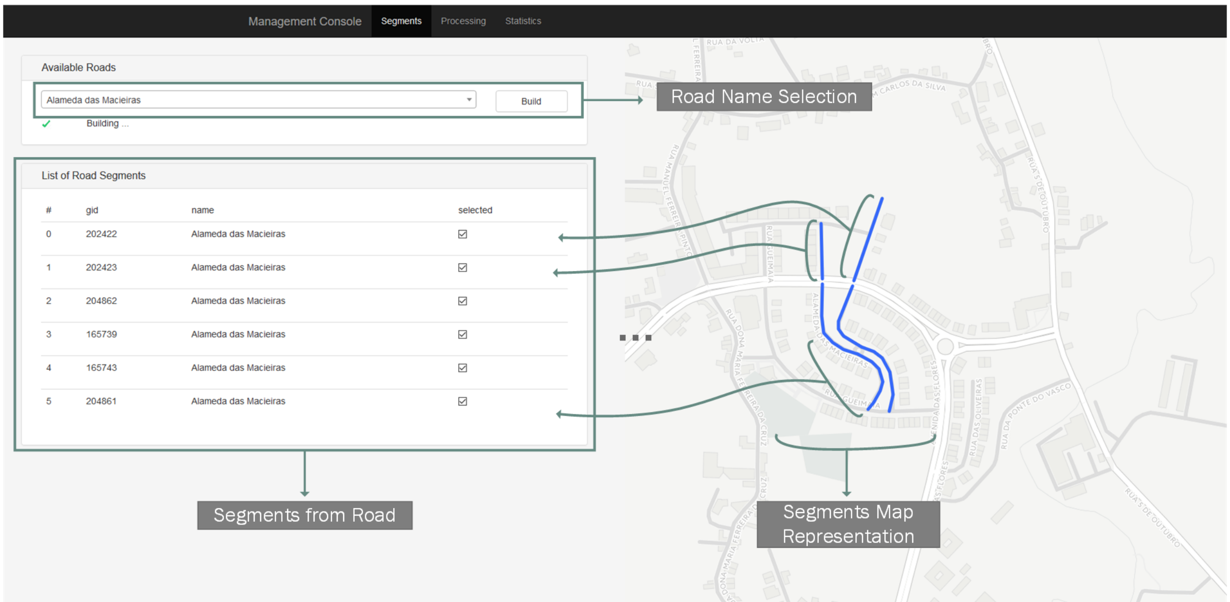


Figure 4.8: Dashboard interface to select and transform Road Segments.

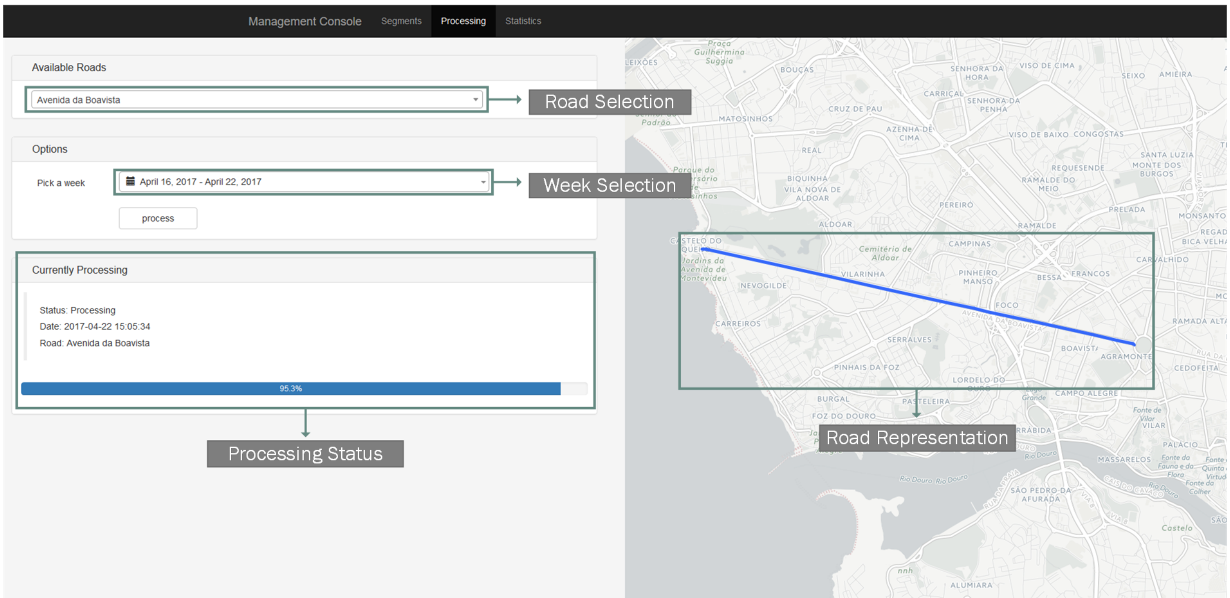


Figure 4.9: Dashboard interface for giving instructions and getting the status of the analysis of road information.

4.5 Summary

The global system has a full stack architecture from the source of the information to the presentation layer. For the management part of the system, the information is gathered from multiple sources of information for different types of information: the vehicular information, the road information and the environment information. This information then passes through analysis and processing algorithms before going to the storage system. Web services are a bridge that make the connection between the database and the dashboard application. Three main APIs are provided to manage the processing of data, provisioning of traffic profile data and provisioning of correlation from sensors and vehicles.

The storage solution is composed of spatial extension to support the geographic operations, functions for the logic inside the database and a data warehouse structure to support the dashboard as a decision support system.

Multiple services were implemented for the management of the information. The road traffic services provides information available from the database of the spatial roads, the profile comparison of roads at different instants, the inputs saved from the dashboard and the way of controlling the processing status. Another service is the traffic profile service which calculates and aggregates the pre-processed road points through a cumulative road distance, providing the speed of multiple points through the distance of the road generating a profile. In the environment area, there is a service that gathers information from the database from the environment sensors and road speed points from the nearby roads in the same perspective. Lastly, the service for the road building provides a way of creating a road from multiple existing segments.

The end part of the system and a very important one provides multiple views for the representation of the road traffic information and environment data. The dashboard provides a global view to show road traffic information on the map from different roads at the same time where different colors on the roads made the distinction of the speed of the traffic in their specific segment. Another visualization is the traffic profile where the user can select a specific road and date in a way to visualize the speed profile through the distance of the road. The comparison of two different occasions of the same road are another way of visualization that provides a way of the final user to compare events or changes on the road. The dashboard also displays environment information by correlating it with the road traffic information in the same chart. The management part of the dashboard provides a way of building road segments and the control of the road data processing.

Chapter 5

Implementation

5.1 Introduction

In this chapter it is presented the implementation options made when building the system. Section 5.2 describes the implementation options and technologies used for the information fetching from the multiple available sources. Section 5.3 describes the languages and frameworks used for the analysis algorithms and the system specification behind. In Section 5.4 it is demonstrated which database solution is chosen and its extensions, the database structure implemented for the solution and the functions implemented for the logic of the application on the database side. Finally, in section 5.3 it is presented the technologies used for the building of the dashboard solution.

5.2 Data Fetching

The source of vehicular information is located on a relational MySQL database from a remote server. For the information retrieval of vehicular information it is used a python script that uses a ODBC driver from the *mysqlclient* python library to perform a select query from the database that returns all the rows within the following constraints:

- Only needed time and spatial related attributes: it is selected only the GPS time, coordinates and vehicle ID.
- Segmenting within a time range: it is added a condition to retrieve the data from a limited time range, normally one hour each time.
- Ordering by vehicle and time: an ordering condition is implemented to retrieve a sequential vehicular navigation.

The environment information is available from a different source of data, and it is hosted in a remote REST web service. To retrieve the environment information it is used a python script that makes a HTTP GET request to the service. The data comes in a specific JSON format that is striped to get the relevant attributes.

5.3 Data Analysis

The analysis of the vehicular information is performed with an application in Python programming language that uses Flask[1] microframework as an endpoint for receiving processing instructions remotely. The application also uses a ODBC driver for the connection with the PostgreSQL database with the help of *psycopg2* python library.

The Environment information processing application is also implemented in Python programming language with the use of *psycopg2* python library to support the persistence of information in the database.

The Applications are hosted in the same virtual machine, with 2 core CPU, 1GB of memory RAM and with Ubuntu 14.04 LTS Operating System.

5.4 Databases and Data-warehouse

The database plays an important role in the global application: it is where most of the processed information is stored to reduce the time it takes to show information on the web application. It is also on the database where the majority of spatial transformations take place with some customized PostGIS functions incorporated as a PostgresSQL extension.

For this work, the database chosen is the PostgreSQL Database Management System. Because this work is mainly focused on spatial analysis and operations, it is chosen the PostgreSQL Server since it has a very good support of Geospatial storage and functionality through the available PostGIS plugin. PostGIS is a spatial extender for PostgreSQL database as it adds support for geographic objects, integrating spatial functions and geometry data types into the database, abstracting all the spatial calculations.

In the database it is implemented a Data warehouse structure using a star schema approach separated by dimension tables (Segment, Date and Sensor) and fact tables (Road Data and Sensors Data), as it is depicted in Figure 5.1.

The structure of the data warehouse is shown in subsection 5.4.1, where it is presented the existing tables and attributes, their overall organization and interactions.

In the Spatial Operations subsection it is presented the main spatial functions used from the database and how they work.

In the subsection 5.4.2 it is analyzed the implemented functions inside the database that help its integration with the interacting applications.

5.4.1 Structure

For the structure of the database it is necessary an efficient way to store the large amount of information that comes from the buses. Because every active bus generates a log every 15 seconds, and there are around 400 buses in the city, this gives around 1600 logs every minute. After processed, this large amount of data needs to be stored in an efficient way, and the data warehouse structure enables to split the data in dimension and

[1]http://flask.pocoo.org/

fact tables to reduce the redundancy of the data. Using the dimension tables, the data inside is unique and can be used multiple times from the fact tables.

There are two fact tables: the Sensor Data and the Road Data that holds the measurements and metrics of the captured data. In the Sensor Data table these measurements can be the captured environment measurements like the temperature or the CO2. The Road Data table can keep information of the position of a vehicle.

The remaining tables are the dimension tables. The date table has information about date and time in different formats, where the date and time units are taken apart. The Sensor table has information about the existing sensors like the name or ID. The Segment table has information about the different road segments, and usually contains the name or location. The detailed description of the tables is the following:

- `Segment`: In this table it is stored the road data from OpenStreetMap.

  In the attributes it has the **gid**, which is an unique identifier of the road segment, the **name** of the segment, the **geom** that holds the spatial representation of the line segment and the **buffer** which has the same segment as the geom attribute, but a bit larger because of the implemented buffer from the original geometry.

- `Date`: This table holds the time with a very low granularity for the events recorded in the fact tables. The time is segmented in different attributes.

  From the datetime value of a new event record, it is extracted the **year, month, day, hour, quarter of the hour and weekday**.

- `Sensor`: this table contains the sensors technical information.

  It has an **id** for an unique identifier, the **type** of the sensor, the **description** that normally has the name of the location of the sensor, the **latitude** and **longitude** for the position of the sensor, and the **geom** which is also the position of the sensor but converted to the geometry type.

- `Road Data`: this is the table that contains the majority of the vehicle data. Here it is also stored the metrics of the processed information calculated previously.

  The **node id** is an unique identification of a vehicle.

  The **road id** is the foreign key from the table Segment. This relationship is necessary to have an association of the metrics to a specific segment.

  The **date id** is a foreign key to the Date table, so all the data has a time associated. This relationship will help when retrieving information segmented by specific instances of time.

  The **course** and SEQUENCE are counters, where the course is the number of each pass a vehicle makes in a segment, and the sequence is the number of the point in that segment.

  The **speed** is a variable of the speed of the vehicle from the previous point, using the distance and delay for this calculation.

The **distance** is the the distance in meters that the vehicle moved from the previous point.

The **fraction** is the location of this point in the segment associated. So with this value and the segment, no coordinates are necessary to locate a point.

The **direction** is the direction of the route associated with this point in the segment.

- `Sensor Data`: table that stores the metrics of the environment sensors. The attributes are the following:

  The **seq** variable is the sequence number of the message retrieved from the associated sensor.

  The **date id** is the foreign key associated with the table Date to relate the sensors metrics with time.

  The **sensor id** is a foreign key to associate the metrics with a specific sensor.

  The **temperature**, **noise** and **co** are metrics from captured environment values from sensors deployed in the city.

There are some dependencies between the tables, and normally only the dimension tables connect to the fact tables, not on the same type. The same applies to the fact tables.

Here both the Sensor Data table and the Road Data table have a connection to the Date table because they both have their data associated with time. This is necessary to later make requests of data based on a specific date.

The Sensor Data table has a connection to the Sensor table because it needs to have additional information about the sensors that are not available in the fact table.

The Road Data table connects to the Segment table to be possible to link the information of the vehicles to a specific road segment.

With this data warehouse structure, it is possible to query information from any moment at different locations. Questions like "What is the speed of the vehicles on Mondays at Avenida da Boavista" are now possible by restricting the data by the Segment and by the date where the weekday variable is available in the table Date.

5.4.2 Functions

Specific functions are built on the database side to shift some of the application logic to the database. The use of these functions will help to divide and simplify the processing algorithm, because the main part of the logic will be in the Python application that will call the specific functions when needed. These functions have some logic more related with database operations, so it makes sense to be closer to the database as it will be faster.

The majority of these functions are used by the vehicle and sensor processing algorithm and also from the web services. This is an advantage as multiple applications and services can use the same functions if necessary.

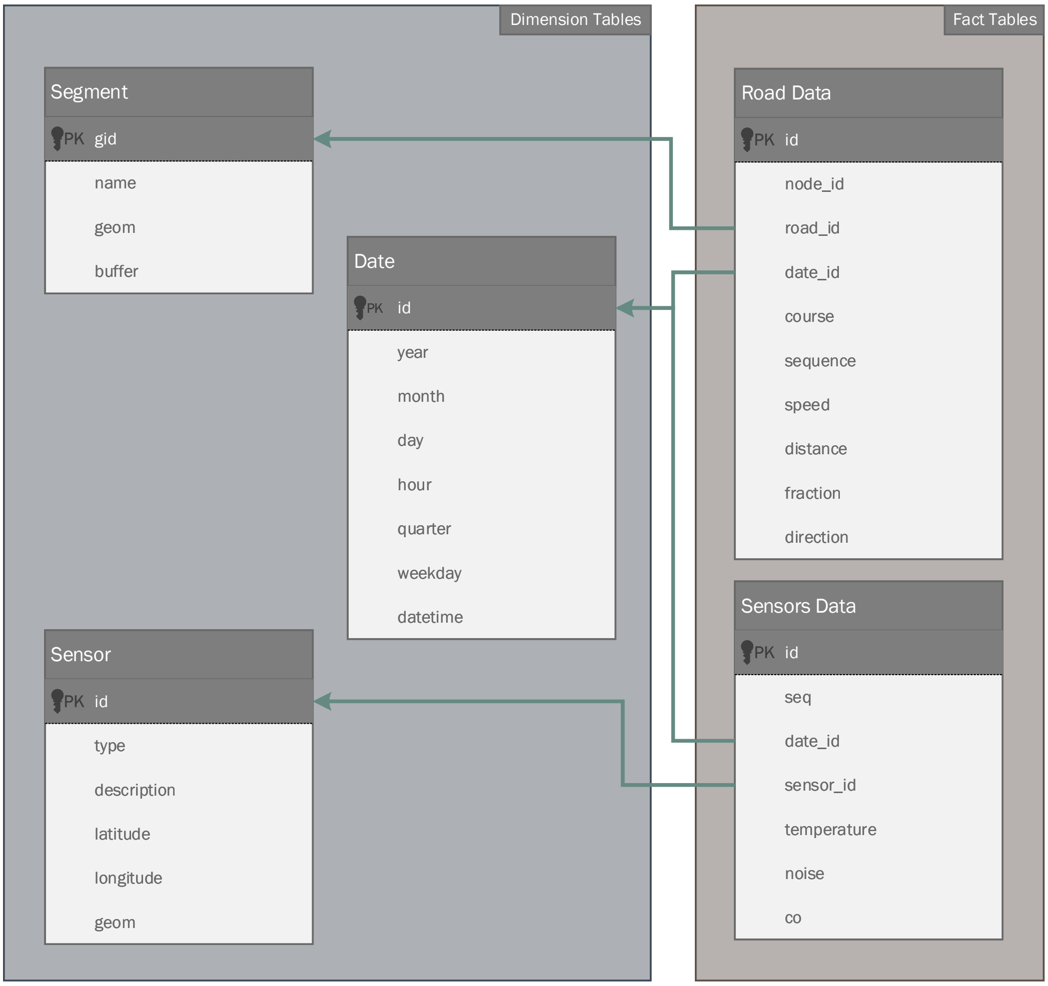


Figure 5.1: Data warehouse structure.

The functions created serve as support for the spatial operations used for the road traffic algorithm, but also for the management and storage format of the data warehouse when some persistence needs to be made. These functions are the following:

- `get closest road`: gets the road that the vehicle intersects.
  - `input`: latitude, longitude.
  - `spatial operations`: distance, intersection, line locate point, make point.
  - `behaviour`: from the coordinates, builds the geometry point used to: first find the road segment that intersects the point to finally get the ID of road and the location inside the road.
  - `output`: road id, road fraction.

- `get date id`: gets the associated ID from a *timestamp* in date table, creating new if non-existent.
  - `input`: *timestamp*.
  - `spatial operations`: not used.
  - `behaviour`: as figure 5.2 shows, from the *timestamp* received as input, it is detached and retrieved the year, month, day, hour, quarter of hour, weekday, week of month. If this *timestamp* does not exist in the table, then a new entry is inserted with the previous variables. Finally, it is retrieved the ID from the entry.
  - `output`: date id.

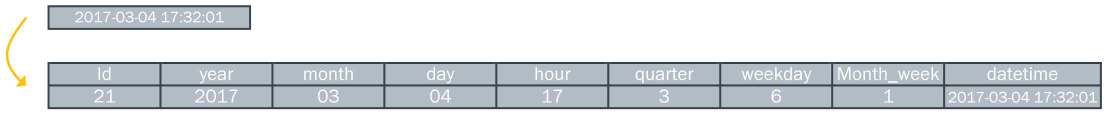


Figure 5.2: Transformation of date-time to dimension table structure of data warehouse.

- `insert detailed road data`: inserts processed road data into the database.
  - `input`: date id, road id, node id, course, sequence, speed, distance, fraction, direction.
  - `spatial operations`: not used.
  - `behaviour`: inserts a new entry in the Road Data table with the selected inputs.
  - `output`: nothing.
- `insert sensor`: creates a new sensor in the database.
  - `input`: sensor id, type, latitude, longitude.
  - `spatial operations`: Make Point.
  - `behaviour`: adds a new entry in Sensor table if the sensor does not exist, creating also a spatial geometry point from the coordinates.
  - `output`: nothing.
- `insert sensor data`: inserts processed sensor data into the database.
  - `input`: sequence, date id, sensor id, delay, luminosity, temperature, humidity, precipitation, wind speed, no2, noise, o3, co, minute.
  - `spatial operations`: not used.
  - `behaviour`: adds a new entry (if not existent) in the Sensors Data table from different sensor variables.

  - `output`: entry id.

- `segment union`: creates a new segment from existing ones.
  - `input`: list of ids from road segments, name for new road.
  - `spatial operations`: union, buffer, line locate point.
  - `behaviour`: executes an union operation from all the IDs of road segments received as input, creating a new geometry. Inserts in Segment table the lowest ID from the list, the input name, and the created segment. Creates a buffer for the created segment. Tests if the segment is valid by using line locate point operating, which gives an error if the segment is not valid, as an indication to remove it.
  - `output`: segment id.

- `insert inputs`: inserts into the database the selected inputs from the dashboard.
  - `input`: selection name, initial date, end date, minimum hour, maximum hour, minimum minute, maximum minute, minimum weekday, maximum weekday, road name, direction.
  - `spatial operations`: not used.
  - `behaviour`: adds a new entry in the inputs table with the inputs created in the dashboard.
  - `output`: nothing.

5.5 Data Visualization

The Dashboard is the visible part of the system to the end user. It is implemented as a web application based dashboard with analytic functionality where the focus is to have multiple visual dimensions to see traffic information. One example is the geographical representation of roads and positions on top of a map layer where it relies on the data provided by the data warehouse API, namely traffic and sensor information.

In the web application it is used HTML and CSS to show static content, and JavaScript[2] programming language and JQuery[3] library for the application logic. From the JavaScript programming logic side there are several implemented main components:

- `Web service client`: Ajax rest requests are used to get data from the web services to be represented in the dashboard.

- `Chart Tools`: d3.js is a library used to represent a more analytic and precise chart representation of road traffic and environment sensors data.

[2]https://www.javascript.com/

[3]https://jquery.com/

- `Map Tools`: leaflet, a GIS JavaScript library used to show maps and represent all geographical data representation of roads, positions, polygons and other overlays.

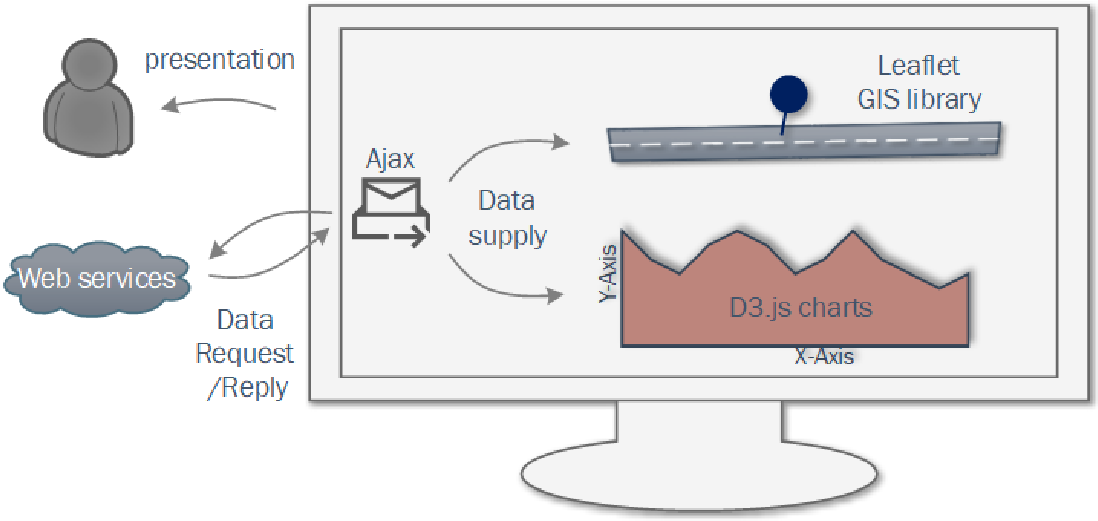


Figure 5.3: Main visualization components from the dashboard implementation.

These are the three main tools used to show all the information to the final user, from the information fetching from web services to the visual mapping representation and statistics with charts in the same web application.

Other components such as Bootstrap[4] library are also used, but they are not crucial for the application to work. They just helps the development and make the user interface more friendly.

5.6 Summary

This chapter presented the implementation aspects of the system. The fetching of information is made using database drivers on the python application to query the mobility MySQL database, and HTTP requests to request environment information from a remote REST service.

The data analysis is implemented with the Python programming language with Flask framework for the web service and psycopg2 python library for the persistence of information.

The data warehouse has an implemented star schema structure with fact tables to save road data and environment data, and dimension tables to store the time, road segments and sensor details. The implemented functions help on the interactions of the applications with the database, simplifying the spatial operations and improving the performance because of its proximity with the database system.

[4] http://getbootstrap.com/

The data visualization, or in other words, the dashboard uses three main components for the representation of information, the d3.js javascript library for the chart representation, the Ajax for making HTTP requests to the web services, and the leaflet.js javacript library for the map representation.

Chapter 6

Case Studies

6.1 Introduction

In this chapter we present four case studies to illustrate the usefulness of our system when making road analysis: the first presents a case study that compares how the traffic is affected in different hours of the same road and day. The second case study is the analysis of a road speed profile results to make a discovery of a critical zone with low traffic flow. The third case study shows the comparison of the same road during the days of an event and the days after the event, in a way to understand the difference a big event makes in road traffic on the different segments of the road. Finally, in the last case study it is compared the results of correlating the pollution levels of a environment sensor with the road traffic information of the nearby roads.

6.2 Traffic in different hours of weekday

A good example when analyzing road traffic information is to compare it from different hours of the day to identify the most affected segments of the road during the rush hours.

We have picked a one-way direction road to compare the road speed profile in the middle of the day, when the traffic is supposed to be low against the end of the afternoon, when the people leave their works and the speed of the traffic is more affected. It was used the following parameters for this example:

- **Street**: Rua da Constituição
- **Week**: 4 - 10 September 2017
- **hour**: 11-15h and 16h-19h
- **weekdays**: Monday, Tuesday, Wednesday, Thursday, Friday
- **direction**: right to left (one-way)

The Rua da Constituição street is chosen for this study because it is a road in the center of the Porto city with many secondary road intersections that can have large impact in certain hours. Also as the road is only one way, it is easier to identify the source of the problem, because it has less variables affecting the traveling of the cars, for example a car trying to turn to the left on a two way direction road. The selected week is a normal week where the majority of the people are not on holidays. It is used the hours when most of the people are working against the hours when the people leave their works. The days of the week are the chosen days to simulate the working days.

The result that can be observed in figure 6.1 shows that the majority of the road is not much affected by the time of the day, although, in a particular segment of the road, the one where the speed of the vehicles is usually low, in the end of the afternoon the road traffic speed is lower as a direct cause of the traffic density that makes it difficult for vehicles to have a bigger acceleration and speed between stop lights. One possible solution to improve the traffic is to have less red light duration in this zone at certain critical hours to accumulate less traffic.

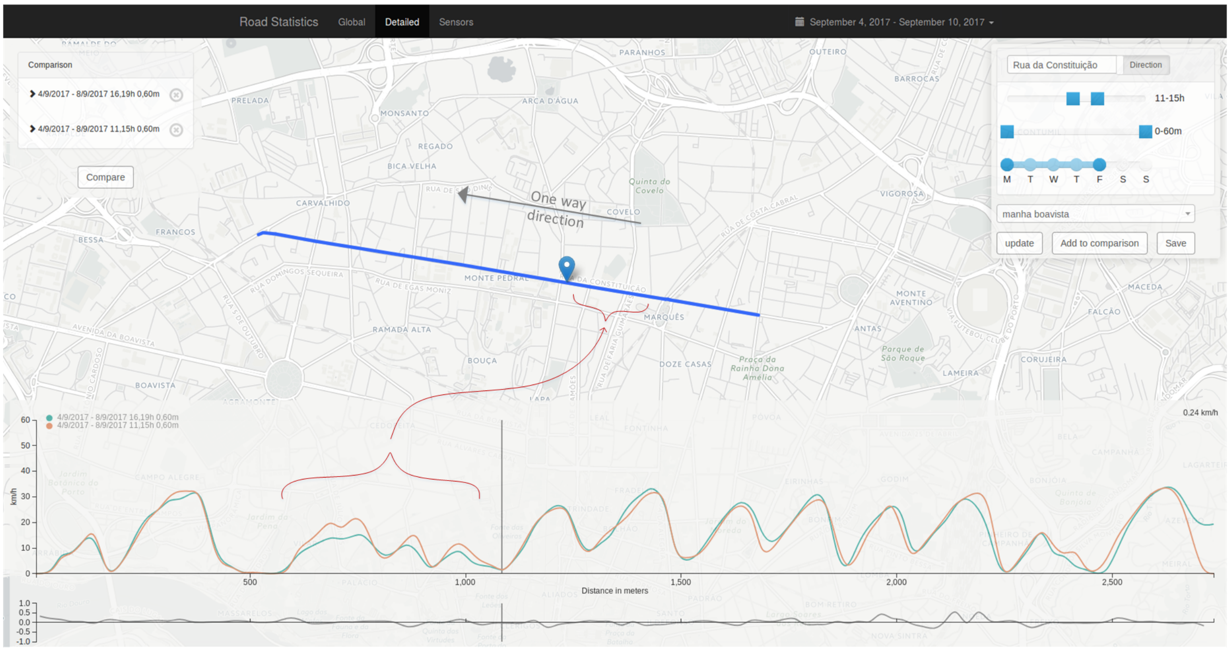


Figure 6.1: Rua da Constituição street in Porto spatial representation and its speed profile comparison of mid-day and end-day hours of a typical week.

6.3 Discovering a Critical Zone

One of the main objectives of this work is to identify critical zones where the speed of the vehicles is normally lower than other zones nearby. For a city manager this helps to determine exactly the source of the problem, where it begins and where it ends.

In order to identify some problems in the streets we pick a specific street in a specific date. Figure 6.2 contains the speed profile of a street using the following parameters:

- **Street**: Avenida da Boavista
- **Week**: 20 - 26 March 2017
- **hour**: 8-9h
- **weekdays**: Monday, Tuesday, Wednesday, Thursday, Friday
- **direction**: right to left

We choose these parameters for the following reasons. Avenida da Boavista street is a good choice because it is a long street with an extension of 5 kilometers with many of traffic lights and intersections, and it is also one of the streets with a large amount of traffic concentrated. The week is not chosen with any special purpose, it is just a normal week. The choice of the picked hours correspond to more traffic on the roads as the people typically go to work on those hours. The weekdays are normally days with more congestion. The direction is chosen to study the traffic that goes to the center.

By choosing these specific parameters, as figure 6.2 shows, we can observe the speed profile through the distance of the street. In the profile, we can see that the speed is not constant along the street, mainly because of the intersections, street lights or bus stops. As seen in the speed profile, there is one particular zone where the median speed of the vehicles is almost zero and the 75% speed does not get much higher, which means that almost all the vehicles stops at that zone or travel at a low speed.

To better understand the issue detected in the speed profile in the figure 6.2, with the help of a dashboard feature, we select a specific zone in the traffic profile chart and zoom it to provide a larger data granularity and better detail. As figure 6.3 shows, the critical zone identified before is zoomed and it is identified with more detail the source of the problem. It is an intersection with traffic lights and strong traffic. From the speed profile we can identify the high deceleration of the vehicles, even before they reach the intersection, which means that normally a queue of cars could be accumulated in the traffic lights, and the slow acceleration at the end confirms that hypothesis.

Going for a more real environment level, and using the street view from Google Maps [1] online tool on the same identified road intersection, in figure 6.4 we can confirm the following facts:

- The street has already multiple lanes which helps the traffic flow.
- Traffic lights are the solution for the management of the intersection.
- Multiple vehicles from all directions are stopped in the intersection, which means that the timing in the lights could be improved.

[1] https://www.google.pt/maps

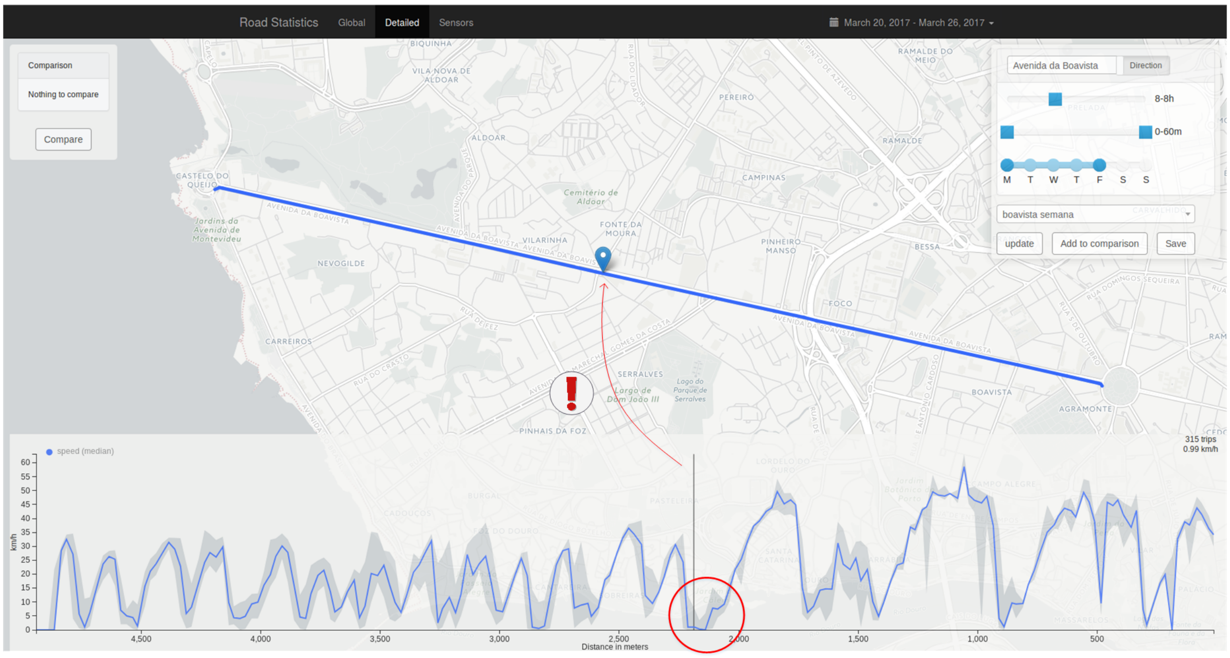


Figure 6.2: The Porto street of Avenida da Boavista with a speed profile during a typical week, in the morning, between 8 and 9 o'clock.

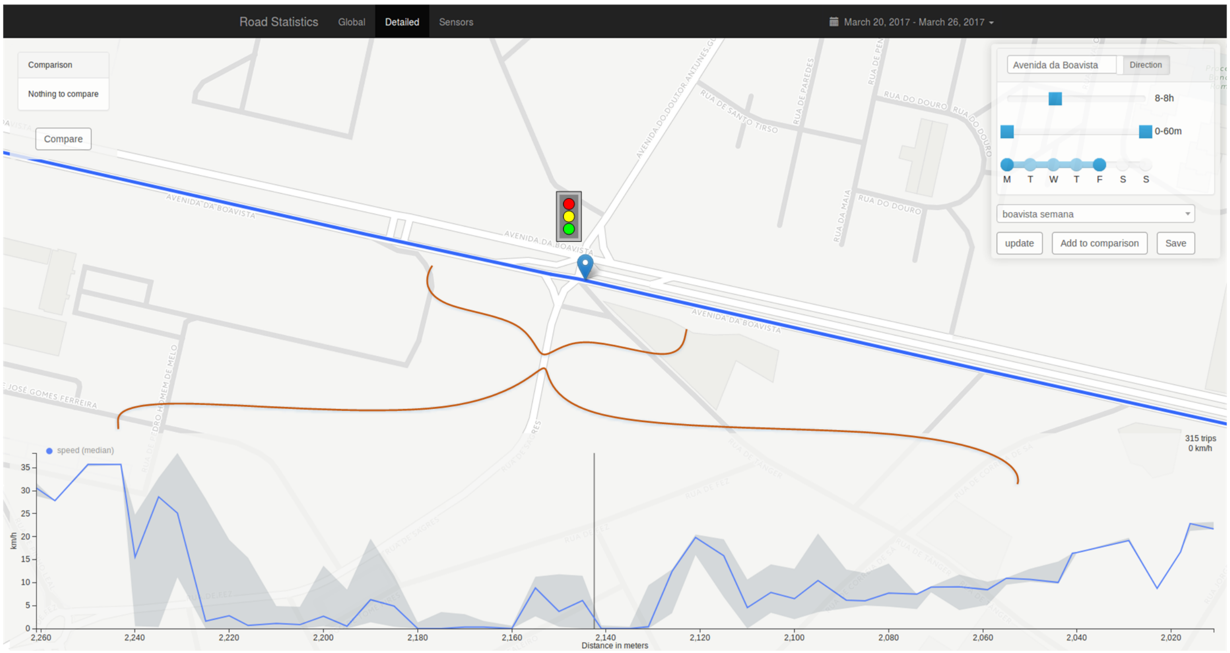


Figure 6.3: Speed profile of a section of the street of Avenida da Boavista in Porto during a typical week in the morning, between 8 and 9 o'clock.

One of the possible changes to be made can be the synchronization of the traffic lights. Also it can be given more time on green light for the main road, where it has more vehicles passing. These are easy changes to be implemented since they do not need construction works, and a trial and error solution can be made to analyze the real benefits made from the changes. These changes made can be analyzed with the same traffic profile chart but with a different date, to understand what are the consequences of those changes in the speed profile of the road, resulting in an improvement or not of the traffic flow.

Figure 6.4: Street view of the intersection of Avenida Antunes Guimarães with Avenida da Boavista road.

6.4 A special event in Porto

Another helpful visualization is the comparison of the speed profile of a street in different moments. For example, a city manager may want to analyze the traffic in the morning versus afternoon, or weekdays versus weekends to understand the traffic in the city in different occasions. Another example of the visualization is to compare the traffic before and after some public road works or traffic mobilization, and check if the change was good for the traffic or became worse.

A good use case is trying to find outliers and different patterns when comparing a normal day against a day with special events. Picking a big event to analyze is a way to see how the roads handle an abnormal amount of traffic. As an example of the event in figure 6.5, it covers a big area of the city, which is good to analyze the surrounding traffic near the event. After searching events in Porto, we came up with the following:

- Event

- **Name**: Red Bull Air Race [49]
- **Year**: 2017
- **Days**: 1, 2, 3 of September
- **Location**: Douro river surroundings

• Dashboard Inputs

- **Street**: Avenida da República
- **Week**: 28 of August - 3 September 2017 and 4 - 10 September 2017
- **hour**: 13-23h
- **weekdays**: Saturday, Sunday
- **direction**: Below to Top

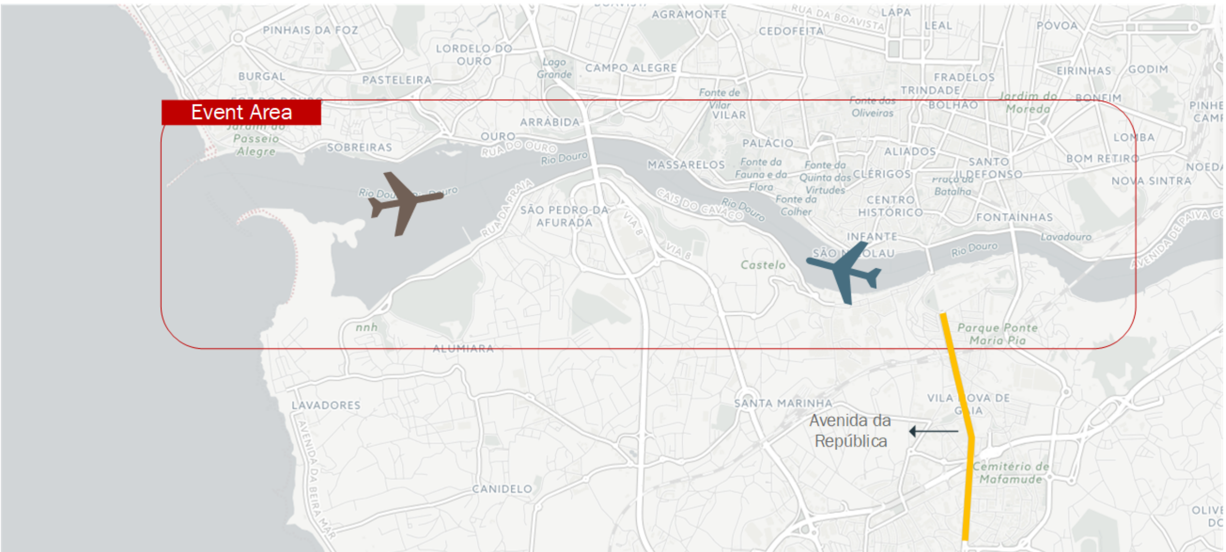


Figure 6.5: Area of Red Bull Air race event in Porto with the selected Avenida da República road to analyze.

The Avenida da República street is chosen because it is one of the streets where the traffic is not closed during the event, but it also gives access to one of the main areas of the event. The picked weeks are composed by the event week and the week after the event for comparison purposes. The range of hours for the statistics are in the afternoon and evening because in the afternoon there is more concentration of people, and in the evening people leave the event. The days of the weekend were chosen, as they are the main days of the event. The chosen direction of the vehicles is from below to top of the street because it is the way to reach the event.

After choosing the two different weekends to analyze, one of the event and the other in the weekend after the event, when nothing not normal happened, we found interesting results. As shown in figure 6.6, it is represented a chart with two speed profiles of the street where the green line is the one from the event date. In the beginning of the road segment, the speed of the traffic is lower in the event weekend with less 5-10 km/h against a normal weekend, mainly because of the accesses in the beginning of the street. In the end / top of the street we see a similar difference in the traffic speed when compared against the normal weekend. In this case, the section is closer to the event area, which means that it is probably affected by the people trying to park their cars or people crossing the street by walking.

This is a good case study for a city manager to analyze and make traffic changes at the beginning and end of the segment when hosting similar events in the area nearby. Something like divert the traffic to secondary roads or allow only public transportation to drive on that road are choices that can be analyzed by a city manager.

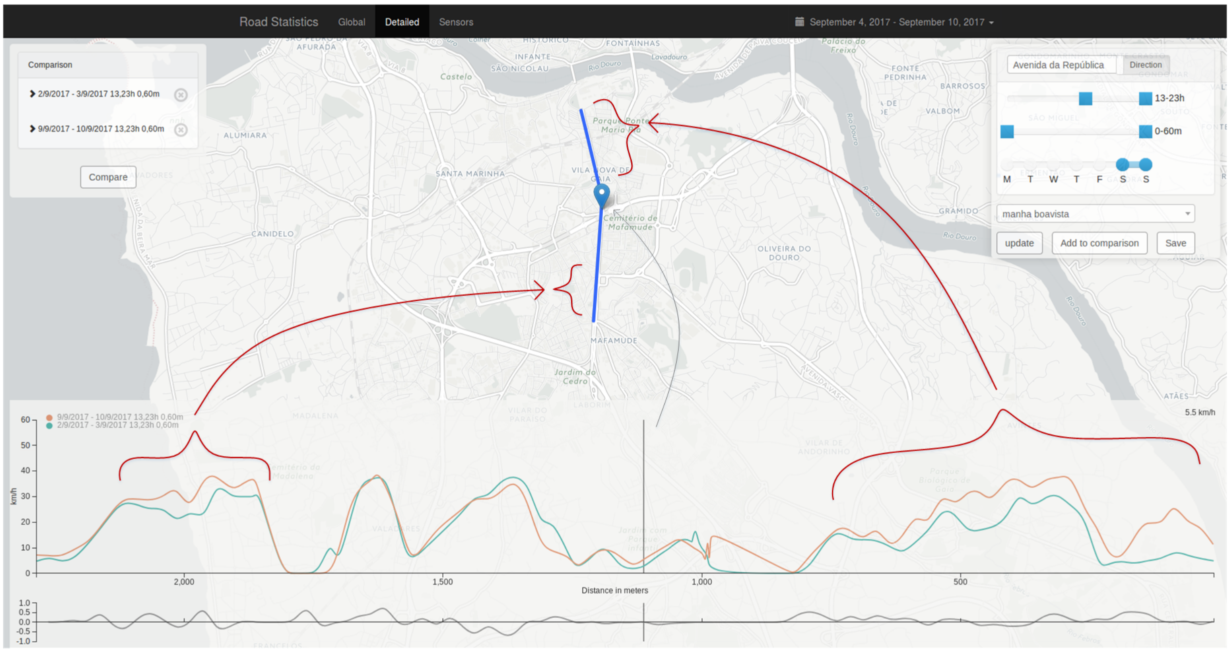


Figure 6.6: Avenida da República road traffic profile comparison in two different time instants, one of the events days and the other in the weekend after the event.

6.5 How the road traffic affects the environment

A good case study for a smart city use to improve its general quality is using traffic information and environment information to understand what are the relations that can be retrieved from them, trying to make conclusions on how the traffic affects the local environment and how the environment affects the traffic. This is a feature that is also present in the dashboard. It was chosen a specific sensor in a date range as follows:

- **location**: Casa da Musica

- **Week**: 16 - 22 April 2017
- **weekdays**: Thursday, Friday
- **Metrics**: Carbon Monoxide (CO), Speed

The location chosen is a good option because it is close to Avenida da Boavista street, where many cars pass by, and that street has a good amount of traffic information. The week is just a normal one with the chosen days being in the end of the week. The preferred metrics to analyze are the speed of the vehicles from the nearby streets and the carbon monoxide levels from the sensor position.

As figure 6.7 shows, we have selected the specific sensor to analyze the result of correlating the traffic and a pollution metric (carbon monoxide, CO). As the chart shows, the result is something already expected: the speed of the traffic affects the pollution index. In the beginning of the day, the average speed of the buses and remaining vehicles is higher and the CO levels are low, but as the speed starts to get lower, the CO level increases. From the results, we can also observe the hours of the day more affected in the sensors area, motivating on making changes in the traffic at certain hours of the day. This also allows to see how the changes made in the traffic affect the pollution levels in the area, if they get better or worse.

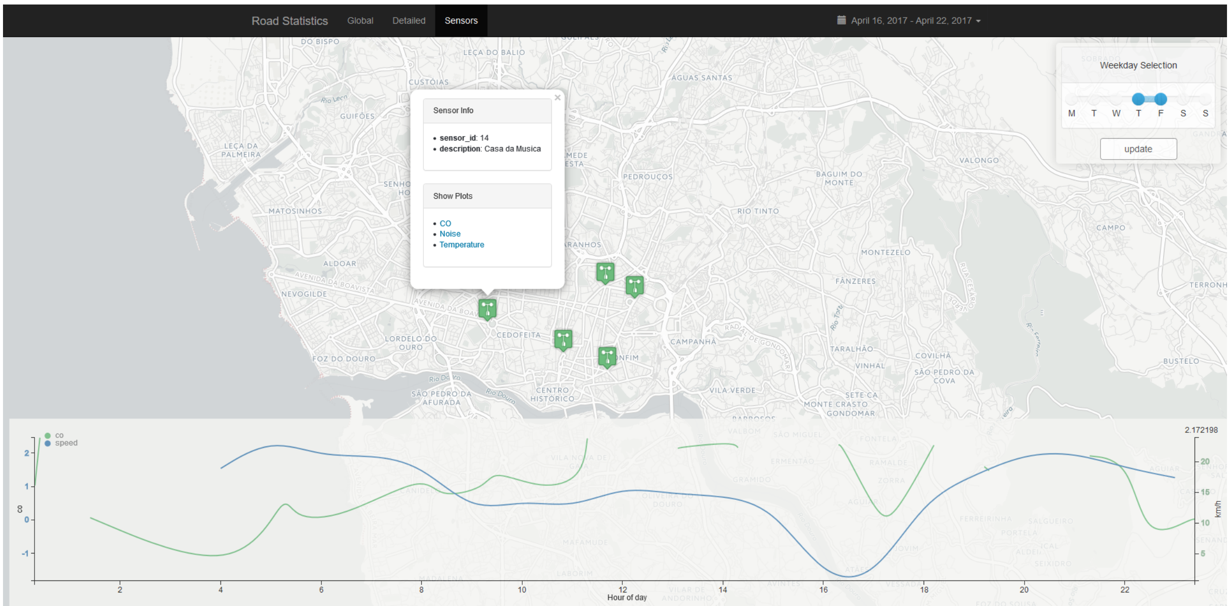


Figure 6.7: Correlation of traffic speed vs CO values through the day from the environment sensor deployed in Casa da Musica, Porto.

6.6 Summary

Multiple case studies were tested to show the effectiveness when using the dashboard to detect anomalies and outliers in different instants, events or places.

One of the cases that can be used is the comparison of different moments of the day at a chosen street to find certain segments of the road where the speed profile can diverge. In the example used, a different pattern was found in a segment of the road, when comparing the speed profile from the middle of the day against the end of the afternoon.

Another use case exemplified in this chapter is the discovery of a critical zone of a certain road. Based on the speed profile of the road, it can be identified a zone of the street where the speed of the majority of the vehicles is close to zero. When zooming on the zone, more details can be gathered and the source of the problem more easily detected.

Big events in the city bring a lot of people to the same area and therefore the road traffic is affected by the flow of vehicles. The Red Bull Air race event in Porto city was the example used to study how an event can make changes on the flow of traffic. For the study, it was chosen a road where the traffic flows to the event area during the days of the event to compare with the same weekdays in the week after the event. The comparison of the profiles show that there are some differences on the speed profile but only in certain zones of the road, the ones that can be improved.

In the last case study it was used environment sensors in the city to verify their relation with the road traffic speed. The objective here was to verify if the speed of the vehicles on the roads near the sensors can make variations on the pollution levels measured. In the example used, it was selected the measured CO values of an environment sensor to compare with the speed values of streets nearby through the different hours of the day. The results show that the speed of the vehicles has a direct effect on the carbon monoxide values.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

With the smart cities demands in involving technologies for smart sensing that can be used to develop smart decisions, we proposed a system dashboard as a solution for abstracting decision makers / city stakeholders from the actual city smart city infrastructure.

We developed a full stack solution to provide a way to analyze mobility patterns from the bus network of the city, to then build statistics focused not so much on the vehicles, but mainly on the roads and stationary sensors. From an available source of real work mobility and environment data, algorithms were used to fetch this valuable information for analysis and transformation in order to obtain road traffic parameters assigned for the roads. A data warehouse storage solution was the choice to feed the dashboard with segmented information for the needs of a decision support system.

Using GIS technologies and applications, the spatial operations on vehicle positions and road identification were made with more precision to provide faster results with the less imprecision, but also to help on the geographic visualization.

The dashboard provides the ability to visualize information of the city traffic from local to city wide scale. We consider that it can be a very useful tool for a city manager, namely in supporting conclusions on how the city is moving, improving it in a way to provide better behavior of the traffic in the city.

The typical scenarios presented in the case studies illustrate how the proposed system can be useful to identify and locate anomalies and unexpected patterns on the roads. The results were conclusive and real effectiveness on using the dashboard for road traffic analysis shows the advantages for a city manager to do better urban planning and traffic management.

7.2 Future Work

The presented dashboard provides a good solution for the road analysis of a smart city, but some aspects could be improved or developed to give more intelligence and reliability of the platform:

- End-user validation: it is a component that is lacking in the system, but it is planned to make available to the city managers and bus company for assessing its usefulness and gather their inputs.
- More data: including more mobility information not only form the buses, but also from other types of vehicles, such as taxis or bicycles, will provide a more wide coverage of the city and more data precision.
- More correlation: adding more variables to correlate the areas of mobility, environment, specific events and public services will bring more knowledge to the city.
- Traffic prediction: using machine learning to add the ability to predict and/or, at least, estimate the probability of atypical events occurrence that may impact the city.

Bibliography

[1] R. E. Hall, B. Bowerman, J. Braverman, J. Taylor, and H. Todosow, "The vision of a smart city", *2nd International Life ...*, vol. 28, p. 7, 2000, ISSN: 02642751. DOI: 10.1017/CBO9781107415324.004. arXiv: arXiv:1011.1669v3. [Online]. Available: ftp://24.139.223.85/Public/Tesis%7B%5C_%7D2011/Paper%7B%5C_%7DCorrection%7B%5C_%7D4-15-09/smartycitypaperpdf.pdf.

[2] R. G. Hollands, "Will the real smart city please stand up? Intelligent, progressive or entrepreneurial?", *City*, vol. 12, no. 3, pp. 303–320, Dec. 2008, ISSN: 13604813. DOI: 10.1080/13604810802479126. arXiv: arXiv:1011.1669v3. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/13604810802479126.

[3] I. a. Harrison, C. and Donnelly, "A Theory of Smart Cities", *Proceedings of the 55th Annual Meeting of the ISSS - 2011, Hull, UK*, no. Proceedings of the 55th Annual Meeting of the ISSS, pp. 1–15, 2011, ISSN: 1999-6918. DOI: 10.1017/CBO9781107415324.004. arXiv: arXiv:1011.1669v3. [Online]. Available: http://journals.isss.org/index.php/proceedings55th/article/viewFile/1703/572.

[4] T. Nam and T. A. Pardo, "Conceptualizing smart city with dimensions of technology, people, and institutions", in *Proceedings of the 12th Annual International Digital Government Research Conference on Digital Government Innovation in Challenging Times - dg.o '11*, New York, New York, USA: ACM Press, 2011, p. 282, ISBN: 9781450307628. DOI: 10.1145/2037556.2037602. arXiv: arXiv:1011.1669v3. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2037556.2037602.

[5] H. Merry, *Population increase and the smart city - Internet of Things blog*, 2017. [Online]. Available: https://www.ibm.com/blogs/internet-of-things/increased-population-smart-city/ (visited on 10/22/2017).

[6] D. Washburn and U. Sindhu, "Helping CIOs Understand "Smart City" Initiatives", *Growth*, p. 17, 2009.

[7] SUPERGEO, *The Smart City brought to you by SuperGIS 3D Earth Server*, 2016. [Online]. Available: http://www.supergeotek.com/Library_1_201402.aspx (visited on 10/22/2017).

[8] T. Bakici, E. Almirall, and J. Wareham, "A Smart City Initiative: The Case of Barcelona", *Journal of the Knowledge Economy*, vol. 4, no. 2, pp. 135–148, Jun. 2013, ISSN: 18687873. DOI: 10.1007/s13132-012-0084-9. arXiv: arXiv:1307.8198v1. [Online]. Available: http://link.springer.com/10.1007/s13132-012-0084-9.

[9] A. McAfee, E. Brynjolfsson, T. H. Davenport, D. Patil, and D. Barton, "Big data: the management revolution.", *Harvard business review*, vol. 90, no. 10, pp. 61–67, 2012, ISSN: 0017-8012. DOI: 00475394. arXiv: 67227835.

[10] R. Kitchin, "The real-time city? Big data and smart urbanism", *GeoJournal*, vol. 79, no. 1, pp. 1–14, Feb. 2014, ISSN: 03432521. DOI: 10.1007/s10708-013-9516-8. [Online]. Available: http://link.springer.com/10.1007/s10708-013-9516-8.

[11] P. G. W. Keen, "Decision support systems: a research perspective", *Decision Support Systems: Issues and Challenges: Proceedings of an International Task Force Meeting*, pp. 23–44, 1980. DOI: 10.1016/B978-0-08-027321-1.50007-9.

[12] R. Kitchin, S. Maalsen, and G. McArdle, "The praxis and politics of building urban dashboards", *Geoforum*, vol. 77, pp. 93–101, Dec. 2016, ISSN: 0016-7185. DOI: 10.1016/J.GEOFORUM.2016.10.006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0016718515300245.

[13] *Dublin Dashboard - Real Time Travel*. [Online]. Available: http://www.dublindashboard.ie/pages/DublinTravel (visited on 11/02/2017).

[14] CASA/UCL, *London City Dashboard*. [Online]. Available: http://citydashboard.org/london/ (visited on 10/27/2017).

[15] *Boston Smartcity*. [Online]. Available: https://boston.opendatasoft.com/page/smart-city-2/ (visited on 11/08/2017).

[16] Universidad Politecnica de Madrid, *Smart CEI Moncloa*, 2015. [Online]. Available: http://ceiboard.dit.upm.es/dashboard/ (visited on 11/08/2017).

[17] C. Pettit, S. N. Lieske, and M. Jamal, "CityDash: Visualising a Changing City Using Open Data", in, Springer, Cham, Jul. 2017, pp. 337–353. DOI: 10.1007/978-3-319-57819-4_19. [Online]. Available: http://link.springer.com/10.1007/978-3-319-57819-4%7B%5C_%7D19.

[18] *City of Sydney Dashboard*. [Online]. Available: http://citydashboard.be.unsw.edu.au/ (visited on 11/08/2017).

[19] H. Hielkema and P. Hongisto, "Developing the Helsinki Smart City: The Role of Competitions for Open Data Applications", *Journal of the Knowledge Economy*, vol. 4, no. 2, pp. 190–204, 2013, ISSN: 18687873. DOI: 10.1007/s13132-012-0087-6.

[20] T. Ojala, "Open urban testbed for ubiquitous computing", *2010 WRI International Conference on Communications and Mobile Computing, CMC 2010*, vol. 1, pp. 442–447, 2010. DOI: 10.1109/CMC.2010.162.

[21] A. Koulakezian and A. Leon-Garcia, "CVI: Connected Vehicle Infrastructure for ITS", *2011 IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 750–755, 2011. DOI: 10.1109/PIMRC.2011.6140066.

[22] L. Sanchez, L. Muñoz, J. A. Galache, P. Sotres, J. R. Santana, V. Gutierrez, R. Ramdhany, A. Gluhak, S. Krco, E. Theodoridis, and D. Pfisterer, "SmartSantander: IoT experimentation over a smart city testbed", *Computer Networks*, vol. 61, pp. 217–238, 2014, ISSN: 13891286. DOI: 10.1016/j.bjp.2013.12.020.

[23] R. N. Murty, G. Mainland, I. Rose, A. R. Chowdhury, A. Gosain, J. Bers, and M. Welsh, "CitySense: An Urban-Scale Wireless Sensor Network and Testbed", *Technologies for Homeland Security, 2008 IEEE Conference on*, pp. 583–588, 2008. DOI: 10.1109/THS.2008.4534518.

[24] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An information framework for creating a smart city through internet of things", *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 112–121, 2014, ISSN: 23274662. DOI: 10.1109/JIOT.2013.2296516.

[25] J. Ormont, J. Walker, S. Banerjee, A. Sridharan, M. Seshadri, and S. Machiraju, "A city-wide vehicular infrastructure for wide-area wireless experimentation", pp. 3–10, 2008. DOI: 10.1145/1410077.1410080.

[26] W. Inmon, *Building the data warehouse*, 9. 2013, vol. 53, pp. 1689–1699, ISBN: 9788578110796. DOI: 10.1017/CBO9781107415324.004. arXiv: arXiv:1011.1669v3.

[27] C. Xavier and F. Moreira, "Agile ETL", *Procedia Technology*, vol. 9, pp. 381–387, 2013, ISSN: 22120173. DOI: 10.1016/j.protcy.2013.12.042. [Online]. Available: https://ac.els-cdn.com/S2212017313001965/1-s2.0-S2212017313001965-main.pdf?%7B%5C_%7Dtid=62b34e6e-b4f3-11e7-931a-00000aacb360%7B%5C&%7Dacdnat=1508434488%7B%5C_%7D%20http://www.sciencedirect.com/science/article/pii/S2212017313001965.

[28] S. H. A. El-Sappagh, A. M. A. Hendawi, and A. H. El Bastawissy, "A proposed model for data warehouse ETL processes", *Journal of King Saud University - Computer and Information Sciences*, vol. 23, no. 2, pp. 91–104, 2011, ISSN: 13191578. DOI: 10.1016/j.jksuci.2011.05.005. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S131915781100019X.

[29] D. Moody and M. A. Kortink, "From Enterprise Models to Dimensional Models: A Methodology for Data Warehouse and Data Mart Design", *Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'2000)*, vol. 2000, pp. 5–16, 2000. [Online]. Available: http://neumann.hec.ca/sites/cours/6-060-00/MK_entreprise.pdf.

[30] R. Kimball and M. Ross, *The data warehouse toolkit: the complete guide to dimensional modelling*. 2011, ISBN: 0471200247. DOI: 10.1145/945721.945741. arXiv: arXiv:1011.1669v3.

[31] Y. Murayama and R. B. Thapa, *Spatial Analysis and Modeling in Geographical Transformation Process GIS based Applications*, 9. 2014, vol. 25, pp. 1682–1690, ISBN: 9780874216561. DOI: 10.1007/s13398-014-0173-7.2. arXiv: arXiv:1011.1669v3.

[32] C. G. Stillwell J., *Applied GIS and Spatial Analysis*. 2004, p. 403, ISBN: 0470844094. DOI: 10.1111/j.0031-868X.2004.282_4.x.

[33] T. C. Bailey, *Spatial Analysis and GIS*. 1994, pp. 8–25, ISBN: 0203221567. DOI: 10.1017/CBO9781107415324.004. arXiv: arXiv:1011.1669v3.

[34] F. Harvey and F. H. PhD, *A Primer of GIS: Fundamental Geographic and Cartographic Concepts*. 2008, p. 310, ISBN: 9781593855659. DOI: 10.1080/13563471003737009. arXiv: arXiv:1011.1669v3.

[35] G. Seeber, *Satellite Geodesy*. 2003, vol. 2, pp. 493–501, ISBN: 3110175495. DOI: 10.1080/00050326.1969.10440171.

[36] *Datum (geodesy)*. [Online]. Available: http://wiki.gis.com/wiki/index.php/Datum_(geodesy) (visited on 10/21/2017).

[37] *WGS84*. [Online]. Available: http://wiki.gis.com/wiki/index.php/WGS84 (visited on 10/21/2017).

[38] M. Haklay and P. Weber, "OpenStreet map: User-generated street maps", *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, Oct. 2008, ISSN: 15361268. DOI: 10.1109/MPRV.2008.80. [Online]. Available: http://ieeexplore.ieee.org/document/4653466/.

[39] P. Mooney and P. Corcoran, "Accessing the history of objects in OpenStreetMap", *Proceedings AGILE*, vol. 353, no. 1, pp. 1–3, 2011. [Online]. Available: https://agile-online.org/conference_paper/cds/agile_2011/contents/pdf/posters/p_115.pdf.

[40] P. Ramsey, S. Santilli, R. Obe, M. Cave-Ayland, and B. Park, *PostGIS — Spatial and Geographic Objects for PostgreSQL*, 2016. [Online]. Available: http://postgis.net (visited on 10/23/2017).

[41] R. O. Obe and L. S. Hsu, *PostGIS in Action*. 2011, p. 520, ISBN: 9781935182269. DOI: citeulike-article-id:11821780. [Online]. Available: https://dl.acm.org/citation.cfm?id=2834495.

[42] T. Honda, "TrailMarker", in *Proceedings of the 2016 on SIGMOD'16 PhD Symposium - SIGMOD'16 PhD*, New York, New York, USA: ACM Press, 2016, pp. 22–26, ISBN: 9781450341929. DOI: 10.1145/2926693.2929903. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2926693.2929903.

[43] S. Erdogan, I. Yilmaz, T. Baybura, and M. Gullu, "Geographical information systems aided traffic accident analysis system case study: city of Afyonkarahisar", *Accident Analysis and Prevention*, vol. 40, no. 1, pp. 174–181, Jan. 2008, ISSN: 00014575. DOI: 10.1016/j.aap.2007.05.004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0001457507000863.

[44] B. Li, S. Tao, and R. W. Dawson, “Evaluation and analysis of traffic noise from the main urban roads in Beijing”, *Applied Acoustics*, vol. 63, no. 10, pp. 1137–1142, Oct. 2002, ISSN: 0003682X. DOI: 10.1016/S0003-682X(02)00024-5. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0003682X02000245.

[45] K. Keay and I. Simmonds, “The association of rainfall and other weather variables with road traffic volume in Melbourne, Australia”, *Accident Analysis and Prevention*, vol. 37, pp. 109–124, 2005. DOI: 10.1016/j.aap.2004.07.005.

[46] G. Broll, H. Cao, P. Ebben, P. Holleis, K. Jacobs, J. Koolwaaij, M. Luther, and B. Souville, “Tripzoom: an app to improve your mobility behavior”, *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*, p. 57, 2012. DOI: 10.1145/2406367.2406436. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2406367.2406436.

[47] Y.-H. Wu, H. J. Miller, and M.-C. Hung, “A GIS-based decision support system for analysis of route choice in congested urban road networks”, *Journal of Geographical Systems*, vol. 3, no. 1, pp. 3–24, 2001, ISSN: 1435-5930. DOI: 10.1007/PL00011466.

[48] R. Sinnott, “Virtues of the Haversine”, *skytel*, vol. 68, p. 158, Dec. 1984, Provided by the SAO/NASA Astrophysics Data System.

[49] *Porto — Red Bull Air Race*. [Online]. Available: http://airrace.redbull.com/en%7B%5C_%7DPT/event/porto (visited on 11/10/2017).