



UNIVERSIDADE D
COIMBRA

João Pedro Simões Marcelino

SIMULATION ENGINE
FOR TRAFFIC SCENARIOS

Dissertation in the context of the Master's Degree in Informatics Engineering, specialization in Intelligent Systems advised by Professor Carlos Bento, PhD and by Ricardo Vitorino and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

July 2023



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D
COIMBRA

DEPARTMENT OF INFORMATICS ENGINEERING

João Pedro Simões Marcelino

Simulation Engine for Traffic Scenarios

Dissertation in the context of the Master's Degree in Informatics Engineering,
specialization in Intelligent Systems, advised by Professor Carlos Bento, Ph.D.
and by Ricardo Vitorino and presented to the Department of Informatics
Engineering of the Faculty of Sciences and Technology
of the University of Coimbra.

July 2023



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D
COIMBRA

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

João Pedro Simões Marcelino

Simulation Engine for Traffic Scenarios

Dissertação no âmbito do Mestrado em Engenharia Informática, especialização em Sistemas Inteligentes, orientada pelo Professor Doutor Carlos Bento e Ricardo Vitorino, apresentada ao Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Julho 2023

Acknowledgements

I would like to begin by expressing my deepest gratitude to my esteemed advisors, Carlos Bento and Ricardo Vitorino, for their guidance, expertise, and support throughout the entirety of this thesis. Their invaluable insights, constructive feedback, and encouragement have played a pivotal role in shaping the outcome of my research. I am truly grateful for their mentorship and the opportunities they have provided for my intellectual and professional growth.

I extend my heartfelt appreciation to the entire team at Ubiwhere for their generous assistance, collaboration, and contributions to this research. The collective knowledge, resources, and meaningful discussions shared by the members of Ubiwhere have significantly enriched my work. I am deeply thankful for their continued support and cooperation throughout this project.

To my loving family, partner, and friends I am immensely grateful for your unwavering support during this difficult year. Your belief in me, your understanding, and your unconditional encouragement have been the bedrock upon which I have built my resilience. Your presence and unyielding love have made all the difference, and I am forever indebted to your relentless faith in my abilities. I hope that my love and appreciation for each of you did not go unnoticed. Your tireless support has been a constant source of strength and motivation throughout this challenging journey.

Finally, I wish to acknowledge all those who have supported me, directly or indirectly, in ways big and small, throughout the completion of this thesis. Your contributions, guidance, and encouragement have played an integral role in my academic achievements.

To all those mentioned and those unmentioned, thank you for being an indispensable part of this incredible journey. In loving memory of my dear grandfather, I dedicate this thesis to his eternal spirit and the unforgettable impact he has left on my life.

Abstract

Many governments are adopting the smart city concept to improve living standards, optimize the utilization of urban resources as well as reach higher levels of transparency and openness. Smart cities can help with big data applications, from energy consumption to garbage maintenance, mobility, and land use.

The development of an efficient plan for mobility changes, specifically road work and infrastructure improvements, poses a significant challenge for municipalities. This challenge stems from the limited resources available to municipalities and their heavy reliance on similar case studies as a basis for decision-making. The task of implementing mobility changes is complex and multifaceted, with each municipality facing unique circumstances. Factors such as population densities, geographical layouts, existing infrastructure, and transportation demands vary from one municipality to another. Consequently, relying solely on similar case studies as a reference point can pose several obstacles and constraints.

A possible solution for the problem can be found in traffic simulation frameworks as analysis support tools and auxiliary to the decision-making process and, as such, lower the effect in costs and schedule, generated by unintended results of the initial plan.

The present dissertation contributes to the development of an integrated system that simulates realistic traffic scenarios, adopting a simulation framework and using data from IoT applications, such as Ubiwhere's Urban Platform, to help public authorities to have a better understanding and evaluating of the traffic flow of the city, and by so, making better decisions and improving the lives of its citizens.

The dissertation results emphasize the potential of combining traffic simulation frameworks with IoT platforms to address challenges encountered by municipalities. The developed integrated system enables the simulation of realistic traffic scenarios, providing valuable insights for optimizing road work, infrastructure improvements, and overall mobility changes. This research serves as a crucial stepping stone towards the realization of smarter and more livable cities, where resources are optimized, and the quality of life is elevated through informed decision-making in urban mobility.

Keywords

Cidades Inteligentes, Simulação de Tráfego, Fluxo de Tráfego, SUMO, Processamento de Dados, Modelo de Simulação, Plataforma Urbana, Veículos Autônomos

Resumo

Muitos governos estão a adotar o conceito de cidade inteligente para melhorar os padrões de vida, otimizar a utilização de recursos urbanos e alcançar níveis mais altos de transparência e abertura. As cidades inteligentes podem ajudar com aplicações de big data, desde o consumo de energia até a manutenção de lixo, mobilidade e uso do solo.

O desenvolvimento de um plano eficiente para mudanças na mobilidade, especificamente obras rodoviárias e melhorias na infraestrutura, representa um desafio significativo para os municípios. Esse desafio decorre dos recursos limitados disponíveis para os municípios e de sua forte dependência de estudos de caso semelhantes como base para a tomada de decisões. A tarefa de implementar mudanças na mobilidade é complexa e multifacetada, com cada município enfrentando circunstâncias únicas. Fatores como densidades populacionais, layout geográfico, infraestrutura existente e necessidades de transporte variam de um município para outro. Consequentemente, depender exclusivamente de estudos de caso semelhantes como ponto de referência pode apresentar vários obstáculos e restrições.

Uma possível solução para o problema pode ser encontrada em estruturas de simulação de tráfego como ferramentas de suporte à análise e auxiliares no processo de tomada de decisão, reduzindo assim o impacto nos custos e cronogramas gerados por resultados indesejados do plano inicial.

A presente dissertação contribui para o desenvolvimento de um sistema integrado que simula cenários de tráfego realistas, adotando uma estrutura de simulação e utilizando dados de aplicações IoT, como a Plataforma Urbana da Ubiquitous, para ajudar as autoridades públicas a terem uma melhor compreensão e avaliação do fluxo de tráfego da cidade, e assim tomar melhores decisões e melhorar a vida de seus cidadãos.

Os resultados da dissertação destacam o potencial de combinar estruturas de simulação de tráfego com plataformas IoT para enfrentar os desafios encontrados pelos municípios. O sistema integrado desenvolvido permite a simulação de cenários de tráfego realistas, fornecendo conhecimentos valiosos para desenvolver obras rodoviárias, melhorias na infraestrutura e mudanças gerais na mobilidade. Esta pesquisa serve como um passo crucial para a concretização de cidades mais inteligentes e habitáveis, onde os recursos são otimizados e a qualidade de vida é elevada por meio de tomadas de decisão informadas na mobilidade urbana.

Palavras-Chave

Cidades Inteligentes, Simulação de Tráfego, Fluxo de Tráfego, SUMO, Processamento de Dados, Modelo de Simulação, Plataforma Urbana, Veículos Autônomos

Contents

1	Introduction	1
1.1	Context and Motivation	1
1.1.1	Opportunity	2
1.1.2	Objectives	3
1.2	Document Structure	3
2	Background	5
2.1	Hosting Company	5
2.2	Ubiwhere’s Urban Platform	5
2.2.1	Architecture	6
2.2.2	Reference City and Smart City Data	7
2.3	IN2CCAM	10
2.4	Scope of the Thesis	10
3	Requirements	13
3.1	Requirements Elicitation	13
3.2	Requirements	15
3.2.1	Functional Requirements for the Framework	16
3.2.2	Functional Requirements for the Integrated System	18
3.2.3	Non-Functional Requirements for the Integrated System	20
3.2.4	Epic and User Stories	22
3.3	Key Performance Indicators	26
3.4	Summary	26
4	Architecture	29
4.1	Context Diagram	29
4.2	Container Diagram	30
4.3	Component Diagram	32
4.4	Technologies	34
5	State of the Art	35
5.1	Traffic Simulation	35
5.1.1	Microscopic Models	36
5.1.2	Macroscopic Models	37
5.1.3	Mesosopic Models	38
5.1.4	Analysis	39
5.2	Frameworks	39
5.2.1	Comparative Analysis	46
5.2.2	Analysis	50

6	Planning and Methodology	55
6.1	Process Management	55
6.2	Planning	56
6.3	Success Criteria	58
6.4	Risk Assessment and Management	58
6.4.1	Risk Assessment	58
6.4.2	Risk Matrix	60
7	Implementation	63
7.1	API Prototype	63
7.2	API Implementation	65
7.2.1	Data Models	66
7.2.2	Public Transportation Schedules	69
7.2.3	Simulation Results	71
7.2.4	Simulated Public Transportation Live Updates	72
7.3	Web Interface	73
7.3.1	OpenStreetMap Exporter	74
7.3.2	Traffic Simulation Proof of Concept	75
7.3.3	Model Creation	76
7.3.4	Model Validation	77
7.3.5	Result Analysis	78
7.3.6	Vehicle Position	79
7.3.7	Simulated Public Transportation Updates	80
7.3.8	Simulation Scenarios	81
7.4	Simulation Efficiency	83
7.5	Scalability	84
7.6	Summary	86
8	Results	87
8.1	Baseline Scenario	87
8.1.1	Porto Roadmap	88
8.1.2	Porto Bus Stops	89
8.1.3	Traffic Congestion Analysis	90
8.1.4	Speed Study	91
8.1.5	Environmental Impact	92
8.1.6	Result Overall Statistics	93
8.1.7	Vehicle Trip information	94
8.1.8	Simulation Time Series	95
8.1.9	Vehicle Position and Statistics	96
8.1.10	Simulated Trip Updates for Public Transportation	98
8.1.11	Key Performance Indicators	99
8.2	Occurrences Scenario	100
8.2.1	Road Block and Speed Restriction	101
8.2.2	Comparison of Traffic Congestion with Baseline Scenario	102
8.2.3	Comparison of KPIs with Baseline Scenario	103
8.3	Road Restriction Scenario	104
8.3.1	Road Restriction	105
8.3.2	Comparison of Traffic Congestion with Baseline Scenario	105

8.3.3	Comparison of KPIs with Baseline Scenario	107
8.4	Reserved Route for CCAM Scenario	108
8.4.1	Autonomous Road Restriction	108
8.4.2	Comparison of Traffic Congestion with Baseline Scenario . .	109
8.4.3	Comparison of KPIs with Baseline Scenario	110
9	Conclusion	113
9.1	Study Findings	113
9.2	Future Work	114

Acronyms

API Application Programming Interface.

CAV Connected Autonomous Vehicles.

CCAM Ccooperative, Connected and Automated Mobility.

GTFS General Transit Feed Specification.

IoT Internet of Things.

KPI Key Performance Indicators.

LLs Living Labs.

O/D Origin Destination.

OSM OpenStreetMap.

PT Public transportation.

R&D Research & Development.

R&I Research & Innovation.

SCTP Sociedade Transportes Colectivos do Porto.

SUMO Simulation of Urban Mobility.

TraCI Traffic Control Interface.

TraMod Traffic Modeller.

TRL Techonology Readiness Levels.

VM Virtual Machine.

List of Figures

2.1	Logical Perspective of Ubiwhere’s Urban Platform.	6
2.2	Example of the web application Urban Platform.	8
3.1	Diagram of the timeline of the interviews and the most important points from each one.	14
3.2	SMART Objectives according to Peter Drucker. The SMART approach to goal-setting has been widely adopted in both business and personal contexts as a way to increase focus and drive toward achieving desired outcomes.	15
3.3	MoSCoW method to classify all the requirements gathered. The MoSCoW method is particularly useful in situations where there are limited resources or time constraints, as it helps teams to prioritize their work and to make informed decisions about which tasks to tackle first.	16
4.1	Context Diagram of the Integrated System	30
4.2	Container Diagram of the Integrated System	31
4.3	Component Diagram of the Integrated System	33
5.1	Visual representation of the simple car-following model example created by Himite[10].	36
5.2	Example created by Zhao et al.[16] using cellular automata to simulate a 4-way intersection with three entrance lanes, where each car is represented by 1x2 cells.	37
5.3	Sketch of the nearest neighbors of a central vehicle c considering a lane change to the left. The new and old successors are denoted by n and o , respectively. Accelerations after a possible change are denoted with a tilde[19].	37
5.4	Example of an intersection with traffic lights, created via scripting, and visualised in PTV VISSIM.	40
5.5	Example of a taxi and bus station created by Sumo with Python and visualized in SUMO GUI.	41
5.6	Example of a roundabout visualized in Aimsun Next.	42
5.7	Example of MATSim network visualization, consisting of 15 roads	43
5.8	Image from case-study of Pilsen using the framework Traffic Modeller in [39]	45

5.9	Image from Flow’s built-in traffic networks, "Figure Eight". The goal is to maximize the system-wide velocity for fourteen vehicles, which necessitates spacing the vehicles so that they do not run into conflicts at the merging points. It is possible to change the rate of autonomous vehicles to human drivers.	46
5.10	Excerpts taken from a) Sumo, b) Aimsun and c) MATSim, using OpenStreetMap (OSM), of Polo2 of Universidade de Coimbra. . . .	47
5.11	Representation of the activity plans created for the experimentation.	49
6.1	Gantt Diagrams of the thesis project, with the first conceptualized diagram, and updated at the end of the first semester. Light blue represents the activities planned at the start of the project. Darker blue shows the new activities added at the end of the first semester. The initial activity duration is represented in yellow. The modified activity duration is shown in green.	57
6.2	Risk matrix	61
7.1	64
7.2	Diagram depicting the intricate connections and interactions between various functions within SUMO. This diagram played a crucial role in facilitating a deep understanding of how the API should be effectively implemented.	65
7.3	Diagram of entity-relationship between resources, and their parameters.	67
7.4	Diagram of interactions between the various GTFS files, by [42] . . .	71
7.5	Diagram of GTFS RT feed messages.	73
7.6	Display of a section from the OSM Exporter webpage.	74
7.7	Display of a section from the Traffic Simulation Proof of Concept webpage.	76
7.8	Display of a section from the Model Creation webpage.	76
7.9	Visualization of a specific segment on the Model Validation webpage, showcasing the various options for road visualization. This includes the ability to view the entire network as well as specific subsets of roads affected by specific incidents or occurrences.	77
7.10	Visualization of a specific segment on the Result Analysis webpage, showing parameters for the analysis of road data.	78
7.11	Visualization of the Vehicle Position webpage.	80
7.12	Visualization of the first component of Simulated Public Transportation Updates webpage.	81
7.13	Example of the Key Performance Indicator display component. . . .	83
7.14	Display of Prometheus running in communication with Django and cAdvisor.	84
7.15	Display of Grafana Dashboard, analysing multiple metrics from Prometheus.	85
7.16	Display of cAdvisor CPU metrics, such as Total Usage and Usage per Core, from Prometheus.	85
8.1	Display of Porto road map used for the baseline scenario.	89

8.2	Analysis of Bus Stops (a) in GTFS data and (b) in SUMO	90
8.3	Visualization of density and occupancy in Porto, between 14h and 17h	91
8.4	Breakdown of average speed and relative average speed in Porto, between 14h and 17h	92
8.5	Visualization of (a) total emissions of CO ₂ , (b) emissions of PM _x per hour per road length, and (c) emissions of NO _x by vehicle. City of Porto, between 14h and 17h	93
8.6	Example of overall statistics that can be gathered from the baseline simulation scenario.	93
8.7	Example of a plot created with the duration of trip and CO ₂ emitted by the vehicle. Each point corresponds to a vehicle.	94
8.8	Plot created with the waiting time and CO ₂ emitted by the vehicle. Each point corresponds to a vehicle.	95
8.9	Example of a plot created using the time series component to visualize the mean waiting time (seconds) of the simulation.	96
8.10	Visualization of vehicle position in Porto at 15h. Red points correspond to private vehicles and yellow points to public transportation vehicles.	97
8.11	Visualization of statistical information about the simulation of Porto at 15h.	97
8.12	Visualization of Porto traffic congestion between 15h and 18h in the baseline scenario.	101
8.13	Analysis of Porto highlighted lane blocked in Ponte da Arrábida, created for the occurrences scenario.	102
8.14	Visualization of comparison of density between the occurrence scenario with the baseline scenario.	103
8.15	Visualization of Porto traffic congestion in Rua do Campo Alegre between 15h and 18h	105
8.16	Visualization of a Porto street limited access to private vehicles.	105
8.17	Visualization of Porto traffic congestion comparison in Rua do Campo Alegre between 15h and 18h with the baseline scenario.	106
8.18	Visualization of Porto highlighted CCAM route for autonomous vehicles.	109
8.19	Visualization of Porto comparison density of autonomous route created and the baseline scenario.	109

List of Tables

3.1	Functional requirements that focus on the necessities of the simulation framework.	17
3.2	Functional requirements that focus on what the needs of the Integrated System are, as well as what can be its capabilities.	19
3.3	Non Functional requirements that focus on what the needs of the simulation are.	21
3.4	Table of KPIs from IN2CCAM that are evaluated by the API of the solution developed in the thesis.	26
5.1	System details of the image used to create the virtual machine	48
5.2	Different runs made in SUMO	49
5.3	Different runs made in MATSim	50
5.4	Summary of reviewed simulators	51
5.5	Evaluation of the framework according to the requirements elicited. (Y) It has the capability of doing the requirement. (N) It was found proof that it is not possible to do that requirement. (-) There exists no information about the technology required to do that requirement	53
6.1	Risk assessment table of the project.	59
6.2	Likelihood categories used in the risk matrix.	61
6.3	Impact categories and description used in the risk matrix.	61
7.1	Table of 5 instances from GTFS file stop_times.txt from Porto City. . .	70
8.1	Table of KPIs obtained from the baseline scenario in Porto.	100
8.2	Table of KPIs obtained from the Ocurrences scenario in Porto and compared with Baseline Scenario.	104
8.3	Table of KPIs obtained from the Restricted Access scenario in Porto and compared with Baseline Scenario.	107
8.4	Table of KPIs obtained from the baseline scenario in Porto.	110

Chapter 1

Introduction

Smart Cities [1; 2] have been supporting innovations that use advanced technology and data analysis to improve the quality of life for their residents while optimizing the use of natural resources and creating a more sustainable livable environment, by promoting policies such as public transportation, encouraging green energy and providing access to public spaces and services for all citizens.

One area in which smart city technology can excel is traffic management, collecting data from traffic cameras and sensors and processing it so that it is possible to take action and optimize the traffic flow and reduce congestion in key areas of the city.

Smart cities traffic management technology can also be a key component in the delivery of the milestones set for 2050 by the European Commission and its plan for green, smart and affordable mobility [3].

Traffic simulation is the mathematical modeling of vehicular movement on networks of roads and highways. This type of simulation is an important tool in the planning and design of transportation systems. The use of traffic simulation has proliferated in recent years, playing an important role in the study of mobility, due in part to the increasing availability of powerful computing resources and the development of more user-friendly modeling software.

It is commonly used to evaluate the impact of proposed changes to transportation infrastructure and to help manage traffic congestion. Traffic simulation models can be used to estimate the travel time and delay experienced by vehicles, to identify bottlenecks and choke points and to assess the effects of incidents and accidents.

1.1 Context and Motivation

The study of traffic flow is the analysis of interactions between people and infrastructure with the aim of understanding and designing an optimal transport network with efficient movement of traffic and minimal traffic congestion problems.

One challenge that governmental entities encounter is the current difficulty in creating a strategy capable of addressing problems in the city network, only having as tools existing case studies, which are lengthy processes until the final evaluation is reached. The duration of the said procedure may allow the technology to update or even the necessities of the city to be adjusted or changed.

The implementation of the solution may also cause some concerns. If the results of the modifications on the roads harm the overall flow of the city, the construction can possibly require to be reverted, to the previous state¹, causing a monetary impact on the city, as well as increasing the duration of the implementation, which may grow impatience and discontentment on the general population.

Consequently, all of these issues together may drive public authorities to be more cautious about making any alterations to the city network unless they are absolutely necessary.

1.1.1 Opportunity

One possible solution is a traffic simulation platform that replicates the city environment, enabling a comprehensive understanding of the potential impacts on the relationship between citizens and the improved infrastructure. This solution aims to mitigate the challenge of implementing city alterations without full knowledge of their potential outcomes. By conducting multiple simulations within a condensed timeframe, this approach allows for a more efficient evaluation of various scenarios and their implications.

Some platforms were created with this issue as the focal point, such as SUMO, PTV VISSIM, Aimsun, and multiple others. Although all these tools can create useful simulations and results, incorporating real data from the urban environment, such as traffic flow, public transportation routes, and city network specifics, presents a significant challenge. This data integration challenge, coupled with the complexity and extensive functionality of these platforms, can often discourage end-users from utilizing them.

In parallel, there are several Internet of Things (IoT) platforms, data and analytical platforms created with the intent of extracting real-time indicators to comprehend the behaviors of the different types of vehicles and situations encountered. For instance, Ubiwhere's Urban Platform provides a comprehensive solution for urban management and optimization, while integrating valuable data such as traffic flow intensity and average speed, road topology and even parking occupancy of various municipalities, but lacks mechanisms to provide traffic scenarios simulations based on this data.

Another project from Ubiwhere is IN2CCAM ², a project funded under Horizon Europe Program [4]. By integrating various aspects of urban management and mobility, IN2CCAM aims to create an intelligent and interconnected infras-

¹Road Work reversion example of one-way streets to two-way streets in Manitowoc, Wisconsin (2022): <https://www.manitowoc.org/2450/8th-and-10th-Street-Two-Way-Reversion>

²IN2CCAM: <https://cordis.europa.eu/project/id/101076791>

structure that can optimize citizen experiences and urban efficiency. This project presents an opportunity to utilize simulation techniques for automating Cooperative, Connected and Automated Mobility (CCAM). The main objective is to minimize risks associated with their implementation and evaluate their impact on municipalities. By employing simulation, the project aims to create a controlled environment to test and study the implications of integrating these vehicles into urban settings. This approach enables a thorough understanding of the potential challenges and benefits, informing decision-making processes and promoting safer and more efficient transportation systems within municipalities.

With these concerns in mind, and based on the needs from customers and partners of Ubiwhere, the opportunity to expand Ubiwhere's solution arose, simplifying the usability of the platforms and consolidating the functionalities on a single integrated system.

1.1.2 Objectives

The objective of this project is to develop an integrated system to manage and execute traffic simulations that can gather information on multiple mobility scenarios, using the data from IoT platforms.

To accomplish this, the project utilizes data extracted from Ubiwhere's Urban Platform, which already possesses the capability to gather, store, and process relevant information for the assignment.

The ultimate objective is for the developed application to be integrated in the future with Ubiwhere's Urban Platform, by providing valuable insights for decision-making processes and promoting safer and more efficient transportation systems in municipalities, and as a prototype for IN2CCAM, with the purpose of establishing a controlled environment for testing and studying the consequences of incorporating these vehicles into urban environments.

As part of the process, the framework has been chosen based on its ability to meet the needs and requirements of both Ubiwhere and its customers. The selected framework underwent further analysis and detailing. These traffic simulation tools use different models and algorithms to simulate the movement of vehicles and pedestrians. A comparative analysis of the tools has been conducted as an auxiliary to the final decision.

In summary, the ultimate objective is to integrate the consolidated system with Ubiwhere's products, enabling the usage of traffic simulation outcomes to enhance the decision-making processes of public authorities in the context of planning new mobility solutions or interventions for public spaces.

1.2 Document Structure

This document acts in accordance with the following structure.

The second chapter presents some background concepts and how this project can be used in Ubiwhere's ecosystem.

The third chapter corresponds to the gathering and processing of the requirements for the integrated system and traffic simulation framework.

The fourth chapter presents the architecture that will be implemented for the integrated system.

The fifth chapter of this document is dedicated to the State of the art, where different models of traffic simulation are presented, as well the comparative analysis of the traffic simulation tools.

The sixth chapter introduces the risk planning of the first and second semesters, along with an assessment of possible risks, their proposed mitigation plan, and an evaluation assessment.

The seventh chapter focuses on the platform's implementation and the decisions made during this phase.

In the eighth chapter, a range of scenarios are presented, along with the obtained results and a subsequent discussion on those outcomes.

The last chapter concludes the dissertation with a summary of the main outcomes and next steps.

Chapter 2

Background

The purpose of this chapter is to provide an overview of the hosting company and highlight important projects related to the implementation of the integrated system.

2.1 Hosting Company

Ubiwhere ¹ is a Portuguese Research and Innovation company created in 2007, that specializes in the development of technology solutions for the Internet of Things (IoT), Smarts Cities and Industry 4.0.

Ubiwhere is dedicated to the development of smart city solutions, with a particular focus on mobility-related technologies. They actively engage in various projects, including but not limited to bike-sharing systems, smart parking solutions, traffic management initiatives, and fleet management platforms. Their expertise extends beyond these areas, as they continuously explore innovative solutions to enhance urban mobility and address the complex challenges faced by modern cities.

The company designs and develops these solutions that analyze and process the data collected by IoT devices and systems and operates closely with municipalities, service providers and government authorities to provide consulting services that help them implement and manage smart city solutions that meet their needs.

2.2 Ubiwhere's Urban Platform

Ubiwhere's Urban Platform ² stands as the flagship product developed for smart cities, released in 2018, that enables the collection, analysis, and management of data from various sources in real-time, such as cameras, sensors and other IoT de-

¹Ubiwhere: www.ubiwhere.com

²Urban Platform: <https://urbanplatform.city/>

vinces. The Urban Platform allows cities to unlock and open the silos between their sectors (mobility, environment, and infrastructure), allowing them to increase the efficiency of municipal operations and the transparency towards citizens, while improving the performance of governmental services, aimed at the citizens' welfare and quality of life.

This platform provides city managers and administrators single integrated system that processes and analyses data from multiple sources and displays it in a user-friendly interface where the municipalities can make decisions to optimize the performance of the systems and services, enhancing the quality of life for their citizens, as well as providing a more efficient and sustainable urban life.

Using Technology Readiness Levels (TRL), a method for estimating the maturity of technologies during the acquisition phase of a program, Urban Platform was evaluated in level TRL 7 or system prototype demonstration in an operational environment, according to the European Commission Decision[5].

2.2.1 Architecture

The Urban Platform is web-based and can be implemented on public and private clouds on local infrastructure. The platform exposes open Application Programming Interface (API)s, which are the entry points, to allow external applications to interact with the database in an indirect yet secure and controlled way.

The architecture presented by Ubiwhere follows a multi-layer approach composed of 5 fundamental layers to the business logic: the support layer, data acquisition layer, knowledge layer, interoperability layer and presentation layer. Each of these specializes in an essential part of the platform.

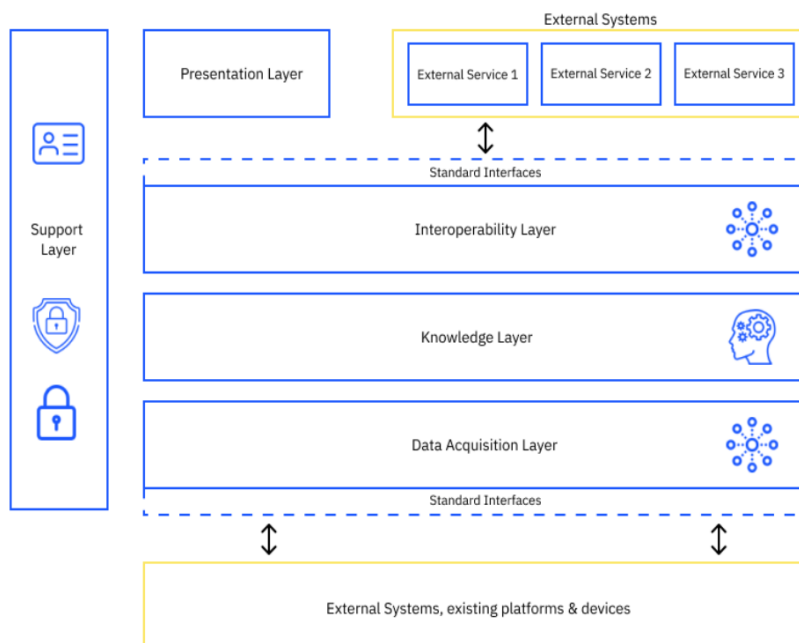


Figure 2.1: Logical Perspective of Ubiwhere's Urban Platform.

The support layer is responsible for the platform's security, authentication and authorization mechanisms. The data acquisition layer has the functionality of integrating all the data into the platform, in an incremental and controlled way, to ensure that the domains are standardized and harmonized. The knowledge layer is where all the platform's data is stored and analysed. The interoperability layer is responsible for providing all the data and interfaces in the form of REST APIs or WebSockets, compliant with open standards and reference best practices. The presentation layer is the visual layer of the platform, where the screens and visual elements are presented to help the decision-making process, including a live map of the city, a live video feed of multiple cameras, a list of occurrences, and even analytics and city indicators.

Therefore, the layers are built with microservices to enhance the solution's scalability and encourage the rapid integration of new components.

2.2.2 Reference City and Smart City Data

The Urban Platform is used by multiple cities and regions, providing the municipalities with the opportunity to have a centralized and organized way of visualizing and analyzing the data that is gathered and worked. This data can be gathered from various sources, be it from sensors, platforms, services via APIs or directly from citizens, who report occurrences through mobile apps, surveys and information systems. One of Ubiwhere's case studies is the city of Guimarães, Portugal, which was created using information from the following datasets:

- **Real-time traffic information** - provided by HERE technologies ³, WAZE ⁴ and CWAY ⁵ radars
- **Off-street parking occupancy information** - provided by the municipality
- **Electrical Vehicle charging stations' locations and usage** - provided by MOBI.E ⁶
- **Touristic and social events data** - provided by the municipality
- **Mobility and Civil incidents and planned roadworks** - provided by Guimaraes Citify application ⁷
- **Air Quality and Weather observations** - provided by CWAY weather and noise stations
- **Foot and bicycle traffic information** - IoT devices provided by CWAY
- **Road network** - provided by OpenStreetMaps

³HERE Technologies: <https://www.here.com/>

⁴WAZE: <https://www.waze.com/>

⁵CWAY: cway.pt

⁶MOBI.E: <https://www.mobie.pt/>

⁷Guimaraes Citify: <https://play.google.com/store/apps/details?id=com.guimaraes.cityfy>

- **3D Geospatial Information** - provided by the municipality and updated periodically due to its high volume
- **Information and Geographic representation of watercourses** - provided by the municipality
- **Micromobility data on vehicles location and docks availability** - provided by Bolt ⁸

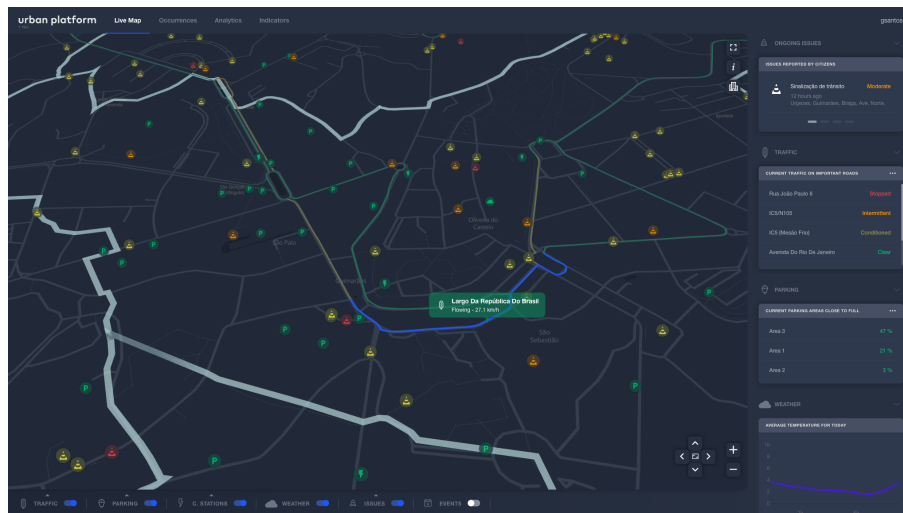


Figure 2.2: Example of the web application Urban Platform.

This case study is interesting because it was considered a best practice of the European Commission, and also due to the number of datasets that are used in the platform and how they can relate to each other, creating an efficient medium for the local government to correlate data from multiple distinct sources and understand the impact of specific events for better decisions and subsequently, create a better environment for all citizens.

Although the platform complies with open standards, the diverse sources for the data and its different structures make the information assimilation and analysis more complex and time-consuming. Data availability and volume also stood a problem to the team, as despite the fact of the historical importance of the city, there was little information available on worldwide data services, shifting gathering responsibility and costs to the city. Geospatial data correlation was also a concern, as it is a time-consuming process due to the large amounts of data.

The collaboration between Ubiwhere on the Platform, extended to the PT2020 project City Catalyst ⁹. The objective was to research and develop technological solutions and innovative services. Ubiwhere's main contribution aimed to promote to promote an effective digital transition in cities, by aggregating data from the existing systems and analyzing it to provide Key Performance Indicators (KPI) with context, which can be used as valuable insights for decision-making.

⁸Bolt: <https://bolt.eu/pt-pt/>

⁹PT2020 project City Catalyst: <https://www.ubiwhere.com/en/research-innovation/smart-cities-and-communities-mob-citycatalyst-porto-aveiro-guimaraes-famalicao-e-cascais>

Guimarães is a city that not only uses Ubiwhere's Urban Platform as a client but also collaborates with Ubiwhere in research and development projects, and holds a significant position in the context of technological advancement. One notable achievement of their partnership is the development of automated KPIs for the city through the City Catalyst project. Additionally, it is envisioned the use of IoT data from the Urban Platform to simulate autonomous vehicles as part of the IN2CCAM initiative. Considering these factors, Guimarães was initially regarded as the most suitable choice and used during the process of requirements elicitation and early technology development.

However, despite its commendable progress, Guimarães still faces certain challenges in terms of data readiness, particularly related to public transportation information, which is currently in the early stages of development. As a result, an alternative city was sought to ensure the smooth execution of comprehensive simulations for the thesis.

Hence, the city of Porto emerged as the selected location for conducting the simulations. Porto, like Guimarães, is a partner of Ubiwhere and benefits from a collaborative relationship. However, Porto has an edge over Guimarães when it comes to data availability and maturity. The datasets in Porto, including open public transportation data, traffic conditions, and information on loading and unloading activities, are readily accessible and well-prepared for use.

By utilizing Porto as the simulation environment, one can harness the benefits offered by its larger scale and dimensions. The city also boasts a more extensive road network, enabling the examination of traffic dynamics in diverse scenarios. Moreover, Porto accommodates a greater number of vehicles per day, thereby providing a realistic and comprehensive foundation for testing and evaluating the developed simulations.

Some datasets that are available to be used in Porto:

- **General Transit Feed Specification (GTFS)** - provided by Sociedade Transportes Colectivos do Porto (SCTP) and Porto open data ¹⁰
- **Traffic condition** - provided by Porto open data ¹¹
- **Loading and Unloading** - provided by Porto open data ¹²
- **Pick-up and Drop-off Zones** - provided by Porto open data ¹³

Succinctly, all these different datasets, with information on traffic, parking occupancy, road network and even electric vehicle charging stations, as well as the various concerns in the implementation process, create an interesting case study that encourages the exploration of numerous functionalities of Ubiwhere's Urban Platform.

¹⁰GTFS: <https://opendata.porto.digital/zhHansCN/dataset/horarios-paragens-e-rotas-em-formato-gtfs-stcp>

¹¹Traffic condition: <https://opendata.porto.digital/dataset/condicionamentos-de-transito>

¹²Loading and Unloading: <https://opendata.porto.digital/dataset/cargas-e-descargas>

¹³Pick-up and Drop-off Zones: <https://opendata.porto.digital/dataset/zona-de-tomada-e-largada-de-passageiros>

2.3 IN2CCAM

IN2CCAM ¹⁴ is a funded project under the Horizon Europe Program[4]. It is composed of a consortium of 20 participants, which Ubiwhere is a part of alongside Quadrilátero, an association composed of the Portuguese municipalities of Barcelos, Braga, Guimarães, and Vila Nova de Famalicão. According to the vision of the Horizon Europe[4] framework program from 2021-2027, that aims to accelerate the implementation of Ccooperative, Connected and Automated Mobility (CCAM) technologies and systems for passenger and goods. As this project was set to begin on 1 November 2022 and end implementation on 31 October 2025, it is still in a very early stage.

This project has the objective of providing benefits to all citizens, with the help of CCAM services in the transportation system with expected results in areas such as safety, environment, and inclusiveness.

A set of operational solutions will be proposed and implemented with 4 Lead Living Labs (LLs) in the following areas: Tampere (Finland), Trikala (Greece), Turin (Italy), and Vigo (Spain). Two follower LLs will help in the design phase and data assessment in Bari (Italy) and in the city of Guimarães (Portugal), in Quadrilátero.

The objective of the Quadrilátero Living Lab is for Ubiwhere and Quadrilátero to research and develop simulation technology CCAM. This involves maintaining communication with other LLs, with the aim of optimizing urban space. This Living Lab also aims to increase the quality and liveability of urban and metropolitan areas by reducing congestion, and air and noise pollution via the impact assessment of simulation of diverse mobility scenarios.

Lastly, this partnership also aspires to provide usable digital tools for municipalities for urban planning and traffic management, as well as valuable data to various decision-makers.

2.4 Scope of the Thesis

The thesis has the objective of providing a needed tool to operate with the data gathered and help accomplish the goal of the smart city movement to reach its full potential by creating traffic simulations that can support the municipalities in making more accurate decisions to improve traffic efficiency and sustainability. This application will create new features that will complement both Urban Platform and serve as a proof of concept for IN2CCAM. These enhancements will further enhance the capabilities and functionalities of Ubiwhere's platforms, providing users with an enhanced and comprehensive experience, of planning and simulation of urban mobility scenarios.

This thesis is intended to deliver a stand-alone solution that responds to IN2CCAM

¹⁴IN2CCAM: <https://cordis.europa.eu/project/id/101076791>

challenges and requirements, in terms of providing different simulation scenarios of CCAM vehicles, that is also aligned with the Urban Platform architecture so as to be integrated in the future as part of the product roadmap.

Consequently, it is crucial to develop the integrated system in a modular and scalable manner, allowing for future integration and expansion. This is achieved by leveraging open-source technology and implementing APIs that are prepared for seamless integration with other systems.

Chapter 3

Requirements

Requirements establish the foundation for product vision, scope, cost, and schedule and they ultimately must target finished product quality and performance. They define the limitations of the project and determine what needs to be done to complete the project successfully. Additionally, requirements help to ensure that the project stays on track and delivers value to the intended audience.

This chapter describes the requirements collection process, describes the requirements for the solution, and organizes them into several categories: requirements of the simulation framework, and requirements for the integrated system that has been created using the simulation framework and UrbanPlatform's data.

3.1 Requirements Elicitation

The vision of creating an integrated system comes from a development backlog of Ubiwhere's product, Urban Platform, as well as a contribution from Research & Development (R&D) projects, such as IN2CCAM, a European-funded project that focuses on CCAM (mentioned in section 2.3). Fundamentally, this means that the requirements do not come from a specific customer with a group or set of necessities, but from interviewing different staff members from Ubiwhere, with different roles, to ensure a complete overview of the needs backlog for the traffic simulation framework.

The members that were interviewed were:

- Beatriz Barateiro - Product Owner of the Urban Platform
- Ricardo Vitorino - Smart Cities Research & Innovation (R&I) Manager and Thesis Advisor
- Rita Santiago - Head of R&I representing Ubiwhere in the CCAM Partnership
- André Duarte - Head of Tech & Future Mobility

Within Ubiwhere, each member holds a distinct role and is assigned a specific set of projects and individuals to oversee. This allocation of responsibilities allows for diverse viewpoints on various subjects, including the definition of the simulation platform, its required capabilities, and even the most optimal approach to achieve desired outcomes.

Every meeting tried to follow the same structure. Initially, the meeting started with a briefing about the problems identified along with the objectives to reduce and control said problems. After the first part of the meeting, based on what was presented, the interviewee was questioned about what the focus of the simulation platform should be, and what requirements were necessary to integrate with other Ubiwhere projects.

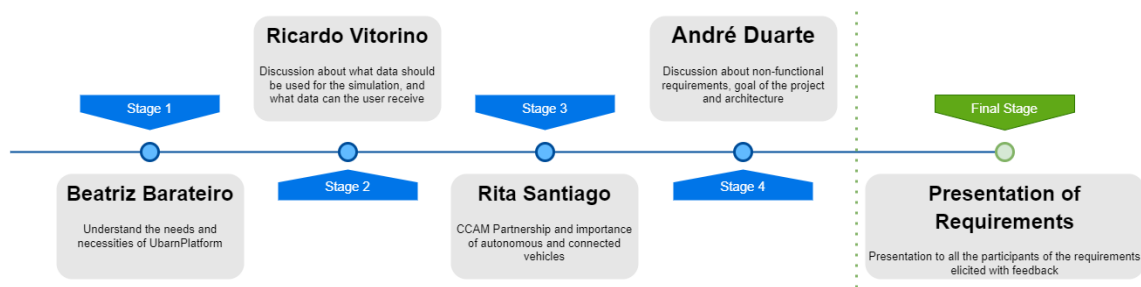


Figure 3.1: Diagram of the timeline of the interviews and the most important points from each one.

The first meeting was with Beatriz Barateiro, Product Owner of the Urban Platform, and focused on understanding the needs and requirements of the Urban Platform’s potential and actual customers.

The interview with Ricardo Vitorino, Smart Cities R&I Manager, and Thesis Advisor focused on the necessities of the integrated system, the important scenarios that the simulation tool must be able to successfully simulate, and the information that the framework must be able to use in its execution. It was also discussed what interactions should be promoted between the user with the integrated system.

The meeting with Rita Santiago, Head of R&I and in charge Ubiwhere’s participation in the CCAM Partnership ¹, focused on the CCAM Partnership, and what should be done in the thesis to integrate it in the most seamless way possible. This encounter allowed me to identify some requirements for autonomous and connected vehicles, and their importance for the future of city mobility.

The final talk was with André Duarte, Head of Tech & Future mobility engineer, and was focused on filtering some requirements and discussing some non-functional requirements. The conversation was also important to adjust the goals of the thesis, and even some suggestions and helpful tips, in terms of architecture and formatting.

These one-on-one meetings allowed me to have a better understanding of the objectives and goals, as it created the inevitability of explaining the project suc-

¹CCAM Partnership reference: <https://www.ccam.eu/>

cinctly but also the understanding of what are the customers' and partners' needs and the company's strategy for the product. After all the meetings, a draft list of the requirements was compiled and presented to the whole group in a scheduled online appointment. After the presentation of the requirements, the list was updated with all the feedback given.

3.2 Requirements

After the requirements for a project have been elicited, several steps can be taken to ensure that the requirements are properly understood and that the project can move forward. The steps taken were: documentation, review and validation, and prioritization of the requirements

Documenting the requirements is a key step in the requirements-gathering process, as it is important to have a clear written record to ensure an understanding of what needs to be done. It also can serve as a reference point throughout the project.

The review and validation of requirements are crucial in ensuring the success of a project. It is necessary to ensure that they are clear, complete, and accurate. To achieve this goal, a popular approach was used, SMART Objectives, developed by Peter Drucker [6]: which defines that the requirements must be specific, measurable, achievable, relevant, and time-bound.



Figure 3.2: SMART Objectives according to Peter Drucker. The SMART approach to goal-setting has been widely adopted in both business and personal contexts as a way to increase focus and drive toward achieving desired outcomes.

The prioritizing of requirements is an essential step in the project planning process. The decision of categorizing the requirements by importance is integral to ensure that the efforts are focused on the most valuable work, resulting in its completion before other tasks. This can help guarantee that the project meets the defined objectives in a timely and efficient manner. The requirements were classified in priority level: must have, should have, could have, or won't have, using a popular classification method, the MoSCoW method (Royce, 1990s).



Figure 3.3: MoSCoW method to classify all the requirements gathered. The MoSCoW method is particularly useful in situations where there are limited resources or time constraints, as it helps teams to prioritize their work and to make informed decisions about which tasks to tackle first.

With the objective of better visualization and interpretation of all the requirements gathered and processed, a second classification was created. In this classification, the requirements were separated according to what problem or main task they were aimed at. As such, three tables were assembled with the following division:

- Functional Requirements for the Framework - aimed to specify the requirements for the choice of the traffic simulation tool used to achieve the objective of the thesis
- Functional Requirements for the Integrated System - describes the functionalities needed for the integration of the framework in a scalable and modular way.
- Non-Functional Requirements for the Integrated System - defines the requirements that guarantee the overall quality of the project, and it meets the desired standards, delivering a high-quality user experience

3.2.1 Functional Requirements for the Framework

The objective of this thesis is to create an Integrated System, which adopts a state-of-the-art traffic simulation framework chosen from the ample list of traffic simulation tools used in the community. Each framework addresses a distinct set of challenges that its creators seek to overcome, resulting in no single traffic simulation tool being universally superior in every scenario. Consequently, selecting the most suitable tool becomes a complex decision influenced by various factors, offering a unique and intricate choice.

To choose the best simulation tool for the integrated system, it is necessary to identify the main functionalities that are crucial. The requirements tabulated in this section were considered to be the most focused on the selection of the simulation framework.

ID	Title	Description	Priority
F-1	Simulate different types of vehicles	The simulation tool must be able to simulate and differentiate personal and commercial vehicles moving around the different roads based on their restrictions and behaviors	Must have
F-2	Simulate public transport vehicles, routes and schedules	The simulation tool must be able to simulate public transportation vehicles and their routes based on the schedules	Must have
F-3	Simulate public service vehicles, routes and schedules	The simulation tool must be able to simulate public service vehicles (e.g. taxis), their patterns and usual waiting spots and routes	Must have
F-4	Simulate autonomous vehicles	The simulation tool must be able to simulate autonomous vehicles on roads classified for such vehicles	Must have
F-5	Simulate pedestrian movement	The simulation tool must be able to simulate pedestrians moving around the different sidewalks	Could have
F-6	Simulate non-motorized vehicles	The simulation tool must be able to simulate non-motorized vehicles on roads classified to allow such vehicles	Could have
F-7	Simulate parking areas occupancy	The simulation tool must be able to simulate parking areas occupancy, such as the ability to park vehicles for some time	Could have
F-8	Simulate unsignalized intersections	The simulation tool must be able to simulate drivers' behavior and patterns on unsignalized intersections between roads	Must have
F-9	Simulate signalized intersections	The simulation tool must be able to simulate drivers' behavior and patterns on traffic signals in roads and intersections	Must have
F-10	Simulate roundabouts	The simulation tool must be able to simulate roundabouts and their rules of passage, as well as the different drivers' behaviors	Must have
F-11	Simulate using batch mode	The simulation tool must be able to do "batch simulation", to verify the output of the simulation with better efficiency	Must have
F-12	Simulate using microscopic models	The simulation tool must use a microscopic model, as it allows to track all individual vehicle movements	Must have
F-13	Obtain new traffic flow data	The simulation tool must be able to obtain new traffic flow data from the simulation executed	Must have

Table 3.1: Functional requirements that focus on the necessities of the simulation framework.

3.2.2 Functional Requirements for the Integrated System

The main purpose of the thesis was to design a system that processes data from multiple sources of data, such as IoT applications or smart cities platforms, and integrate it with the chosen traffic simulation framework so that changes to the city's transportation environment can be made, and the results can be comprehensively examined.

To create the Integrated System, it was necessary to gather all the information about the types of data available for the study, and what functionalities can be extracted from it.

This group of requirements focuses on the functional requirements that aim to create a good understanding of the Integrated System, its needs, and capabilities.

ID	Title	Description	Priority
IS-1	Create road map based on OpenStreetMap (OSM) data	The platform must be able to create a road map based on OSM data, either by specific file or bounding box	Must
IS-2	Process historical traffic data	The platform must be able to process historical traffic data as files, to generate simulations based on traffic intensity and average speed	Must
IS-3	Process road slope data	The platform must be able to process data on road slope to simulate more realistic traffic speeds and vehicle restrictions	Should
IS-4	Process road topology data	The platform must be able to process data on road topology to simulate more realistic traffic speeds and vehicle restrictions	Must
IS-5	Process speed limit data	The platform must be able to process data on the speed limits of each road to simulate more realistic traffic speeds and vehicle restrictions	Must
IS-6	Process parking availability data	The platform must be able to process data on parking availability to simulate more realistic traffic scenarios	Could
IS-7	Process public transportation routes	The platform must be able to process data on public transportation routes and schedules to simulate scenarios with public transit vehicles	Must
IS-8	Process non-motorized vehicle data	The platform must be able to process data on non-motorized vehicle movement, such as historical traffic data from non-motorized vehicles	Could
IS-9	Close roads for a specific period	The platform must be able to close one or more roads for a specified period and segment	Must
IS-10	Close lane segments for a specific period	The platform must be able to close one or more lanes for a specified period and segment	Must
IS-11	Choose from a list of possible events	The platform must be able to choose from a list of possible events for the lane/road/area in question to create the simulation	Should
IS-12	Change the available number of parking spaces	The platform must be able to change the available number of parking spaces on a given road or parking area	Could
IS-13	Change the speed limitations	The platform must be able to change the speed limits on certain roads, lanes, and intersections	Must
IS-14	Create new public transportation	The platform must be able to create new public transportation routes and schedules	Must
IS-15	Change new public transportation	The platform must be able to change existing public transportation routes and schedules	Must
IS-16	Create new non-motorized only lanes and roads	The platform must be able to create new micro-mobility tracks, zones and lanes only allowed for non-motorized vehicles	Could
IS-17	Evaluate low-emission zones and CO2 emissions	The platform must be able to evaluate low-emission zones, estimating the reduction in CO2 emissions	Could

Table 3.2: Functional requirements that focus on what the needs of the Integrated System are, as well as what can be its capabilities.

3.2.3 Non-Functional Requirements for the Integrated System

Non-functional requirements serve great importance because they describe the behavior of the system, rather than what it should do. Meeting these requirements is crucial as they have a significant impact on the overall satisfaction and usefulness of the Integrated System to the final users. Ensuring that these conditions are met is critical to its success and long-term viability.

ID	Title	Description	Priority
NF-1	Extensibility of the platform	The platform must be built in a way that is extensible, such that it can be possible to add features in the future	Must have
NF-2	Efficiency in simulating	The platform must be able to perform simulations efficiently, without wasting unnecessary resources	Must have
NF-3	Processing of large amounts of data	The platform must be capable of processing large amounts of data, including multiple files with approximate sizes in the megabyte range.	Must have
NF-4	Handling multiple simulations	The platform must be able to handle multiple simulations at the same time, between 2 and 100 simulations (interval values based on requests at Ubiwhere's Urban Platform)	Must have
NF-5	Generating simulations reports	The platform must be able to generate reports on simulation results, assessing information about the requests, such as duration of response and response status codes	Must have
NF-6	Integration with other systems and tools	The platform must be able to integrate with other systems and tools via APIs	Must have
NF-7	Security and Protection of data	The platform must be secure and protect sensitive data to comply with GDPR	Must have

Table 3.3: Non Functional requirements that focus on what the needs of the simulation are.

3.2.4 Epic and User Stories

With the goal of having a better understanding of the requirements, some epic and user stories were created. Each epic story represents a set of actions in a high-level way and it is then broken down into smaller user stories so that all the functionalities and necessities can be easily described.

ES-01: As a user, I want to select some roads and lanes to be closed for some time **so that** I can analyze the impact of the modifications on the city flow and mobility.

US-01: Block road

As a user, I want to select a certain road **so that** it can change its status to blocked

As a user, I want to select a certain period of time **so that** the road can be blocked for that period.

Acceptance Criteria

1. When the user selects a road, its status should be successfully changed to "blocked" in the system.
2. The system should allow the user to specify a valid period of time during which the selected road will remain blocked.
3. Once the specified time period has elapsed, the road's status should automatically revert to its original state.

US-02: Block lane

As a user, I want to select a certain lane of a specific lane **so that** it can change its status to blocked

As a user, I want to select a certain period **so that** a certain lane of a specific road can be blocked for that period.

Acceptance Criteria

1. The user should be able to choose a specific lane on a particular road, and the selected lane's status should be successfully changed to "blocked" in the system.
2. The system should provide the user with the option to specify a valid time period for which the chosen lane of the specific road will remain blocked.
3. After the designated time period ends, the status of the selected lane should automatically revert to its initial state.

ES-02: As a user, I want to change the speed limits and number of parking spots around certain areas of the city **so that** I can analyze the impact of the alterations on all the surrounding areas.

US-03: Change speed limit

As a user, I want to select a certain road **so that** it can be altered

As a user, I want to select a speed limit **so that** the road selected can have its speed limit altered

As a user, I want to select a certain period of time **so that** the road can have its speed limit altered for that time period

Acceptance Criteria

1. The user should be able to select a specific road in the system and successfully alter its speed limit.
2. The system should provide the user with the option to specify a desired speed limit for the selected road.
3. The user should also have the ability to specify a valid time period during which the altered speed limit will be in effect for the chosen road.

US-04: Add the number of parking spots

As a user, I want to select a road **so that** can see and change the number of parking spaces on the curb

As a user, I want to select the number of parking spots **so that** it can be added to the road

As a user, I want to select a certain period **so that** the changes to the parking spots be effective in that period.²

Acceptance Criteria

1. The user should be able to select a road and view the existing number of parking spaces on the curb.
2. The system should allow the user to modify the number of parking spots on the selected road.
3. The user should have the option to specify a certain time period during which the changes to the parking spots will be effective.

US-05: Close number of parking spots

As a user, I want to select a group of parking spots **so that** it can be closed

As a user, I want to select a certain period of time **so that** the changes to the parking spots be effective in that period.

Acceptance Criteria

1. The user should be able to select a group of parking spots in the system and successfully mark them as closed.

²Parking space changes can modify mobility patterns, as it can increase or decrease the number of lanes in the street or sidewalk.

2. The system should provide the user with the option to specify a valid time period during which the selected parking spots will remain closed.
3. Once the designated time period elapses, the closed parking spots should automatically revert to their original state and be available for use again.

ES-03: As a user, I want to create and change some public transportation routes and schedules **so that** I can analyze the impact of the changes on the city's overall mobility.

US-06: Create public transportation route/schedule

As a user, I want to add certain points in the map **so that** a new public transportation route can be created

As a user, I want to select the schedule of the Public transportation (PT) route **so that** a new public transportation route can be created with a set of trips and calendars

As a user, I want to select the number of PT vehicles **so that** a new public transportation route be created

Acceptance Criteria

1. The user should be able to add specific points on the map, indicating the desired locations for a new public transportation route.
2. The system should allow the user to select and set the schedule for the PT route, specifying the trips and calendars associated with it.
3. The user should have the option to select the number of PT vehicles required for the new public transportation route.

US-07: Change public transportation route/schedule

As a user, I want to select a PT route **so that** its schedule or stops can be altered

Acceptance Criteria

1. The user should be able to add specific points on the map, indicating the desired locations for a new public transportation route.
2. The user should be able to select an existing PT route in the system.
3. The system should allow the user to alter the schedule or stops of the selected PT route as desired by the user.

US-08: Delete public transportation route/schedule

As a user, I want to select a PT route from a list **so that** it can be deleted.

Acceptance Criteria

1. The user should be able to access a list of existing PT routes in the system.
2. The user should have the option to select a specific PT route from the list.
3. Once a PT route is selected, the system should allow the user to delete the chosen route from the system.

ES-04: As a user, I want to select a certain zone of the city so that I can block all the roads for specific vehicles and evaluate a possible low-emissions zone.

US-10: Choose the area to be evaluated

As a user, I want to select a certain zone drawing a polygon so that its roads can be blocked

As a user, I want to select a certain period of time so that the roads can be blocked for that time period.

Acceptance Criteria

1. The user should be able to draw a polygon on the map to define a specific zone.
2. The system should recognize the drawn polygon and consider it as the designated area for road blocking.
3. The user should have the option to specify a certain time period during which the roads within the selected zone will be blocked.

ES-05: As a user I want to view all the results of the simulation so that I can better make decisions about the flow of the city/zone.

US-11: Run simulation

As a user I want to run a simulation so that I can see the results

Acceptance Criteria

1. The user should have the ability to initiate the simulation process.
2. The system should execute the simulation using the selected parameters and configurations.
3. The user should be able to observe and view the results of the simulation.

US-12: Save simulation results

As a user I want to save the results so that I can analyze them at a later stage

Acceptance Criteria

1. The user should have the option to save the results of the simulation.

2. The system should provide a mechanism for saving the simulation results in a specified format or location.
3. The saved results should be accessible to the user for analysis and review at a later stage.

3.3 Key Performance Indicators

The thesis exhibits a degree of integration with IN2CCAM, a collaborative initiative involving Ubiwhere that commenced in December 2022. During the course of the thesis, certain KPI were provided, which can assist in directing the focus of the visualization component toward these specific indicators.

KPI	Description	Unit
Energy use	Energy use per kilometre of the vehicle	Joule/meter
Average Speed	The average speed of the vehicle	meter/second
Speed per vehicle type	Average vehicle speed per vehicle types	meter/second
Single vehicle km traveled	Percentage increase in single-vehicle km travelled	meter
Vehicle stops	Number of vehicle stops per vehicle for all vehicle types	Stops
Non-scheduled stops	The number of non-scheduled vehicle stops per kilometer. A non-scheduled stop is recorded is a stop during a trip, e.g. stop for a red light, congestion, or avoiding a collision.	-
Noise levels	Noise level of the city	dB
CO2 emissions	Current emissions of CO2	kg/meter
PM emissions	Current emissions of PM	kg/meter
NOx emissions	Current emissions of NOx	kg/meter
Traffic flow	Number of vehicles per km, record average and standard deviation over time or road length	-

Table 3.4: Table of KPIs from IN2CCAM that are evaluated by the API of the solution developed in the thesis.

3.4 Summary

This chapter covered the process of gathering, processing, and categorization of the requirements for the integrated system implemented, as well as epic and user stories that are auxiliary to the understanding of the needs and necessities that Ubiwhere aims to assist.

Finally, these requirements were used as an assistance to the conceptualization and modeling of the architecture of the integrated system, as well as to vali-

date that the future work follows the proposed objectives in the implementation phase. The requirements also aided the selection of the traffic simulation tool, to guarantee that the best framework for this project was chosen.

Chapter 4

Architecture

Software architecture is a continuous development process throughout the project that aims at finding the best system organization in order to meet the requirements established to provide an overview and complexity of all the components integrated, how they communicate and how data is exchanged, stored and delivered to the end-users.

The architecture of the simulation platform was created and thought using the C4 Model [7]. The C4 model is a lightweight and straightforward approach to visualizing software architecture. This model was created with the objective of helping software development teams describe and communicate software architecture.

The C4 model is done by the creation of four diagrams with different levels of detail:

1. System Context diagram
2. Container diagram
3. Component diagram
4. Code diagram (optional level of detail)

With the goal of planning the architecture, the first three levels of detail were created and explained.

4.1 Context Diagram

A System Context diagram provides a starting point, showing how the software system in scope fits into the world around it. In the Context diagram for the solution, we can analyze how the simulation tool will fit in, using data from Ubiwhere's Urban Platform and receiving instructions via User Interface.

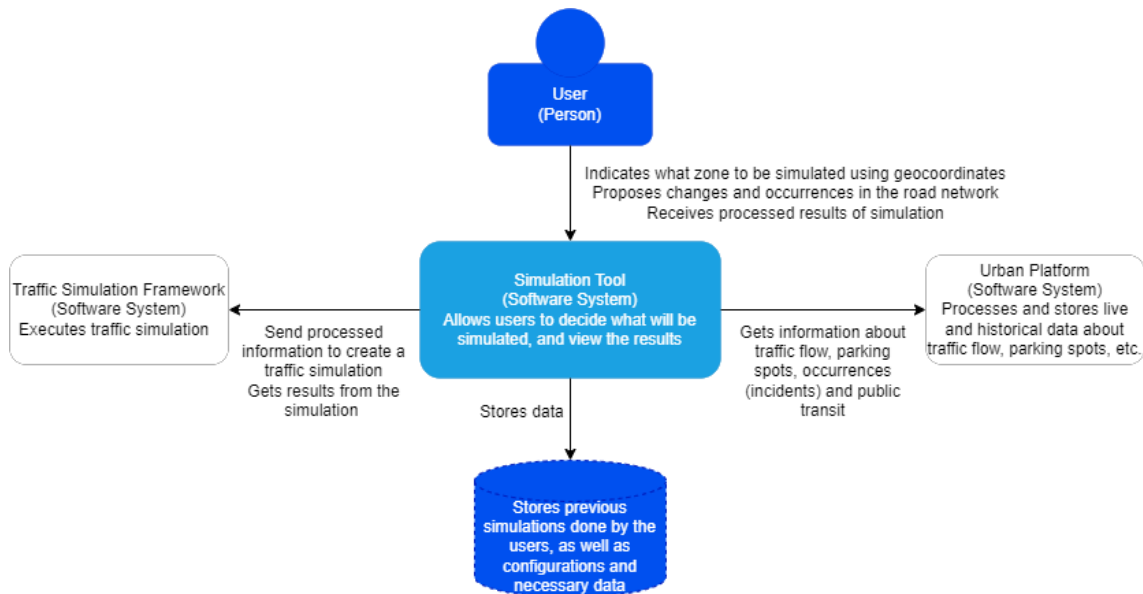


Figure 4.1: Context Diagram of the Integrated System

This diagram presents the architecture of the Integrated System, which comprises various components working together seamlessly. At the center of the system is the Simulation Tool, a powerful software tool that enables users to interact with the system. The Simulation Tool plays a crucial role in retrieving data from the Urban Platform provided by Ubiwhere.

Once the data is obtained, the Simulation Tool utilizes a carefully chosen Framework, specified in Section 5.2, to perform simulations. These simulations aim to replicate and analyze various urban scenarios based on the acquired data. The chosen Framework acts as the engine that drives the simulation process, ensuring accurate and reliable results.

To ensure the preservation and accessibility of the simulation results, the system incorporates a database. This database serves a dual purpose: firstly, it stores the simulation outcomes for future use and analysis. This allows users to refer back to previous simulations, compare results, and derive valuable insights. Secondly, the database also stores essential parameter configuration information. For example, it retains data related to the road network graph, enabling the system to expedite the simulation process by leveraging pre-stored information.

By utilizing this integrated system, users can effectively leverage the Simulation Tool to interact with Ubiwhere’s Urban Platform, retrieve relevant data, simulate urban scenarios using the chosen Framework, and store the outcomes and configuration details for efficient and accelerated future simulations.

4.2 Container Diagram

A Container diagram zooms into the software system in scope, showing the high-level technical building blocks. In this diagram, we can detect how the tool

will function, via a single-page application making API calls to the Simulation Tool which will run the simulation according to the data used, and store it in a database for future use.

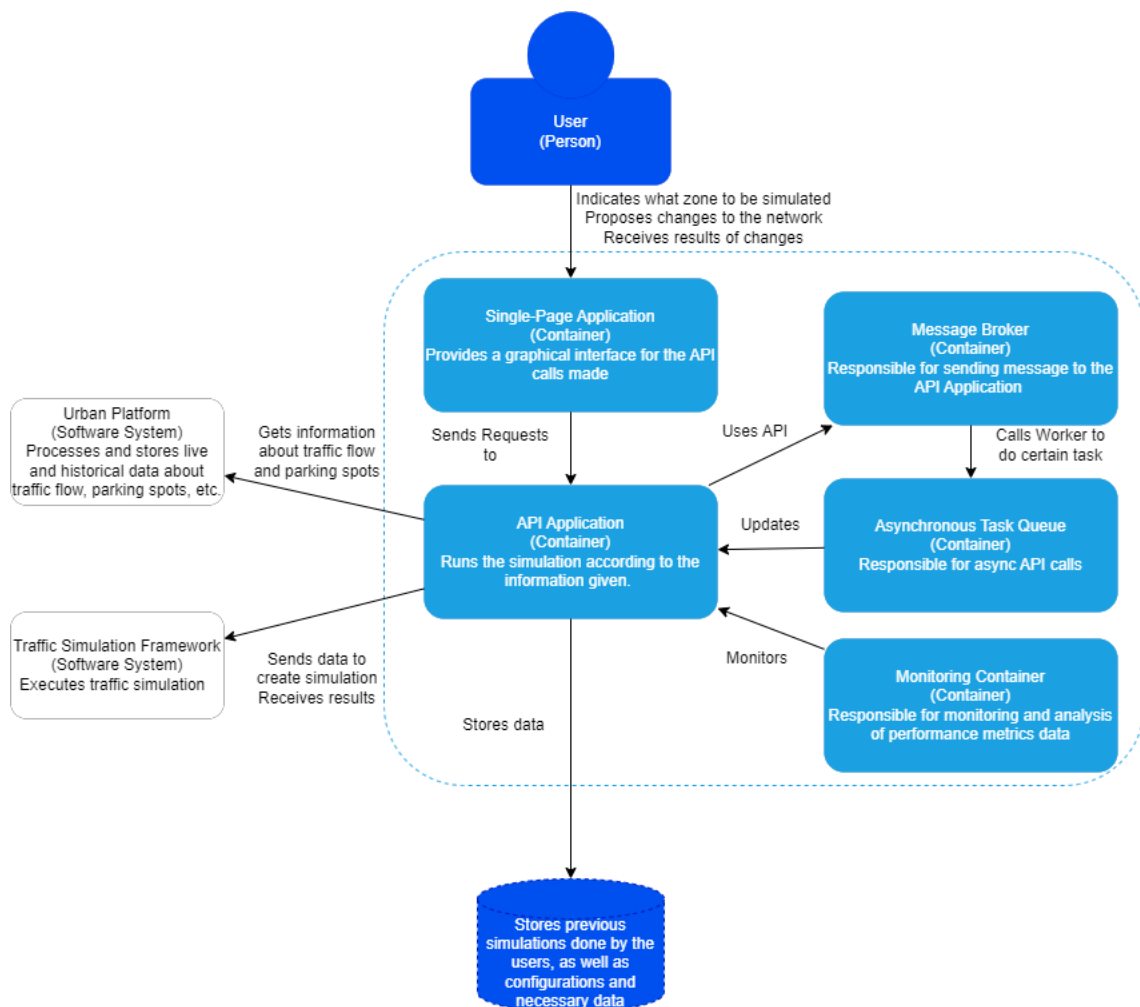


Figure 4.2: Container Diagram of the Integrated System

In a more detailed diagram, we can observe five primary containers, namely the Single-page Application, Message Broker, Asynchronous Task Queue, API Application, and Monitoring container.

The Single-page Application container takes charge of providing a user-friendly graphical interface that simplifies the visualization and creation of simulations. It enables users to interact with the system seamlessly.

The Message Broker container plays a vital role in routing API calls, ensuring efficient organization and preventing any message loss. By utilizing a message broker, the application gains enhanced scalability, setting the foundation for future expansion.

The Asynchronous Task Queue container facilitates the execution of asynchronous API calls. For instance, it allows for data retrieval from the Urban Platform during periods of lower request volumes. This decouples the system and provides

support for background tasks that may be large in size, time-consuming, or numerous in quantity.

The Monitoring container takes on the responsibility of collecting essential data for evaluating various metrics related to scalability and overall system performance. This data aggregation enables comprehensive analysis and assessment of the integrated system.

Lastly, the API Application container is responsible for processing data, executing simulations, and storing the results in the database. It serves as the core component of the system, handling crucial tasks related to data manipulation and simulation execution.

Together, these containers form an interconnected and robust system that empowers users to interact with the integrated platform effectively. The system's architecture ensures a user-friendly interface, efficient routing of API calls, asynchronous task management, comprehensive monitoring, and seamless execution of simulations, contributing to an enhanced user experience and reliable system performance.

4.3 Component Diagram

A Component diagram zooms into an individual container, showing the components inside it. In this final diagram, we can view the components that compose the simulation API. It is possible to visualize how the API interacts with the Urban Platform, as well as the exchanges with the asynchronous task queue and the message broker.

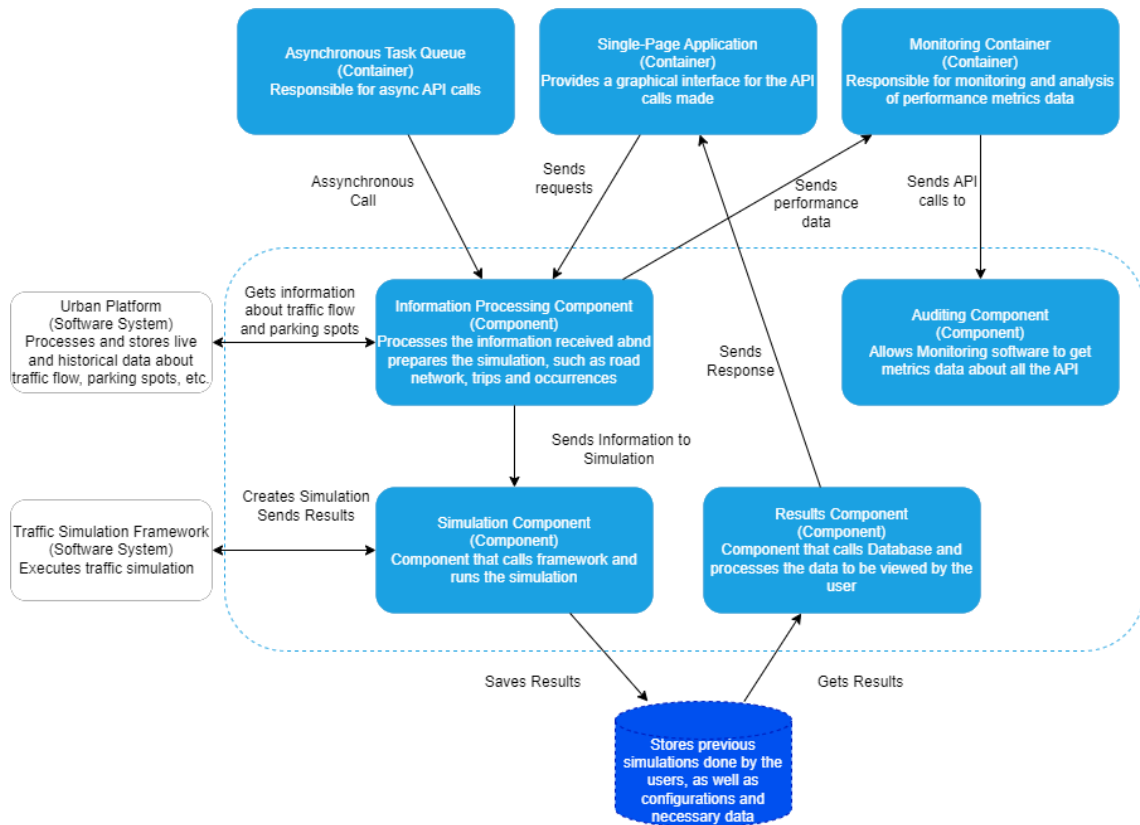


Figure 4.3: Component Diagram of the Integrated System

The API application container consists of four distinct components: the Information Processing Component, Simulation Component, Results Component, and Auditing Component.

The Information Processing Component plays a vital role in processing the data received from the Urban Platform. This includes historical traffic flow data, public transportation routes, and any user-provided information for network modifications. By effectively processing this data, the component ensures that accurate and relevant information is available for analysis and simulation.

The Auditing Component is responsible for collecting performance metrics from the API. These metrics are crucial for evaluating the scalability and overall performance of the simulations. By analyzing these metrics, one can gain insights into the limitations and areas of improvement within the Integrated System.

Within the Simulation Component, simulations are executed using the processed data from the Information Processing Component. The results of these simulations are then stored in a database for future analysis. This component needs to meet specific requirements, such as the ability to run multiple simulations simultaneously, ensuring efficient and effective simulation execution.

The Results Component undertakes the processing of all stored simulation data. This processing enables the utilization of the data for various visualizations and displays. From map visualizations to time-series graphs and key performance indicator tables, this component ensures that the simulation results are transformed

into actionable and meaningful representations.

Overall, these components within the API application container form a cohesive system that empowers users to efficiently process data, conduct simulations, analyze results, and evaluate the performance of the Integrated System.

4.4 Technologies

In order to implement the architecture described in this chapter, the thesis required decisions on the technologies to be used. As such, the technologies used in the message broker, in the asynchronous task queue, and in the simulation API, must be selected.

The message broker framework used for this project was established with Ubiwhere, in order to maintain consistency in tools used by other products, such as the Urban Platform allowing for a seamless integration and preservation of work.

The choice of a traffic simulation tool for the simulation API was selected using the requirements gathered, to ensure that the objective could be accomplished with the chosen framework.

The selection of the simulation tool was, subsequently, decide what programming language was mainly used in the project. As such, the asynchronous task queue was also decided to fit with the programming language used in the integrated system, so that the integration of the traffic simulation happens in the most seamless way possible.

In conclusion, there is an existent dependence on the rest of the architecture in the choice of the traffic simulation framework, as it is the most critical component of the integrated system and the direction of the project. As such, the next chapter focuses on the numerous types of traffic models and the state of the art of traffic simulation, with a comparative analysis of all the frameworks.

Chapter 5

State of the Art

The state of the art provides an understanding of the research that has been done so far about a certain topic. It is necessary to be aware of what others have been working on, avoiding past mistakes and supporting future decisions.

This chapter provides information about the development of traffic simulation as a research field in the past years. The crucial subject matter is the types of traffic simulations, as well as the tools used to create those simulations. The chapter ends with a brief reflection on the knowledge acquired.

5.1 Traffic Simulation

A simulation is a model that mimics an operation or proposed system, providing help in decision-making by being able to test various scenarios and even process changes [8]. Simulations can be used when the real system is inaccessible, too dangerous to assess, or even in theory stages.

The purpose of creating a simulation of a traffic system is to generate data in a faster and more economical way. Instead of testing new ideas on how to manage traffic in the real world or collecting data using sensors, a model can be used to predict traffic flow. Traffic simulation can also be used to create various traffic conditions that can be subsequently added to a data set to train and test autonomous vehicles [9].

The focus of traffic simulation is to discover how the evolution of traffic occurs, given initial vehicle states, a behaviour model, and a road network. This question can be answered using mathematical models and simulations of traffic flow [10]. These models and simulations are crucial for a comprehensive investigation of the urban transportation system in a safe and suitable environment [11]. These traffic system models can be classified into three categories: microscopic models, macroscopic models, and mesoscopic models. Each category has its advantages as well as its disadvantages, depending on the initial data and the final objective of the simulation.

5.1.1 Microscopic Models

Microsimulation models track individual vehicle movements, relying on random numbers to generate vehicles, select routing decisions and determine behavior [12]. These models can be car-following models, cellular automata (CA) models, and lane-Changing models[13]. These models attempt to explain the behavior of individual vehicles and drive [14]. Currently, according to [15], the following modeling programs are widely used: PTV Vissim, AIMSUN, SUMO, FREESIM, CORSIM, Paramics.

Car-following models Car-following models are developed based on human driving behavior, which defines interactions with the following vehicle. These models assume that the traffic flow consists of scattered particles and model detail interactions among cars. These models represent the position and speed of each car through continuous-time equation [10].

Simple car-following models can be defined by the equations:

$$s_i = x_i - x_{(i-1)} - l_i$$

$$\Delta v = v_i - v_{(i-1)}$$

where the i -th vehicle follows the $(i-1)$ -th vehicle, the x_i its position along the road, the v_i its velocity and l_i its length. The value of s_i represents the distance between the front of the i -th vehicle with $(i-1)$ -th vehicle.

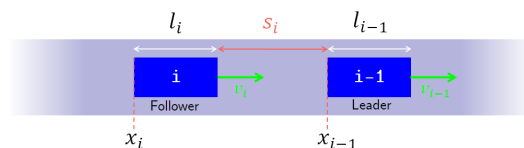


Figure 5.1: Visual representation of the simple car-following model example created by Himite[10].

Cellular Automata models A cellular automaton is a collection of cells on a grid of specified dimensions that evolves through specified steps according to a set of rules based on the states of neighboring cells. The neighborhood of the cell is the parameter where the rules apply to the cells [14]. Cellular Automata models are computationally efficient and can simulate a large group of vehicles on a large road network, represented by a grid of cells. However, due to its discrete nature, this model can only create a limited number of real-world traffic behaviors [10].

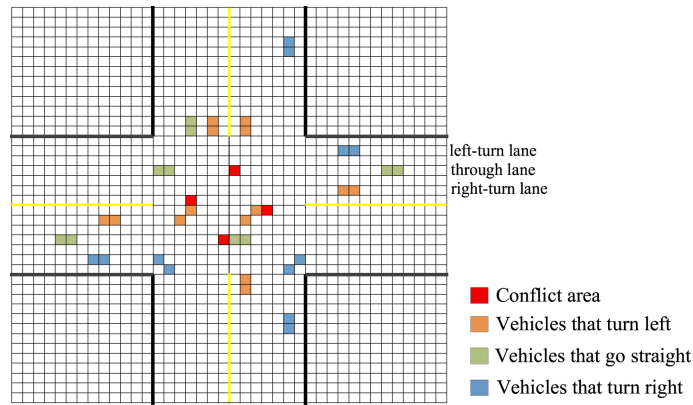


Figure 5.2: Example created by Zhao et al.[16] using cellular automata to simulate a 4-way intersection with three entrance lanes, where each car is represented by 1x2 cells.

Lane-Changing models A number of lane-changing models have been proposed over the past decades. These models can be divided into 2 categories: analytical and data-driven models. The analytical focus is more on rule-based models and discrete-based models. The Data-driven models proposed are based on Neural Networks, Support Vector Machines, and Bayesian Filtering [17][18]. These models can be divided into 2 distinct parts: the lane-changing decision model (LCD) and the lane-changing implementation model (LCI). The LCD is responsible for the individual decision of the vehicle. If the chosen decision is to change lanes, then the LCI model will be executed, otherwise, the vehicle can implement other microscopic models mentioned previously.

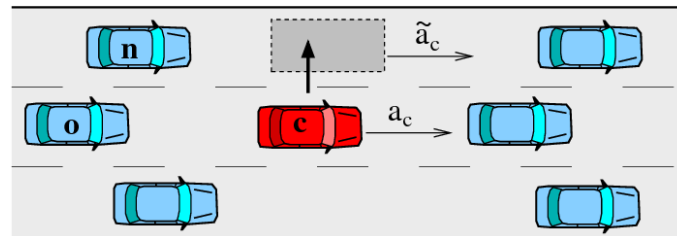


Figure 5.3: Sketch of the nearest neighbors of a central vehicle c considering a lane change to the left. The new and old successors are denoted by n and o , respectively. Accelerations after a possible change are denoted with a tilde[19].

5.1.2 Macroscopic Models

Macroscopic modeling presents traffic flow as a combination, measured in terms of components such as speed, flow, and density, considered essential characteristics of the movement. Macroscopic modeling is described as having a low detail of transport infrastructure and traffic flows. These models study the behavior of cars on a large scale. According to Dorokhin et al.[15], the main software products used in modeling are Visum and Saturn.

Continuum traffic model One of the earliest car-following macroscopic traffic models extensively applied to traffic flow on the straight lane. This model considers traffic as a compressible fluid and replicates its behavior on a macroscopic level through aggregated state variables such as flow and density[20].

It is defined by the equation:

$$\frac{dx_n(t + \tau)}{dt} = V(\Delta x_n(t))$$

where $x_n(t)$ is the position of the n th vehicle at time t , $\Delta x_n(t)$ is the optimal speed and τ is the delay time.

Full Angular Velocity Difference model This model was proposed by Xue et al.[21] and was created with the goal of studying the influence of curved roads and angular velocity difference in generating congested traffic.

This model uses assumptions from the continuum traffic model. It was assumed, based on the previous mode, that a driver adjusts the vehicle angular velocity $\frac{d\theta_n(t)}{dt}$ to reach the optimal angular velocity $\omega_e(\Delta\theta_n(t))$ in the delay time τ according to the observed angular spacing $\Delta\theta_n(t)$. The full angular velocity difference model (FAVD) on the curved road is proposed as follows:

$$\begin{aligned} \frac{d\theta_n(t + \tau)}{dt} &= \omega_n(t + \tau) = \omega_e(\Delta\theta_n(t)) \\ &\quad + \lambda(\omega_{n+1}(t) - \omega_n(t)) \\ \omega_e(\Delta\theta_n(t)) &= \frac{V(\Delta x_n(t))}{r} = \frac{V(r \cdot \Delta\theta_n(t))}{r} \end{aligned}$$

where $\theta_n(t)$ is the angular displacement of the n th vehicle, $\Delta\theta_n(t)$ is the angular displacement difference (angular spacing) of two vehicles and $\omega_n(t)$ corresponds to the angular velocity of the n th vehicle on the curved road at time t . τ is the delay time and λ is a sensitivity to the angular velocity difference $\omega_{n+1}(t) - \omega_n(t)$ called the relative angular velocity. Road radius r is an invariant constant due to just considering the circular motion.

After studying this model, Xue et al. [21] suggest that the radius of a curved road has a great impact on the formation of traffic jams.

5.1.3 Mesoscopic Models

Mesoscopic models have an intermediate position between macroscopic and microscopic models. A distinct feature is the conservation of a high level of detail in the description of the simulation street-road network but simplified reproduction of the dynamics of the traffic flow. These models try to explain behavior in groups of cars, classified according to their speed [14]. The disadvantages of mesoscopic models include the insufficient development of theory and software [15].

POLARIS mesoscopic traffic flow model This mesoscopic model is an attempt to obtain enough for applications that require microscopic data, for example, estimating fuel consumption, while still inheriting the computation efficiency of macroscopic traffic flow models. The model is macroscopic at the link level and microscopic at the node level.[22]

5.1.4 Analysis

Researchers need to be aware of the purpose of their simulation and choose a simulation model that supports the required level of detail for dealing with their research objectives. Going into more detail than necessary can make a simulation model complex and also requires more input data [23].

As mentioned previously in the objectives, the goal is to create a traffic simulation platform that can analyze the impact of traffic jams and incidents on the road, as well as the flow of different types of vehicles. Therefore, it is necessary to choose a model type that can differentiate between private and public vehicles, as well as the management of lanes, with the possibility of changing its nature in any partition.

Mesoscopic models, which operate at an intermediate level of abstraction, fail to capture the precise behaviors and attributes of individual vehicles required to analyze the impact of traffic congestion, incidents, and varying vehicle types. Consequently, they do not possess the level of detail essential for our study's comprehensive analysis.

Similarly, macroscopic models, which focus on macro-level traffic flow patterns and disregard individual vehicle interactions, lack the granularity necessary to examine specific lane management strategies and their effects on traffic dynamics. Our study aims to delve into the complexities of lane management, including the ability to alter lane characteristics within distinct partitions, necessitating a more fine-grained approach.

In light of these requirements, the use of microscopic models becomes essential. These models enable us to capture and leverage individual vehicle data, which is pivotal for effectively implementing all the desired functionalities

Therefore, the adoption of microscopic models is critical for our simulation platform to achieve accurate and comprehensive results.

5.2 Frameworks

In order to create the simulation models, numerous frameworks were built, each focusing on a problem that was found in the ecosystem of traffic simulations. In this section, some frameworks were analyzed with the objective of understanding what is the best framework for the use case

PTV VISSIM

PTV VISSIM is a commercial microscopic traffic simulator, created by the PTV Group in 1992, developed with the objective of simulating individual vehicle movements, focusing on creating an accessible interface for the users. More recently was added support for pedestrian simulation, with the objective of simulating the interaction between vehicles and pedestrians [24]. The framework's main functionalities are the pedestrian, car, and public transportation, the possibility of choosing what microscopic model to use, like the car following model, lane changing model as well as vehicle speed parameter models [25].

The PTV VISSIM simulator is one of the most popular choices in the traffic simulation area, but also used in many other areas of application, such as in-route optimization, route planning, green logistics, and active mobility. The main benefit of using this platform in the project would be the seamless integration between the vehicle simulation with pedestrian, bus, and train transportation routes, and even non-motorized vehicles. Another PTV VISSIM asset would be the opportunity to calculate the emissions created, as well as the fuel consumptions of each specific node [26][27].

The biggest disadvantage of the framework is the incompatibility with OSM, which is an important dataset used in Ubiwhere, such as the Urban Platform. Although other simulators created by PTV Group have the possibility of drawing the network using .OSM files, PTV VISSIM does not have the technology. Another potential disadvantage is that the focus given by the creators to create a user-friendly interface may deviate the efforts from the scripting part of the framework. The framework appears to have a smaller community or receive fewer dedicated efforts, resulting in limited availability of tutorials and open projects.

As a proof of concept, a tutorial of the framework was followed to gain an understanding of its strengths and limitations. The tutorial introduces a simple intersection, that can be built in a diverse number of languages, like Python, C#, C++, Java, and MATLAB.

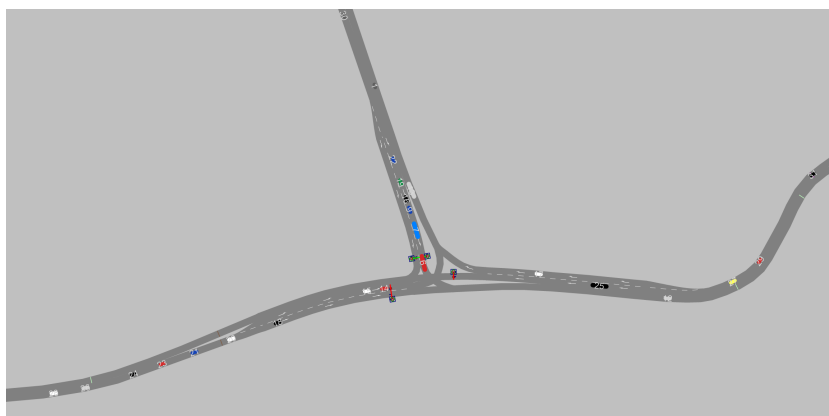


Figure 5.4: Example of an intersection with traffic lights, created via scripting, and visualised in PTV VISSIM.

Due to the lack of licensing, it was not possible to modify the script with the objective of testing the framework limitations to the fullest.

SUMO

Simulation of Urban Mobility (SUMO) is an open-source microscopic traffic simulator [28], created in 2001 and continuously developed by the German Aerospace Center and community users. This framework was created with the intent of supporting the traffic simulation community with a free tool into which custom algorithms can be implemented.

SUMO framework allows users to create simulations of the traffic flow by defining each vehicle with an identifier, departure time, and route, with the possibility of associating extra variables to the vehicle, [25]. The tool also has multiple applications like Traffic Control Interface (TraCI) which provides statistical data and direct access to the elements, or OSM Web Wizard, which can create a network based on an OSM file, or even OD2Trips, which converts Origin Destination (O/D) matrices to single-vehicle trips that assign vehicles to edges of the road network as departing and arrival position [29]. This framework can model various types of personal vehicles, pedestrians, and public transportation, and limit lanes to special vehicles [30].

However, SUMO is not agent-based like many other platforms, but due to its popularity, it is possible to integrate other frameworks, like FLOW, to create an agent-based simulation, if needed. SUMO's main features that are more interesting for the objectives of the project consist of Multimodal Traffic, automated driving, Online Interaction, and Traffic Management. The biggest advantage of this framework is that has a lot of community support, as well as very extensive documentation, which makes it more friendly to learn and use.

In an attempt to observe the framework, it was tested the example of city mobility from SUMO tutorials, where X number of taxis are called to a bus station where the pedestrian is picked up.

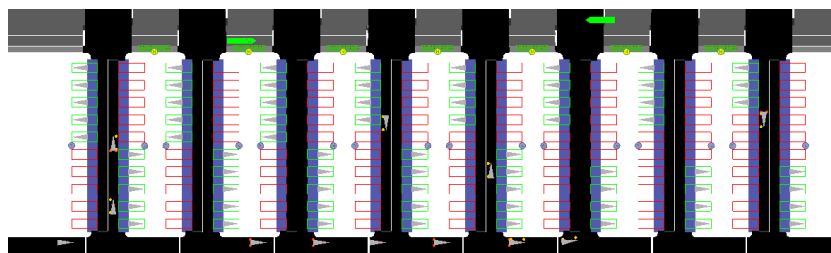


Figure 5.5: Example of a taxi and bus station created by Sumo with Python and visualized in SUMO GUI.

AIMSUN NEXT

AIMSUN NEXT is a commercial multi-modal traffic simulator [31], created in 1997 by Yunex Traffic, with the objective of giving deep insights into mobility:

past, present, and future, using simulation and machine learning. This framework enables the modeling of small to large transportation networks through macroscopic, mesoscopic, and microscopic modeling. AIMSUN Next also allows the automation of operations, such as modifying model parameters, importing and exporting data, and launching multiple simulations or assignments, using C++ and Python Scripting [25].

AIMSUN is mainly used for its features like modeling Connected Autonomous Vehicles (CAV), mobility planning for cities, highways, and regions, real-time transportation management, and even shared mobility.

One disadvantage of this platform is the incapability of scripting using the free trial period, which prevents testing functionalities like O/D trips, or creating networks based on OSM files, and limits the information that was possible to gather regarding the requirements. As a proof of concept, some examples were run on Aimsun Next with the object to understand its limitations.

As a proof of concept, some examples were run on Aimsun Next with the objective of understanding its limitations.

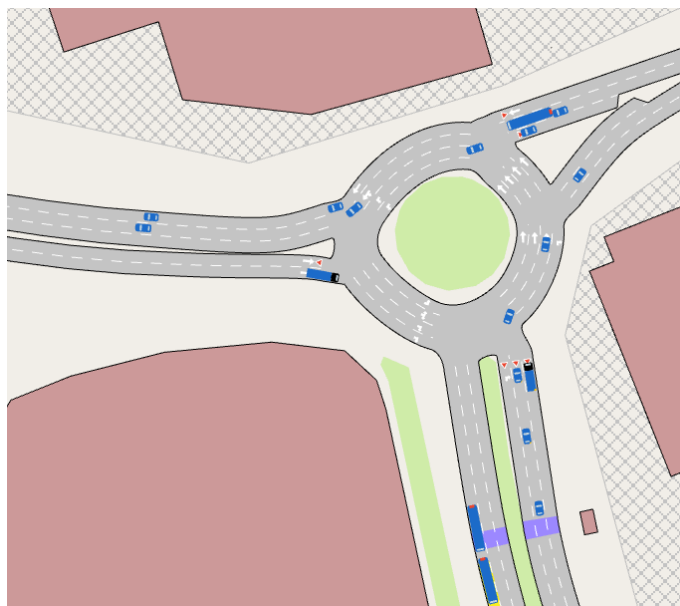


Figure 5.6: Example of a roundabout visualized in Aimsun Next.

As the license available for Aimsun is a trial version, it is not possible to create and run script files.

MATSim

MATSim or Multi-Agent Transport Simulation is an open-source, activity-based extendable multi-agent simulation framework[32] that is microscopic with regard to demand modeling but mesoscopic with respect to traffic dynamics [33]. It was created between 1999 and 2004 by Kai Nagel and his interest in improving his work with the TRANSIMS project.

MATSim is based on the co-evolutionary principal, which means that every agent, represented by their plans, repeatedly optimizes its daily activity schedule in competition for space-time slots with other agents [34]. These agents can be generated based on user input, or randomly [23].

This platform was used to create Ziemke et al.[35] MATSim Open Berlin scenario, where the metropolitan region of Berlin is represented realistically, only using open source data and tools.

This tool, due to its open-source nature, has a lot of community support, in scripts like OSMNetworkReader, which can create a network only using an OSM file, or even matsim-tools which is a Python library created with the objective of expanding the reach of the framework [36].

MATSim's main capabilities are fast dynamic and agent-based traffic simulation, private and public Traffic simulation, and its support of large scenarios. One disadvantage of this platform is the way it simulates streets, which may limit some interactions with the simulation, such as limiting the number of parking spots or altering the number of lanes on the road.

In order to better evaluate this framework, It was decided to install and try to execute the initial example. Following the tutorial, it was possible to run a simple example with 15 roads and 100 cars. Although MATSim does not have the ability to show a visualization of the network or the simulation in progress, it installed VIA, a Simunto platform, which allows to analysis MATSim results, directly supporting the loading of the typical input and output files.

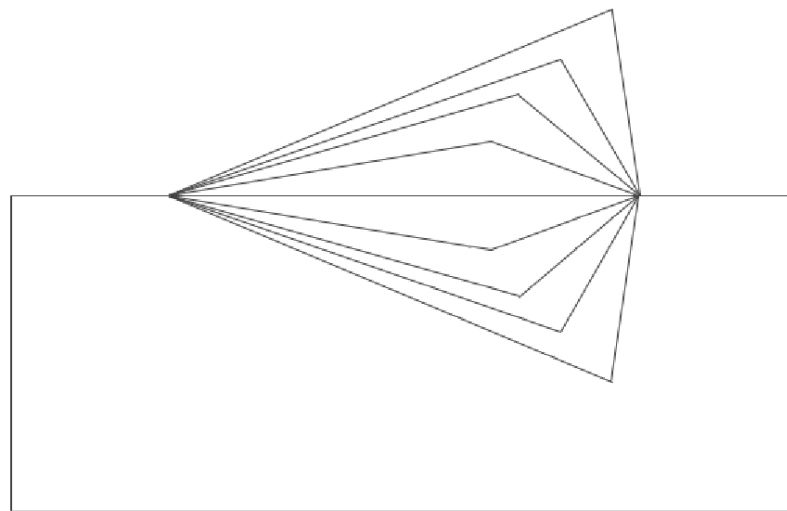


Figure 5.7: Example of MATSim network visualization, consisting of 15 roads

POLARIS

POLARIS is a publicly available agent-based software framework, created in 2013 by the Argonne National Laboratory. This framework was originally developed with the objective of helping the U.S. Department of Transportation (DOT) ana-

lyze transportation system management strategies [37]. The motivation behind POLARIS then evolved into finding a way to combine different traffic-related modeling aspects into a single framework that otherwise requires several separate standalone software applications [23].

POLARIS mesoscopic models allow focusing mainly on large-scale transportation scenarios, with the possibility to use connected and automated vehicles.

Some applications where the tool was used include, Connected and automated vehicles (CAVs), Mobility as a Service (MaaS), Transit and Freight systems, Charging infrastructure, and System level optimization.

Although POLARIS is considered publicly available, it is necessary to request a license, which delayed the testing part of the framework, as access to the documentation only is given when the license is accepted.

Traffic Modeller

Traffic Modeller (TraMod) is an open-source, client/server-based solution for responsive traffic modeling, created in 2020 by the University of West Bohemia [38]. This framework was created with the objective of presenting an alternative software solution for responsive traffic modeling, with features such as open source code, advanced graphical user interface (GUI), and thorough testing of the backend and frontend, that the creators believe have an advantage in comparison to other modeling frameworks.

One of its main case studies was the pilot user, the city of Pilsen, where using the API it is possible to block roads, or limit traffic and visualize the results of the flow in the rest of the network.

One disadvantage of this framework is the limited documentation of the functionalities, and how to implement them, as well as the lack of community using this software, which could be problematic if some specific framework error would appear. Another disadvantage is the lack of many functionalities that would be necessary for the project, such as public transportation or the differentiation between types of vehicles.

Upon conducting a thorough search for documentation, it was found that there is no tutorial available for utilizing the framework or any guidance on creating nodes and edges using an OSM file. Despite sharing similar objectives to the desired outcome and having received recent updates as of January 12, 2022, this framework is lacking in documentation and community usage.

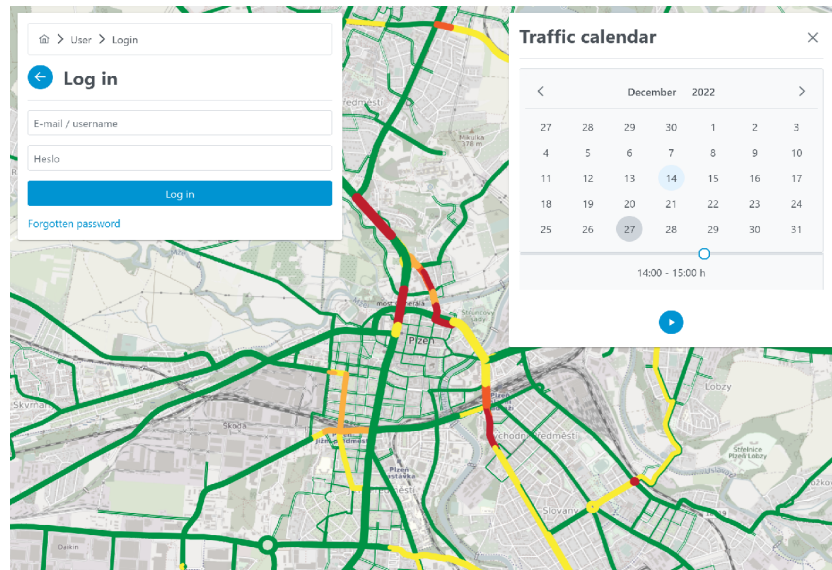


Figure 5.8: Image from case-study of Pilsen using the framework Traffic Modeller in [39]

FLOW

Flow is a computational framework for deep reinforcement learning and control experiments for traffic microsimulation¹. It was created in 2018, C. Wu et al. [41], with the objective of facilitating the research, as well as helping with the understanding of the implementation of automated vehicles, as well as the complexity of large-scale transportation systems.

This tool is a traffic control benchmarking framework. It provides a suite of traffic control scenarios (benchmarks), tools for designing custom traffic scenarios, and integration with deep reinforcement learning and traffic microsimulation libraries.

One of the advantages of using Flow is the ability to work with other above-mentioned frameworks, such as SUMO, as is preferred according to the creators, and even Aimsun, expanding the possibilities of both frameworks in agent-based simulation and mixed autonomy environments.

¹Reinforcement learning is a machine learning training method based on rewarding desired behavior and punishing undesired ones [40]. This approach situates the agent in an environment where beneficial activity and non-beneficial activity are clearly defined

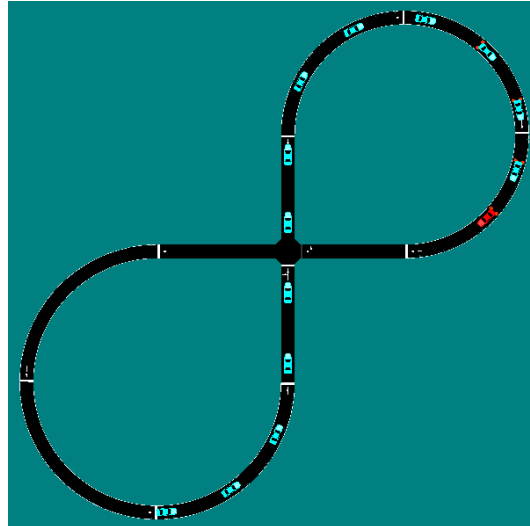


Figure 5.9: Image from Flow's built-in traffic networks, "Figure Eight". The goal is to maximize the system-wide velocity for fourteen vehicles, which necessitates spacing the vehicles so that they do not run into conflicts at the merging points. It is possible to change the rate of autonomous vehicles to human drivers.

5.2.1 Comparative Analysis

To comprehend the strengths and limitations of all the frameworks, a series of experiments were conducted. These tests focused on fundamental tasks that held significance for the subsequent phases of the project. While these tests did not ensure that the chosen framework would fulfill every requirement, they provided valuable guidance in selecting the most suitable group of frameworks.

Experiment 1 - Create a Network based on OSM

After doing the tutorials on each framework, it was decided to experiment with some functionalities required for the project, such as OSM, to understand the level of difficulty of creating a simple network based on the map, as well as the level of detail of the map created.

The framework must be able to create the network of roads, given any section of the OSM, either a file of type .osm or the bounding box of said section. For this test, the zone chosen was the Polo2 of Universidade de Coimbra, which has 2-way roads, a rectangular roundabout, and a 3-way intersection. Using this zone of the city it is possible to experiment with a various number of functionalities that will be needed for the proof of concept.

The first framework that was tested was PTV VISSIM, which although considered feasible by PTV Group, it was only found possible to do in PTV Visum, which is the macroscopic framework from the group.

SUMO had the most straightforward implementation, using OSMWebWizard, a

SUMO solution where the user can create a network based on an excerpt of the map.

For the creation of the network in Aimsun, an OSM file was created using the official OSM website was used and visualized in the Aimsun application.

The visualization of the MATSim was more complicated, as the framework does not provide an interface. As so, VIA was installed, a Simunto application, intended for the visualization of MATSim output data, as well as other large spatio-temporal datasets.

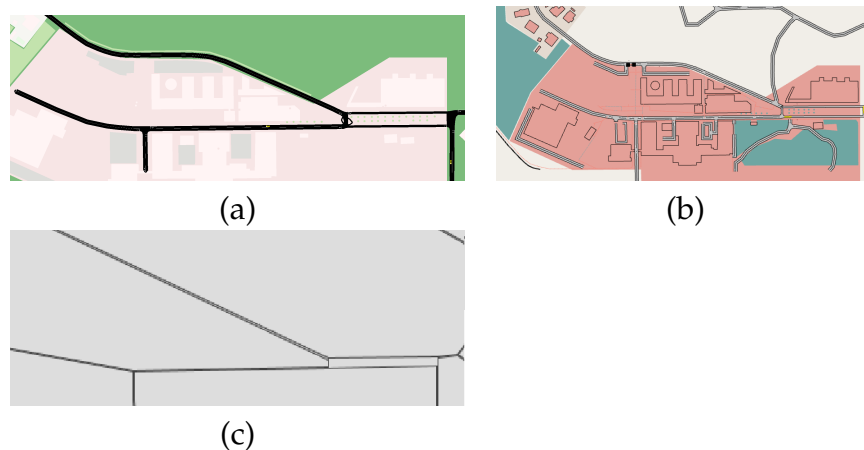


Figure 5.10: Excerpts taken from a) Sumo, b) Aimsun and c) MATSim, using OSM, of Polo2 of Universidade de Coimbra.

Experiment 2 - Run a Simulation

After analyzing what frameworks were capable of creating a network using OSM files, it was decided to run a new experiment.

The second test that was executed was to assess how fast could the algorithm run depending on the size of the network, the frequency period of car spawns, and even the number of simultaneous simulations.

The goal was to compare how the simulations were created for each framework, what differs in the variables and data necessary to run the simulations, and how it all affected the duration of the simulation. The frameworks that were decided to test were SUMO, MATSim, and Aimsun, as they were the tools that supported the creation of the road network based on an OSM file and/or OSM bounding box.

In this stage of the project, some problems arose regarding Aimsun and its licensing. Until this point, every tutorial and test done with the tool was using the 30-day free trial, which gave limited access to functionalities, such as scripting. In order to combat this problem, It was asked via e-mail, a request to receive a student license, which would grant the use of all the functionalities, and as so, allow the continuation of the evaluation of Aimsun.next.

Virtual Machine

This experimentation was created in an Ubuntu 22.04 LTS environment using a Virtual Machine (VM) using VMware Workstation 17 Player, for the sake of isolation, reproducibility and compatibility.

Environment	Ubuntu 22.04 LTS
Memory RAM	4GB
Processors	2
Hard Disk	512GB
/boot	512 MB
/	502GB LVM
SWAP	1 GB
Network	NAT

Table 5.1: System details of the image used to create the virtual machine

SUMO

In order to do this study with the SUMO framework, the network file received from the previous experimentation was used. This network file is composed of 32 different types of nodes, each with a different priority, number of lanes, speed, and the entities allowed to use that segment. This file then specified all the edges, lanes, junctions, connections and roundabouts, for roughly 37000 lines of markup language XML, creating a single file of 3.8 MB.

In order to populate the simulation with vehicles, it was necessary to create random trips around the network. SUMO has a script in its tools called *RandomTrips.py*, which is a Python script, which can receive a network as input, a time frame and the period of transit and creates an XML file with a number of trips. This file is composed of a number of trips, composed by id depart time, with the origin and destination coordinates.

In order to create an experiment with the same number of vehicles during the same time period, a simple function was used to calculate the value of the period:

$$N_{vehicles} = \frac{(T_{end} - T_{start})}{P} + 1$$

Where $N_{vehicles}$ is the number of vehicles that are wanted, T_{end} the time in seconds of when the simulation should end, T_{start} the time in seconds of when the simulation should start, P the value of the period that is wanted. The subtraction of 1 is used as a counter-balancing method to the script, as it includes 0 in the creation of the cars, creating $N_{vehicles} + 1$.

Using the values of a 24-hour period, we get to the equation:

$$P = \frac{86400}{(N_{vehicles} - 1)}$$

Running the simulation, we get a statistical report composed of 5 main points: Reason, Performance, Vehicles, Teleports and Statistics. Using this statistical report we were able to create the following table:

Network (MB)	Vehicles	Processes	Duration (s)
3.8	10000	1	10.10
3.8	10000	2	14.63
3.8	10000	3	25.47
3.8	25000	1	27.77
3.8	25000	2	39.12
3.8	25000	3	64.50

Table 5.2: Different runs made in SUMO

MATSim

For the creation of the network file used in the Image 6.1 c), it was utilized a Java project around the functionality *OsmNetworkReader*, which formed a 208KB XML file formed only of nodes and links.

One file that is essential for the execution of the simulation is the *config.xml*. This file is composed of the network, the plans, the output directory, the number of iterations, the duration and the fitness calculation for the co-evolutionary algorithm.

As no function for the creation of individual plans was found, a simple Python script was developed, which would output an XML file with N activities of individuals going from "home" to "work" and back "home", at different times of the day.

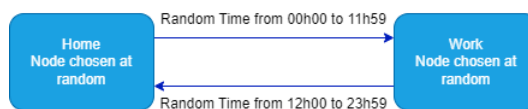


Figure 5.11: Representation of the activity plans created for the experimentation.

This script was responsible for choosing two random nodes from the network file, and arbitrarily deciding which tag "home" or "work" would be set in that node, as well as the timestamps for each event in the day.

The duration of the simulation would be set in the config file, to be a full day, 24 hours, to be consistent with the other experiments. An arbitrary number of iterations (11) was employed with the objective of understanding if the number of iterations affected the overall duration process of the simulation. It was found, that the first iteration uses more time than the other 10 iterations, as it needs to load the network and activities for the whole simulation.

The fitness calculation values were not altered, as it was not the focus of the experimentation.

After running the simulations, a folder with multiple files of varying sizes is created with information about the simulation, iterations and plans. In order to

obtain the values for the duration of each simulation, another Python Script was created, which using Pandas data frame was able to extract the initial and final timestamps and calculate the total duration, and duration per iteration.

After some simulations, it was possible to create the following table:

Network (KB)	Vehicles	Processes	Duration (s)	Per iteration
208	10000	1	43.0	3.91
208	10000	2	52.0	4.73
208	10000	3	88.3	8.03
208	25000	1	77.0	7.00
208	25000	2	97.5	8.86
208	25000	3	175.3	15.93

Table 5.3: Different runs made in MATSim

5.2.2 Analysis

Before choosing which framework to create the simulation engine, we must analyze all the frameworks and their functionalities based on the requirements elicited previously.

	VISSIM	SUMO	AIMSUN	MATSim	POLARIS	TraMod
License	Commercial	Open-Source	Commercial	Open-Source	Open-Source	Open-Source
Model Type	Microscopic	Microscopic	Microscopic	Mesoscopic Microscopic	Mesoscopic	Microscopic
Model Used	-	Car-Following Agent-Based	-	Agent-Based	POLARIS Algorithm	Car-Following
Script	Yes	Yes	Yes	Yes	No	Yes
Doc Access	Limited	Free	Limited	Free	Limited	Free
Last Updated	28 Sep 2022	5 Oct 2022	8 Feb 2022	26 Sep 2022	Not Found	12 Jan 2022
Github Stars	-	1.7k	-	393	-	1
Agent-Based	No	w/Flow	w/Flow	Yes	Yes	Yes

Table 5.4: Summary of reviewed simulators

After analyzing Table 5.4, it is possible to find some differences between the frameworks, if it has an open-source or commercial license, what models are used in the simulation, and even when the framework was last updated. Although this piece of information alone isn't beneficial in choosing what framework should be used, combined with other information, it can be useful to make the final choice.

Id	Title	PTV VISSIM	Aimsun	SUMO	MATSim	POLARIS	TraMod
F-1	Create Roadmap based on OSM data	N	Y	Y	Y	-	-
F-2	Process historical traffic data	Y	Y	Y	Y	Y	Y
F-3	Process road slope data	-	-	Y	N	Y	-
F-4	Process road topology data	Y	Y	Y	Y	Y	Y
F-5	Process speed limit data	Y	Y	Y	Y	-	Y
F-6	Process parking availability data	-	-	Y	Y	-	-
F-7	Process public transportation routes and schedule data	Y	-	Y	Y	-	-
F-8	Process non-motorized vehicle data	Y	-	Y	-	-	-
F-9	Close roads for a specific period of time	Y	-	Y	Y	-	Y
F-10	Close segments of lane for a specific period of time	-	-	Y	Y	-	Y
F-11	Choose from a list of possible events	-	-	-	-	-	Y
F-12	Change the available number of parking spaces	Y	-	Y	-	Y	-
F-13	Change the speed limitations	Y	-	Y	Y	-	Y
IS-1	Create new public transportation routes and schedules	Y	-	Y	Y	Y	N
IS-2	Change new public transportation routes and schedules	Y	-	Y	Y	Y	N
IS-3	Create new non-motorized only lanes and roads	-	N	-	-	-	N
IS-4	Evaluate low-emission zones and CO2 emissions	-	-	-	-	Y	-
IS-5	Simulate different types of vehicles	Y	Y	Y	Y	Y	Y
IS-6	Simulate public vehicles, routes and schedules	Y	Y	Y	Y	Y	-
IS-7	Simulate public service vehicles, usual routes and patterns	Y	Y	Y	Y	Y	-
IS-8	Simulate autonomous vehicles	Y	Y	Y	Y	Y	-
IS-9	Simulate pedestrian movement	Y	Y	Y	N	N	N
IS-10	Simulate non-motorized vehicles	Y	-	Y	Y	Y	N
IS-11	Simulate parking areas occupancy	Y	Y	Y	N	N	N
IS-12	Simulate unsignalized intersections	Y	Y	Y	Y	Y	Y
IS-13	Simulate signalized intersections	Y	Y	Y	Y	Y	Y
IS-14	Simulate roundabouts	Y	Y	Y	Y	Y	Y
IS-15	Simulate using batch mode	Y	Y	Y	Y	-	Y
IS-16	Simulate using microscopic models	Y	Y	Y	N	N	Y
IS-17	Obtain new traffic flow data	Y	Y	Y	Y	Y	Y

Table 5.5: Evaluation of the framework according to the requirements elicited. (Y) It has the capability of doing the requirement. (N) It was found proof that it is not possible to do that requirement. (-) There exists no information about the technology required to do that requirement

After analyzing Table 5.5 we can detect that each framework was designed to combat different problems existing in the traffic simulation world. Some requirements that stand out more, for their importance to the project, are the requirements (S1), (S14), and (F12). The requirement (S1) i.e., "be possible to create a roadmap based on OSM data", is not achieved by the frameworks PTV VISSIM, for its impossibility to create the roadmap, as well as the frameworks POLARIS and TraMod, for having no information in the documentation or a community help so that it could be recreated.

The requirement (S14), "Be possible to create new public transportation routes", was not achieved by TraMod for having no method of incorporating public transportation routes and schedules into the simulation. This requirement was also not possible to experiment with Aimsun as the full license was not received, and the free trial was limited, in documentation, functionalities, and in time.

The requirement (F12), "The simulation model must be microscopic, to track all individual vehicle movements", was not achieved by MATSim and POLARIS. This requirement is important because of its implication in the calculation of roads and may limit other requirements, such as the ability to close one or more lanes (S10).

After analyzing Tables 5.4 and 5.5, the best framework to continue the project and create the proof of concept is the SUMO framework. SUMO has a lot of functionalities that are needed for the project, good and updated documentation, as well as a large and active community.

In conclusion, it is believed that the traffic simulation framework which would satisfactorily correspond to the requirements required is the SUMO framework. As SUMO's programming language is Python, it is more suitable for the rest of the architecture, and the other components and frameworks used would be also in Python, for more seamless maintenance and better support.

Chapter 6

Planning and Methodology

A good outlining of a project is the key to its success. As such, it was created a plan for the project, with the objective of comprehending possible barriers and blocks, generating better time management, and a more satisfactory perception of the functionalities that can be experimented with and implemented in the final integrated system.

6.1 Process Management

During the course of the project, an Agile-based methodology was adopted. This methodology is based on a set of principles for software development that emphasizes flexibility, collaboration, and rapid delivery. Agile approaches are designed to help teams respond quickly to changing requirements or new opportunities. This methodology was chosen, so that can exist more collaboration with the advisors, such as with weekly meetings with Ubiwhere's advisors, or more global meetings with both advisors every 3 weeks. This kept the project under control and with enough feedback about its direction.

With the objective of documenting the time spent on the project, the time was recorded on Easy Redmine¹, a solution for project management used by Ubiwhere to evaluate project costs.

For the organization of the documentation of the project, a Google Drive folder was created, where minutes of the meetings and interviews are stored, along with diagrams that help explain important parts of the project and even presentation slideshows.

¹Easy Redmine: <https://www.easyredmine.com/>

6.2 Planning

The internship comprises two main phases organized in two semesters. This section presents the planning for both semesters, what tasks were expected to be finished, and the approximate week.

With the objective of planning the internship, a Gantt diagram was created. Although this diagram is most commonly used in Waterfall methodologies, it was decided that the development of a high-level Gantt diagram would be a good method to understand the outlines of the whole project, what would be delivered, as well as the time-bound of each task.

After each meeting with both advisors, a copy of the diagram would be altered, with the goal of having more knowledge about the duration of each task, and subsequently, the planning of future tasks recalculated.

The two Gantt diagrams were grouped into a single Gantt diagram, with the expected start and end, and new columns, with the actual duration of each activity and new estimations of the second-semester plans. Visualizing the Gantt, it is possible to see that some points took more time than originally expected, such as the *Choice of the Framework*. The delay occurred because the area of study, which is traffic flow, necessitated additional research efforts to gain familiarity with traffic flow observations, simulation techniques, and the selection of an appropriate simulation framework.

The point *Exploration of Framework* was then affected by the multiple tests carried out on all the simulation tools that were analyzed, with the objective of guaranteeing that the best framework would be selected to be used to develop the Integrated System.

Although new activities occurred in the creation of the thesis report, the duration of the activities remained approximately correct. New second-semester plans were also created after the conceptualization of the architecture, it was more uncomplicated to devise plans for the second portion of the project, mainly in the area of development.

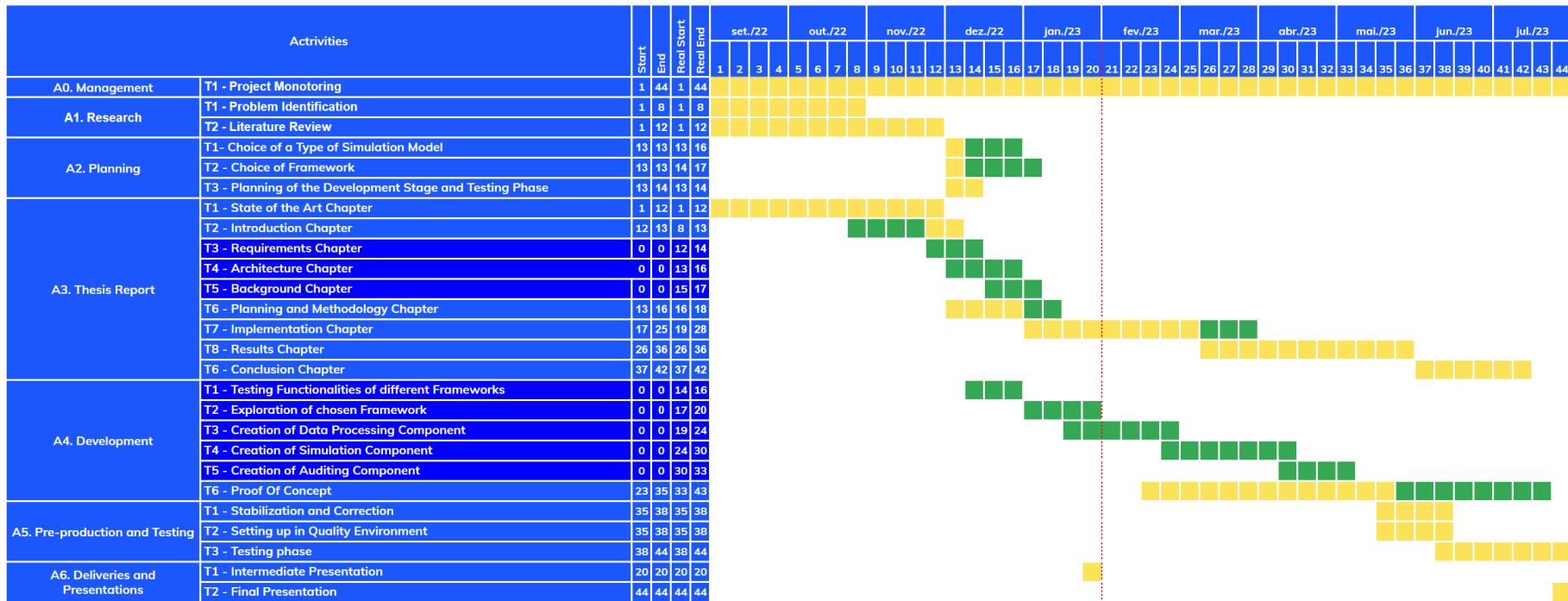


Figure 6.1: Gantt Diagrams of the thesis project, with the first conceptualized diagram, and updated at the end of the first semester. Light blue represents the activities planned at the start of the project. Darker blue shows the new activities added at the end of the first semester. The initial activity duration is represented in yellow. The modified activity duration is shown in green.

In the second semester, there was a bigger focus on the development of the integrated system, implementing the requirements gathered as well as the architecture presented in Chapter 4.

In the first weeks, the implementation of the Information Processing Component and the Simulation Component started, creating a way to organize the data according to the SUMO traffic simulation tool and reviewing the results. The second step was to adjust the data sent to the simulation, according to changes that a specific user wants to test. The next component to be operated with is the Auditing Component, allowing the user to save the traffic simulation data for future use.

6.3 Success Criteria

Success criteria refer to the specific goals or standards that must be met in order for a project, task, or activity to be considered successful. These criteria can include a wide range of factors, such as meeting certain performance targets, achieving certain milestones on time, or satisfying certain customer or stakeholder needs. Success criteria can be used to measure the performance of individuals, teams, or entire organizations, and can be used to determine whether a project or initiative is on track to meet its objectives.

In this project, some success criteria were established to ensure all guidelines are respected. The following criteria were:

- A minimal set of functionalities with “Must Have” priority regarding the Integrated System are implemented and tested
- The final product fulfills the quality attributes defined
- The implementation of the Integrated System is approved by Ubiwhere

6.4 Risk Assessment and Management

Risk assessment is the process of identifying, evaluating, and prioritizing potential risks in a specific situation, or during a particular activity or project. The goal of risk assessment is to identify potential risks and to determine the likelihood and severity of problems that may occur if the risks were to occur.

6.4.1 Risk Assessment

Using risk assessment methods a table was created with all the potential risks that may have occurred during the duration of the project. These risks were identified, as well as the consequences that would happen if those risks were to be

confirmed, and the mitigation plan that would be put in effect in order to prevent the risk and consequences from emerging.

Id	Risk	Consequence	Mitigation Plan
R-1	Poorly defined requirements	Development of unwanted or incorrect features. Non-fulfillment or delay of the objectives. Redefinition of the requirements	Intense planning and reviewing of the requirements. Redefining the project's objectives
R-2	The project is too ambitious	Non-fulfillment or delay of the objectives	Monitoring of the development. Redefining the project's objectives
R-3	Lack of experience using technologies necessary	Non-fulfillment or delay of the objectives.	Studying and experimenting with the chosen technologies before the start of the development
R-4	No traffic simulation tool has the ability to implement all the requirements	Non-fulfillment or delay of the objectives. Implementation of a new simulation tool or upgrade of an open-source framework.	Intense studying and experimentation of the simulation technologies before the start of the development.
R-5	The traffic simulation tool selected cannot implement the requirements needed	Non-fulfillment or delay of the objectives. Alteration of simulation tool chosen or upgrade of an open-source framework	Intense studying and experimentation of the simulation technologies before the start of the development
R-6	Lack of data to execute the simulations	Non-fulfillment or delay of the objectives	Redefining the project's objectives

Table 6.1: Risk assessment table of the project.

Each risk had a probability of happening as well as an overall impact on the project during the internship.

R-1 is a risk that could happen in every project, and if not addressed, can grow to large proportions, delaying immensely the final objectives of the project. In order to mitigate this risk, meetings were organized to focus on the requirements elicited and classified using methods such as SMART objectives, so that the requirements can be verified by Ubiwhere's members.

R-2 could happen if the scope of the project is too large or the duration of the tasks is incorrectly estimated and may cause the incompleteness of the Integrated System. This can have a relatively high chance of occurrence, as inexperience

with traffic simulation technologies may lead to an incorrect estimation of the duration or even the wrong classification of the requirements.

R-3 is a risk that was very likely to happen, not only with the traffic simulation platforms but also with platforms associated with the architecture, such as message brokers or asynchronous task queues. Although there exists some experience with some frameworks that can act as message brokers or task queues, further research was made into the technologies so the impact can be diminished.

R-4 was unlikely to happen. Multiple frameworks were created with the focus of simulation traffic flow and the focus of each is unique, making it unlikely that there would not exist a tool that can do the necessities of the project. To mitigate this risk, it will be analyzed a number of frameworks and the functionalities of each.

R-5 was somewhat unlikely to happen, for the same reasons as the risk R-4, but it's possible that the incorrect platform is chosen and the possibility of implementing a functionality is wrongly evaluated. To mitigate this risk, a thorough analysis of the simulation tool platform was performed.

R-6 was a very unlikely risk to happen, as the objective of this project is to use the data from Ubiwhere's Urban Platform, and as the project was completed under Ubiwhere's guidance, and with that associated knowledge of what data is possible to obtain. To mitigate this risk, meeting with the Ubiwhere advisor will happen to gain knowledge of what data is possible to use for the project.

6.4.2 Risk Matrix

A risk matrix is a tool used to evaluate and prioritize risk by combining information about the likelihood of the potential risk occurring with information about the potential severity of harm that may result if the risk does occur. This tool was used as it can help to visualize, prioritize and manage risk more effectively.

The risks mentioned above are presented by their ID in a risk matrix, where the horizontal axis represents the impact of each risk and the vertical axis means the likelihood of the said risk happening.

Both the likelihood and impact axis was divided into an even number of categories with the objective of not defaulting the value to the middle option, while also giving more choices of categorization of each risk.

The values of the vertical axis are divided into 4 different categories, each with its likelihood of occurrence, as can be seen in the following table:

The impact axis is also divided into 4 categories, each described in the following way:

Using these categories, each risk was evaluated and positioned in a risk matrix. Putting all of this together, we obtain the following risk matrix:

Category	Likelihood(%)
High	75 - 99
Medium High	50 - 74
Medium Low	25 - 49
Low	0 - 24

Table 6.2: Likelihood categories used in the risk matrix.

Category	Impact
High	The consequences are extremely hard to recover from
Medium High	The consequences are manageable but with significant effort
Medium Low	The consequences are relatively manageable
Low	The consequences are easily manageable

Table 6.3: Impact categories and description used in the risk matrix.

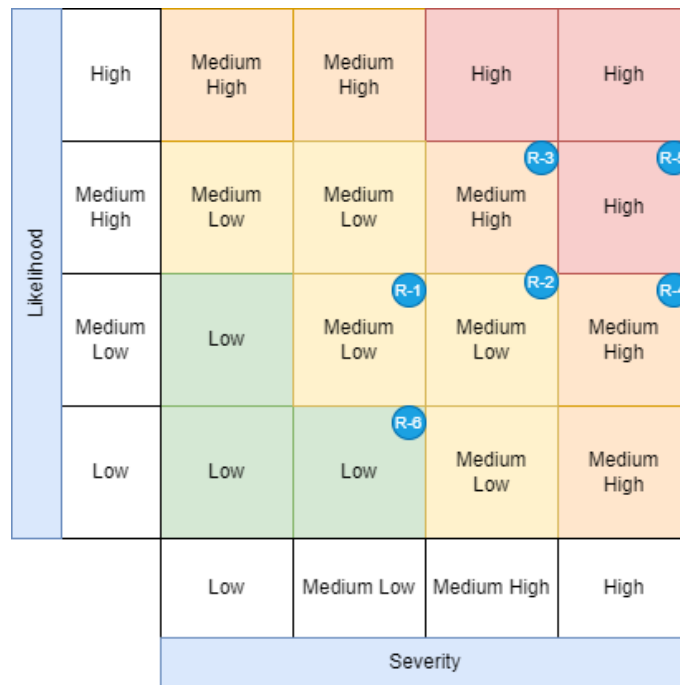


Figure 6.2: Risk matrix

Chapter 7

Implementation

7.1 API Prototype

One of the initial challenges faced in the project was the design of the SUMO API, which played a critical role by handling user information, processing data, and storing the results. The decision-making process involved carefully considering factors such as the choice of framework for the API, data aggregation methods, and the interaction between different models and web pages. It was crucial to thoroughly understand how each component of the SUMO framework operates and integrates with the rest.

To gain comprehensive knowledge, extensive research was conducted by studying the SUMO documentation and exploring important libraries like Sumolib and TraCI. Additionally, hands-on experience was gained by working with the code and creating a small proof of concept to test the core functionalities.

For the proof of concept, Flask, a user-friendly API framework, was utilized. This choice enabled experimentation with various tasks, such as network creation, generation of files containing random vehicle trips and routes, implementation of functions to simulate events within the network, conducting simulations, and analyzing the results. By employing Flask, different aspects of the project's functionality could be effectively explored and evaluated.

In the Flask API, several endpoints were established to serve different purposes:

- `"/networks"` was responsible for creating the road network.
- `"/trips"` facilitated the generation of vehicle trips.
- `"/configurations"` handled the preparation of the configuration file required for running the simulation.
- `"/reroutes"` allowed the obstruction of a full road to be created.
- `"/simulations"` encompassed the creation of all necessary simulation components and the execution of the simulation.

- `"/visualizations"` enabled the visualization of data results, organized by edges and lanes.

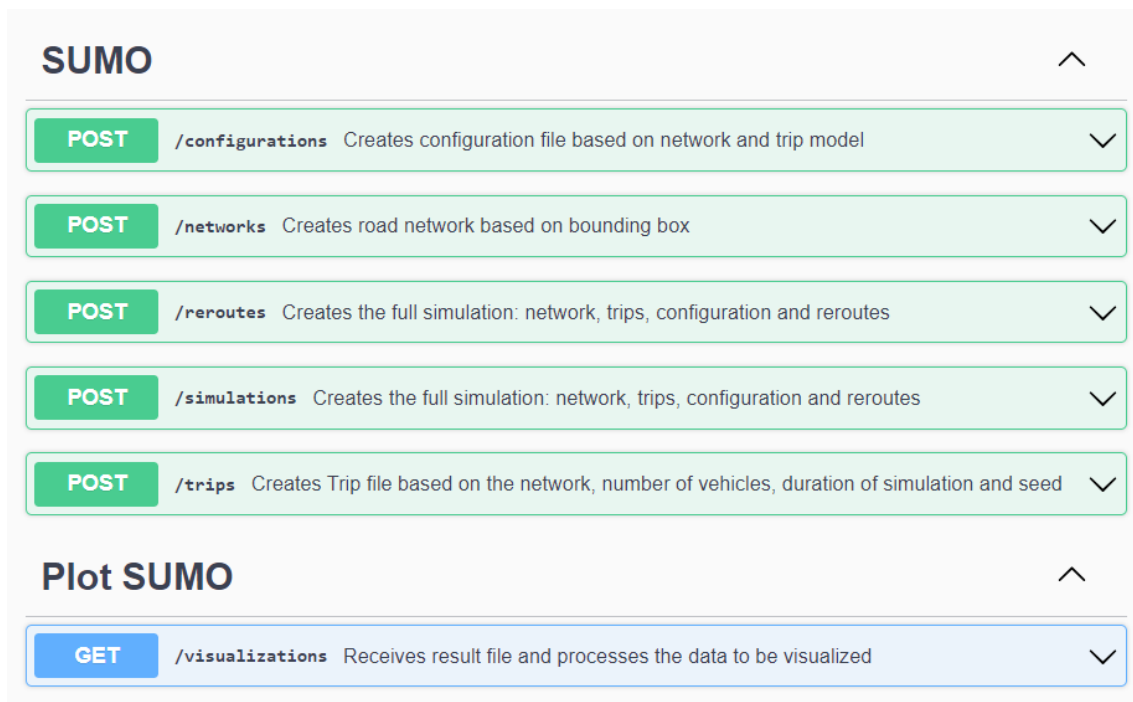


Figure 7.1

These endpoints played a crucial role in understanding SUMO design, what was needed for each component, and its limitations.

Given the complexity of the SUMO project and the requirements placed on the API, it became apparent that a more robust framework, featuring strong model relationships, would be essential. This realization led to the exploration of alternative options that could provide the desired capabilities and scalability. Ultimately, the decision to move away from Flask and consider frameworks similar to Django was driven by the necessity to accommodate the intricacies of SUMO and the heightened demands placed on the API.

By referring to the diagram ??, one can conduct a comprehensive analysis of significant inputs for the SUMO functions, understand the output generated by each function, and examine their interconnections.

The process of simulating SUMO can be categorized into three distinct phases: the establishment of the road network, the creation of vehicles for both private and public transport, and the generation of occurrences. Once all these elements are successfully created, the simulation can be configured and executed, offering a wide range of result options. These results can include general data or be grouped according to vehicle or road, providing valuable insights into various aspects of the simulation.

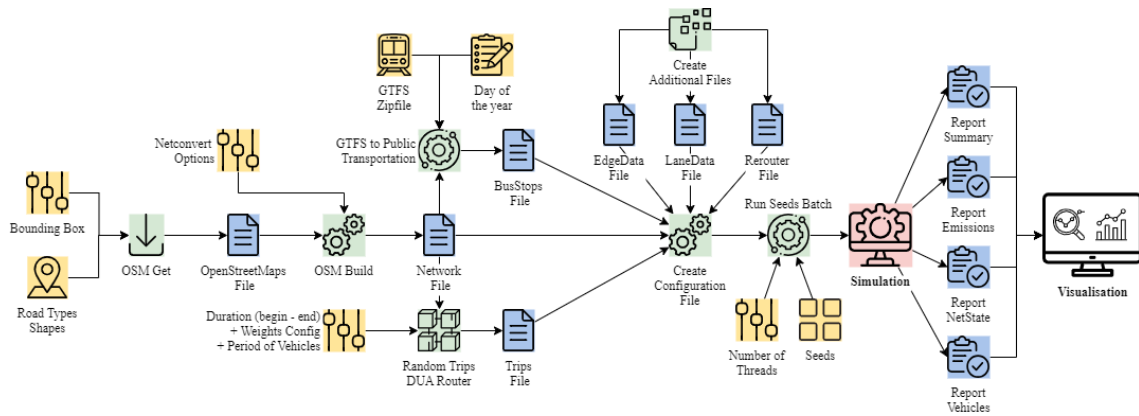


Figure 7.2: Diagram depicting the intricate connections and interactions between various functions within SUMO. This diagram played a crucial role in facilitating a deep understanding of how the API should be effectively implemented.

7.2 API Implementation

The decision to develop the SUMO API in Django was a straightforward one due to several factors. Firstly, Django offers a built-in Object-Relational Mapping (ORM) system, which simplifies the management of model relationships. This feature was particularly important for the SUMO API, as complex relationships between various models needed to be established. Django's ORM provides convenient field types like `ForeignKey`, `OneToOneField`, and `ManyToManyField`, allowing seamless representation of these relationships within the database. The entity-relation diagram that was created using Django ORM is referenced later in section 7.2.1.

Additionally, Django provides an integrated admin interface, automatically generated based on the defined models. This feature greatly simplifies the management and manipulation of related data through a user-friendly interface. The ability to add, edit, and delete related objects without writing custom code significantly streamlines data administration tasks.

Another contributing factor to choosing Django was the existence of libraries that enhance its capabilities as a REST framework for Web APIs. These libraries provide additional functionality and facilitate the development of RESTful APIs, aligning well with the requirements of the SUMO API.

Furthermore, prior knowledge and experience with Django played a role in the decision-making process. Familiarity with Django's conventions, syntax, and best practices allowed for a smoother transition and accelerated development. This familiarity also fostered synergy within the company, as Django was commonly used within Ubiwhere. This ensured better integration of the SUMO API with other ongoing projects within the organization.

7.2.1 Data Models

The organization of the database is crucial to ensure efficient handling of large data files, typically around 1 GB in size. Therefore, it is essential to have a well-implemented and structured relationship between the models. The following models have been developed to address these needs:

- **Network:** This model stores information about bounding boxes, and tags, and creates the roadmap for a specific zone.
- **Type:** The Type model facilitates the creation of different vehicle types, including regular, electrical, autonomous, or even buses.
- **Trip:** This model enables the creation of multiple vehicles within a specific period.
- **GTFS:** The GTFS model is designed to store the GTFS file pertaining to a specific zone. This file follows the Google standard and aims to encompass comprehensive information related to public transportation.
- **Public Transport:** This model translates the GTFS file into SUMO format, according to a specified day of the year.
- **Road Block:** The Road Block model stores information about blocked lanes and edges for a specific duration.
- **Speed Restriction:** This model stores details about lanes and edges with modified maximum speed limits for a specific duration.
- **Partial Road Block:** The Partial Block model stores information about incidents affecting specific parts of a lane, blocking all vehicles from traveling
- **Road Access Restriction:** This model stores information about restricted access to certain lanes and edges. This model also allows recreating of autonomous vehicles following European rules. It also includes the period and step of creation for autonomous cars to travel on those lanes.
- **Simulation:** This model allows the selection of other models and specifies the files to be created for simulation purposes.
- **Result:** The Result model executes the SUMO simulation and stores the resulting files.

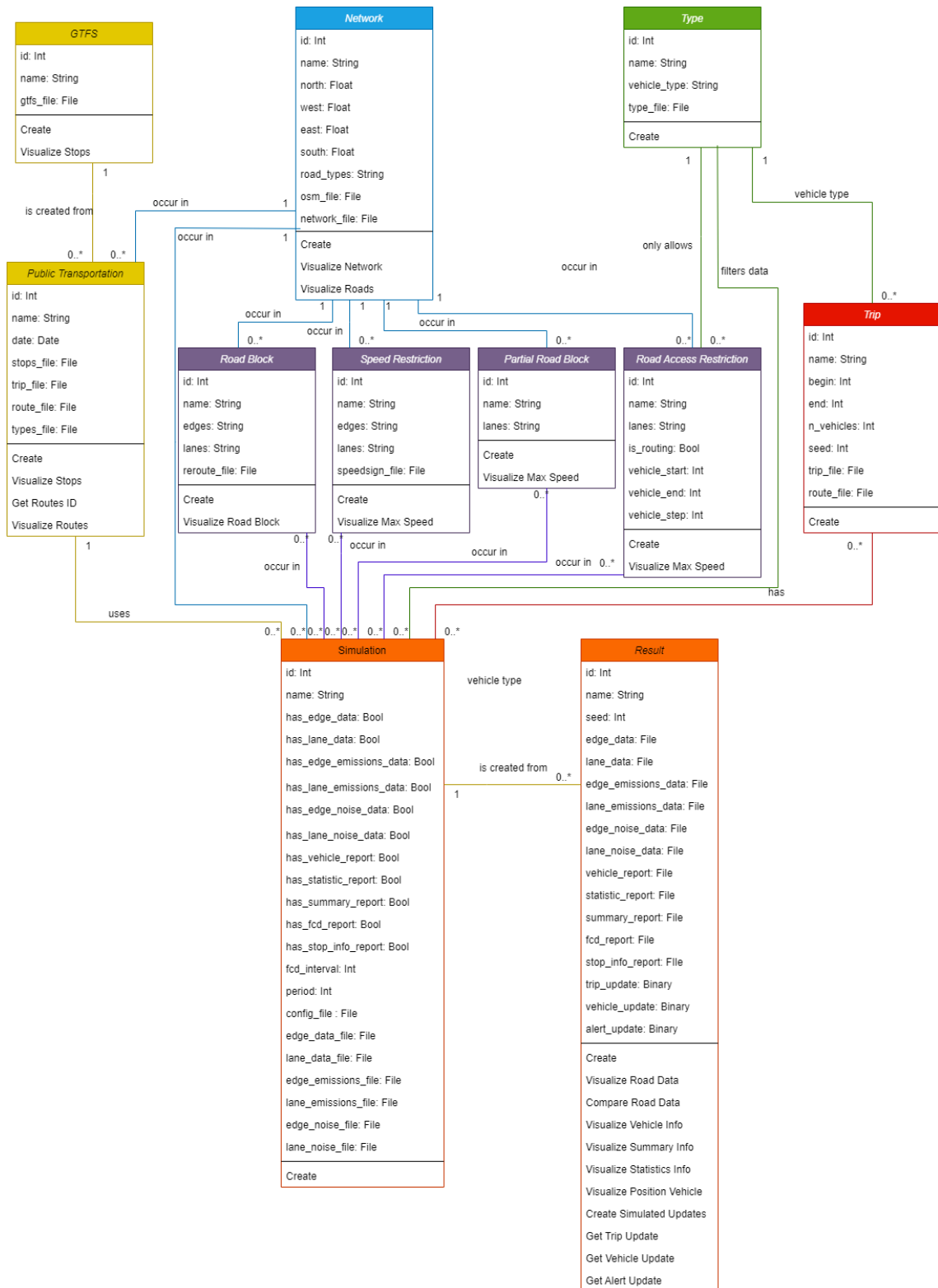


Figure 7.3: Diagram of entity-relationship between resources, and their parameters.

Every "occurrences" model, including Road Block, Speed Restriction, Partial Road Block and Road Access Restriction incorporates a visualization function that enables users to view the specific lanes and edges where the occurrences will take

place. This feature allows users to assess whether the selected model aligns with their intended modifications or if they need to create a new model with different road selections.

Occurrences and Roadblocks

One of the project's objectives was to incorporate various occurrences and events into the simulation. SUMO provides several built-in capabilities for creating such events, including road blocking, speed limitations in specific areas, and restricted access for certain vehicles.

However, there are common occurrences in everyday life, such as temporary road blockages due to vehicle accidents, minor repairs, or mechanical issues with buses, which do not align with SUMO's predefined event options. These events involve only a tiny portion of a specific lane being blocked. To address this, a new functionality was developed using TraCI, enabling real-time road traffic simulation during the simulation process.

Since SUMO does not directly support the partial blocking of a road segment, an alternative approach was conceived. The concept involved creating a "ghost" vehicle of variable size at a specific location on the road and setting its speed to a static value until a user-specified time. To ensure that the presence of the "ghost" vehicle did not significantly impact important simulation results, such as increasing traffic density or drastically reducing average road speeds, a distinct vehicle type was introduced specifically for "ghost" vehicles. These vehicles were then excluded from regular data calculations.

Another roadblock that was discovered in the project was in the creation of autonomous vehicles. The initial implementation of autonomous vehicles, and still functional, was to adapt the functionality of creating random trips and changing the type and model of the vehicles, allowing to use more accurate models for the implementation of autonomous vehicles. The roadblock was that in Portugal/Europe the regulations limit the circulation of autonomous vehicles to be in a segregated route, from people and/or other vehicles. As so, some functionalities were added to the model Road Access Restriction, that allows the creation of vehicles in the specific route given.

Vehicle Creation

One objective of this API was to facilitate the creation of vehicles in a more natural manner, with the aim of achieving more accurate results. Extensive research was conducted on the Binomial distribution and how the trip creation function could be modified to incorporate a different distribution beyond the default uniform distribution. However, it proved challenging to accomplish this goal due to limited documentation on the function. Understanding the precise location within the code where changes should be made without compromising the overall functionality proved to be quite complex.

To enhance the vehicle distribution without modifying the SUMO code directly, a decision was made to allow a Simulation model to reference multiple Trip models. This approach could potentially enable the creation of a more diverse distribution, incorporating different vehicle types such as electric and gas vehicles. By introducing such a variety, the simulation becomes more realistic and reflects the mix of vehicles commonly found on the roads.

Connected Cooperative and Automated Mobility

There are notable distinctions between human driving and autonomous vehicles, encompassing factors such as decision-making, reaction time, consistency, adaptability, and limitations. Human drivers base their decisions on their perception of the environment, skill level, reaction time, and inherent limitations. In contrast, autonomous vehicles utilize data from cameras and sensors to interpret their surroundings and make decisions, offering consistent and standardized driving behaviors.

Within SUMO, various models have been implemented, although they were not specifically designed for autonomous driving. Models such as the Intelligent Driver Model and the ACC (Adaptive Cruise Control) car-following model can serve as starting points for simulating autonomous driving behavior. These models can capture aspects such as faster reaction times and better control of the distance between vehicles. However, they may not fully replicate the advanced decision-making and control strategies employed by autonomous vehicles.

Another model implemented in SUMO is the CACC (Cooperative Adaptive Cruise Control) car-following model. This model extends the functionality of the ACC system by enabling vehicles to communicate cooperatively with each other, maintaining desired vehicle distances and speeds. While the CACC model offers a simplified representation of real-world behavior and may not perfectly replicate the advanced CACC systems deployed in reality, it provides valuable insights into the behavior of autonomous vehicles.

During the implementation of autonomous vehicles in the project, a roadblock was encountered. The initial approach involved adapting the functionality for generating random trips and modifying the vehicle type and model to incorporate CACC capabilities. This adaptation aimed to utilize more accurate models for implementing autonomous vehicles. However, regulations in Portugal/Europe restrict the circulation of autonomous vehicles to segregated routes, separate from other vehicles. Consequently, additional functionalities were added to the Road Access Restriction model, allowing the creation of vehicles within specific designated routes to comply with the regulations.

7.2.2 Public Transportation Schedules

To fulfill the project requirements, it was necessary to establish public transportation networks, including routes and schedules. To achieve this in SUMO, a widely

accepted standard called GTFS was utilized, making the recreation process more convenient

GTFS, or General Transit Feed Specifications, is a widely used format for organizing public transportation schedules and geographic information. It enables public transit agencies to publish their transit data, allowing developers to utilize this information for developing applications.

The GTFS format is composed of a large number of files. The following files are required for the use of GTFS:

- `agency.txt`: Transit agencies with service represented in this dataset.
- `stops.txt`: Stops where vehicles pick up or drop off riders. Also defines stations and station entrances.
- `routes.txt`: Transit routes. A route is a group of trips that are displayed to riders as a single service.
- `trips.txt`: Trips for each route. A trip is a sequence of two or more stops that occur during a specific period.
- `stop_times.txt`: Times that a vehicle arrives at and departs from stops for each trip.
- `calendar.txt`: Service dates specified using a weekly schedule with start and end dates. This file is required unless all dates of service are defined in `calendar_dates.txt`.
- `calendar_dates.txt`: Exceptions for the services defined in the `calendar.txt`. If `calendar.txt` is omitted, then `calendar_dates.txt` is required and must contain all dates of service.

An example of a `stop_time` feed:

<code>trip_id</code>	<code>arr_time</code>	<code>dep_time</code>	<code>stop_id</code>	<code>stop_seq</code>	<code>stop_headsign</code>
106_1_A_1	7:00:00	7:00:00	FRC	1	-
106_1_A_1	7:00:57	7:00:57	FRCL2	2	-
106_1_A_1	7:01:46	7:01:46	ISG2	3	-
106_1_A_1	7:02:19	7:02:19	ACBL2	4	-
106_1_A_1	7:03:02	7:03:02	VGM2	5	-

Table 7.1: Table of 5 instances from GTFS file `stop_times.txt` from Porto City.

This example was extracted from Porto's GTFS open data file. These instances illustrate the first five stops of vehicle "106_1_A_1" when it arrives and departs.

The following image shows in more detail the GTFS data model, and its interactions.

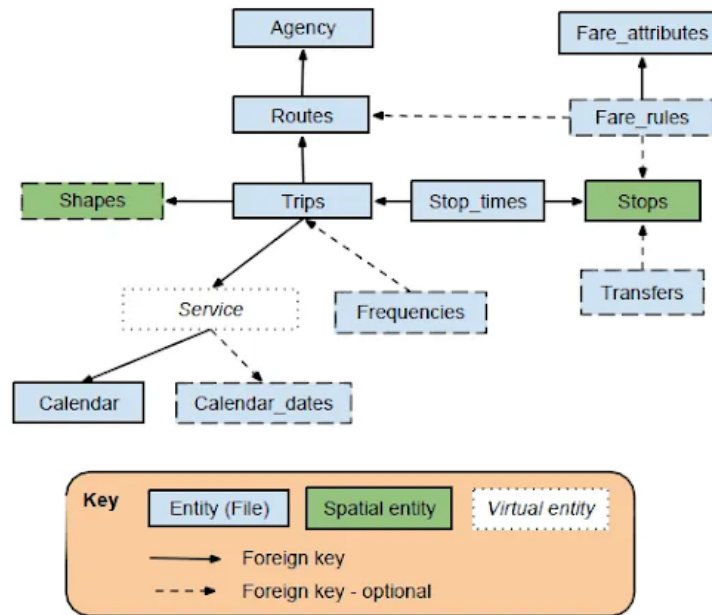


Figure 7.4: Diagram of interactions between the various GTFS files, by [42]

SUMO allows the integration of GTFS in the simulation, translating all of these files into readable files by the SUMO simulation, separating the information into 2 different files: Bus Stops - with position, road, and identification Vehicles - with information about which route the vehicle would take. This route would have specified the edges traveled, the bus stops, and how much time to wait at a stop To guarantee the translation of the GTFS to public transport in SUMO, a couple of visualizations were done, in both models to compare where the bus stops were positioned and what bus stops and roads each bus route would represent.

7.2.3 Simulation Results

The API was designed with the objective of processing results to facilitate information visualization in third-party applications like Urban Platform. To achieve this, the API incorporates a crucial model called the results model. This model is linked to the simulation model, which stores all variables and relationships. The results model not only creates the simulation but also stores all the data associated with that particular simulation. The data is stored in various formats to extract important information effectively:

- Data is grouped by road, including details such as density, pollutant quantities, noise levels, etc.
- Data is grouped by vehicle, capturing departure and arrival times, number of stops, emission quantities, etc.
- Data is summarized based on different steps, providing information like the number of loaded cars per step or the duration of each step's execution.

- Statistics are collected, presenting overall figures such as the total number of loaded vehicles, routing errors, average speed, and more.
- Information regarding intentional bus stops is recorded, including the time, bus stop location, delay, and the number of passengers loaded and unloaded.
- Detailed vehicle data is available for each step, including vehicle positions, speed, and lane position.

These various methods of grouping data enable the extraction of a wide range of information about any specific simulation.

7.2.4 Simulated Public Transportation Live Updates

Google also designed a feed called GTFS Real Time, which complements GTFS. This feed offers users real-time updates on transit data, providing current details like arrival and departure times, route delays, and other relevant information. GTFS Real-Time encompasses the following types of data:

- Trip updates: Comprises details regarding delays, cancellations, and modifications to the route.
- Service alerts: Informs users about station relocations, and unforeseen events impacting a station, route, or the entire network.
- Vehicle positions: Provides data on vehicle locations and congestion levels.

To deliver this information to users, the transportation fleet requires sensors that update multiple times throughout the day. However, equipping every bus with GPS trackers for public transportation can be expensive. As a solution, an idea emerged to utilize the SUMO API, designed to facilitate the implementation of this system in smaller cities lacking the resources for individual bus tracking. The proposed approach involves using the simulation results, such as vehicle positions and the stops of each public transport vehicle created by GTFS, to generate a file structured similarly to GTFS Real Time but with simulated information from SUMO.

However, the process of creating these files turned out to be more complex than initially anticipated. The documentation necessary for generating these files was found to be nonexistent, which posed a challenge faced by many other developers. In order to address this issue, information was gathered from the GTFS Real-Time proto buff files implemented by various entities, particularly BigBlueBus, a bus company in Los Angeles. Examples of components provided by Google Transit were also examined ¹. To enhance comprehension of the hierarchical structure of GTFS Real Time, a diagram was created.

¹gtfs real-time documentation: <https://developers.google.com/transit/gtfs-realtime/examples/vehicle-positions>

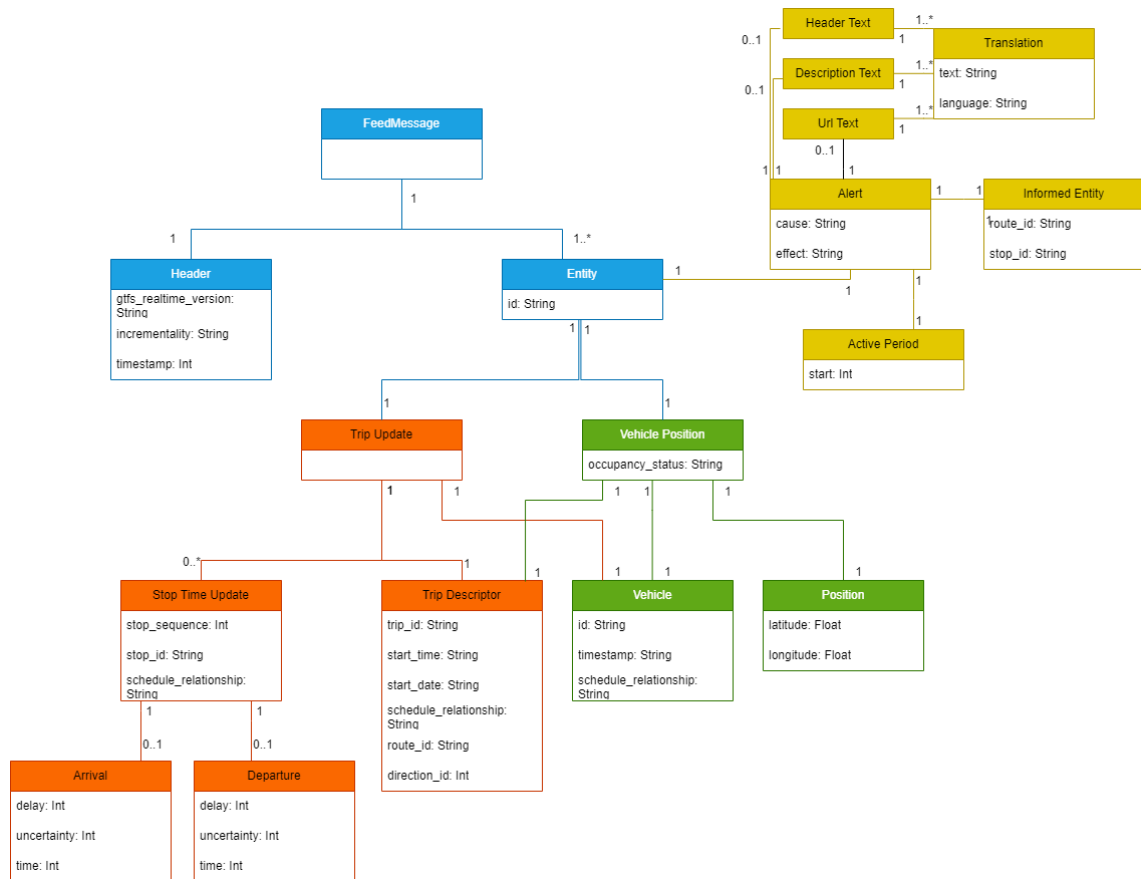


Figure 7.5: Diagram of GTFS RT feed messages.

In order to create the protobuf files, an algorithm was implemented which used information about each specific bus from GTFS, and at the same time, read information from different simulation files, such as vehicle position and bus stop information. Once the protobuf files are generated, they are stored in the Result Model as binary data. This allows the files to be saved and downloaded, enabling them to be utilized like any other GTFS Realtime (GTFS RT) feed.

7.3 Web Interface

While developing the API for integration with SUMO, a web interface was created to simplify the visualization of results. Streamlit is an open-source Python framework that allows you to create interactive web applications for machine learning, data visualization, and other data-related tasks. With Streamlit, one can quickly build and deploy applications without having to write extensive HTML, CSS, or JavaScript code. It simplifies the process of creating interactive and visually appealing interfaces, making it accessible to both experienced developers and data scientists without extensive web development expertise.

This framework was chosen so that the project could focus on the API, its functionalities and integration with SUMO.

Using this framework, a number of webpages were created during the development of the project, so that the various functionalities could be presented in the biweekly reunions with the thesis advisors.

7.3.1 OpenStreetMap Exporter

One of the early web pages developed using Streamlit was an OpenStreetMap Exporter. Its primary aim was to provide a straightforward solution for obtaining the bounding box, a crucial component in creating city roadmaps for our implementation. This webpage was also created with the possibility of being used for other projects inside Ubiwhere, where a specific bounding box is required.

At the core of this web page, there is an interactive map featuring a green translucent box. This box's position can be adjusted using three key components:

- **Location Text Input:** This input field allows for significant alterations to the bounding box coordinates, thanks to integrated geolocator technology. By entering specific locations or addresses, users can retrieve corresponding latitude and longitude values from the Nominatim Ubiwhere service. These values are then utilized to dynamically update the position of the box.
- **NorthWest and SouthEast Inputs:** These inputs enable users to modify the bounding box by specifying two distinct points typically employed in defining its boundaries.
- **Individual Integer Inputs:** To facilitate fine adjustments, the application includes four individual integer inputs. These inputs allow users to make small changes to each coordinate point, leveraging the user-friendly interface of Streamlit's integer input, which includes buttons for convenient value adjustments based on defined increments.

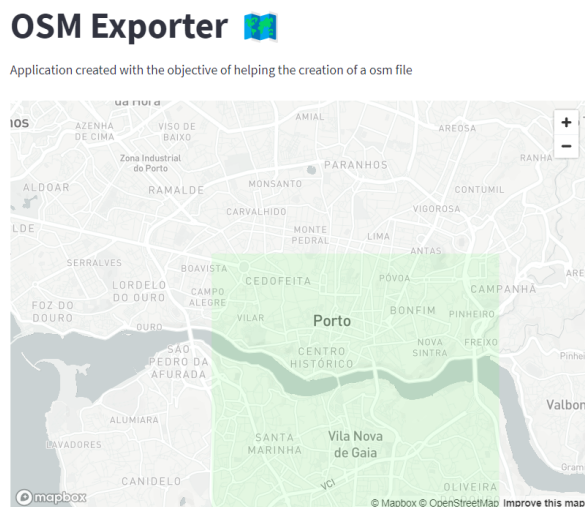


Figure 7.6: Display of a section from the OSM Exporter webpage.

The second part of the webpage revolves around selecting the different road tags used by OpenStreetMap to describe roads. In order to simplify this selection process, the tags have been categorized into the five most commonly used keys in OSM: highways, airways, waterways, aerial ways, and routes.

The third and final section of the webpage showcases three export buttons. The first button facilitates the export of the GeoJSON file. This button was designed to provide a straightforward means of analyzing the exported data to ensure its accuracy and usability.

The remaining two buttons are responsible for exporting the XML files generated by SUMO calls. The first button generates a file that contains all the information about the OSM within the specified bounding box, but in XML format. However, this file alone is not directly usable. The second file contains processed information about the roadmap, which can be read by SUMO. It is important to note that the second file is only created once the first one has been generated.

7.3.2 Traffic Simulation Proof of Concept

This webpage was created with the objective of using and showing the proof of concept that was implemented with Flask. Two main functionalities were created in this testing webpage:

1. **Simulation Creation:** This feature enables the user to create simulations by defining the roadmap layout, placing vehicles within the simulation, and testing various simulation occurrences. Users can experiment with different scenarios, such as blocking specific edges of the road network, adjusting maximum speeds, and restricting access of certain vehicle types to particular roads.
2. **Simulation Visualization:** The webpage offers a map visualization component that allows users to view and analyze road data. Users can explore the road network individually or compare it with data from other simulations. This visualization capability provides insights into traffic patterns, road conditions, and other relevant information.

As referred to previously, this proof of concept allowed to understand the depth of the SUMO framework, and how it works, from the creation of the mapping of the streets and vehicles, even the creation of occurrences, such as blocking edges, changing the maximum speed, and even limiting access of vehicle types to certain roads. Some functionality was also tested within this iteration of the project, mainly translating General Transit Feed Specification (GTFS) to SUMO files, and understanding its limitations and interactions with the other vehicles within the simulation environment.

Traffic Simulation

Application created with the objective of helping the creation of a traffic simulation with SUMO

Create network file in OSM Exporter

Network Parameters

Bounding Box

```
{ "west": -8.44300134452906, "south": 40.17648, "east": -8.383001344529061, "north": 40.23648 }
```

Tags

```
{"highway": ["primary", "secondary", "tertiary"]}
```

Prefix of files

```
osm
```

Trip Parameters

Seed Value of Trip Creation	Number of Vehicles
42 <input type="text"/> - <input type="button" value="-"/> <input type="button" value="+"/>	1000 <input type="text"/> - <input type="button" value="-"/> <input type="button" value="+"/>
Start of Simulation	End of Simulation
0 <input type="text"/> - <input type="button" value="-"/> <input type="button" value="+"/>	3600 <input type="text"/> - <input type="button" value="-"/> <input type="button" value="+"/>

Figure 7.7: Display of a section from the Traffic Simulation Proof of Concept webpage.

7.3.3 Model Creation

Following the transition of the API framework from Flask to Django, a higher level of organization became essential not only within the Django API but also within the webpage itself. The primary objective of this webpage implementation was to facilitate the creation of all available models in the SUMO API, encompassing the ability to modify various parameters associated with each model.

Model Creation

Application created with the objective of assisting the creation of models in SUMO API

Create Network

The Network model is the foundation of the SUMO simulation, being crucial for the creation of many other models.
It is responsible for storing the data of the roads and edges that exist between the bounding box.
This model is also the main component in all the map visualizations.

Name

```
Porto Network
```

Bounding Box

```
{"north": 41.198489100000046, "west": -8.703727623541692, "south": 41.0534891, "east": -8.523727623541692 }
```

Tags

```
{"highway": ["motorway", "motorway_link", "primary", "secondary", "tertiary"]}
```

Figure 7.8: Display of a section from the Model Creation webpage.

Functioning as a central hub, the webpage establishes connections with all the models offered by the API, focusing particularly on enabling the creation of new models. Furthermore, the “occurrences” Models feature a special function that al-

allows users to preview the modified roads. This preview functionality offers users a glimpse of how the altered roads will appear within the simulation, enhancing their ability to assess and visualize the direct implications of the new model.

7.3.4 Model Validation

One of the primary considerations when using the previous webpage to create models is the potential risk of unintended outcomes or discrepancies between the user's intentions and the actual implementation. This concern becomes particularly pronounced in models that directly modify the road network, known as "occurrences" models. To address this, a straightforward model selection and map visualization feature was implemented to provide users with a clearer understanding of the specific changes being made in the simulation.

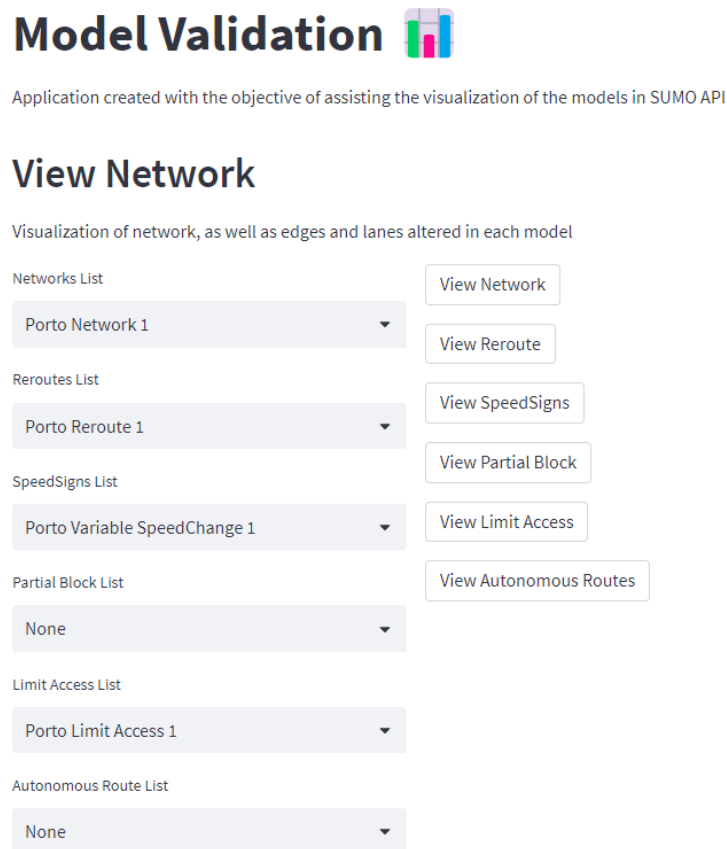


Figure 7.9: Visualization of a specific segment on the Model Validation webpage, showcasing the various options for road visualization. This includes the ability to view the entire network as well as specific subsets of roads affected by specific incidents or occurrences.

Another aspect of model creation that posed a potential concern was the accuracy and fidelity of the translation process from GTFS files to SUMO files. It was crucial to avoid any problems, positional errors, or the creation of inaccurate bus routes. To mitigate these risks, a visualization tool for GTFS stops was developed, along with a comprehensive representation of stops and bus routes

extracted from the created SUMO files. These visualizations aided in identifying any potential issues and ensuring the accuracy of the translation process.

7.3.5 Result Analysis

The simulation results yield a diverse range of data, encompassing information grouped by roads, vehicles, simulation timesteps, and overall statistics. To effectively visualize this information, multiple components were developed and integrated into the webpage.

The screenshot shows a web interface titled "View Road Data" with the subtitle "Visualization of road data after simulation." The interface is organized into several sections:

- Results:** A dropdown menu set to "Porto Result 1".
- Comparison Results:** A dropdown menu set to "Porto Result 1".
- Time Period:** A numeric input field set to "5" with minus and plus buttons.
- Comparison Time Period:** A numeric input field set to "0" with minus and plus buttons.
- Parameters:** A dropdown menu set to "occupancy".
- Color Max Value:** A numeric input field set to "-1" with minus and plus buttons.
- Color Min Value:** A numeric input field set to "-1" with minus and plus buttons.
- Color Map:** A dropdown menu set to "Green-Yellow-Red".
- default_color:** A text input field containing "[0, 0, 0, 255]".
- Checkboxes:** Three checkboxes are present: "Log Colors" (unchecked), "Include Junctions" (unchecked), and "Edge Data" (checked).

Figure 7.10: Visualization of a specific segment on the Result Analysis webpage, showing parameters for the analysis of road data.

The first component, inspired by the SUMO function “plot_net_dump.py”, focused on visualizing road-specific data. It was enhanced to facilitate not only the comparison of different simulation results but also the comparison of values across different time periods within the same simulation. This allowed for a more comprehensive understanding of the significance of the data, including parameters such as density, average speed, emissions, and noise pollution.

Another vital visualization feature involved the ability to select any two parameters from the vehicle-specific information and plot them. This capability facilitated the analysis of potential correlations between various values. For example, it enabled the examination of whether vehicles spending more time in the simulation were due to traffic congestion or to assess if vehicles experiencing higher

levels of traffic were associated with a greater emission footprint.

Additionally, a summary plot component was created to display information primarily in the form of time series plots. Users were granted the flexibility to choose the parameters for the plot, allowing for customized visualizations.

Lastly, a straightforward visualization component was developed to present statistics parameters. This feature enabled users to visualize information pertaining to performance metrics (e.g., simulation duration, time periods) as well as vehicle-specific details (e.g., count, speed, total route length), among other relevant parameters.

7.3.6 Vehicle Position

A subsequent visualization feature that was implemented allowed the user to view the positions of all vehicles within the simulation at any given moment. However, creating this visualization posed challenges due to the large file size required to store the positions of each vehicle throughout the entire simulation duration. In order to address this issue, modifications were made to the simulation, described below.

Instead of capturing the position of each vehicle every second, the time step was adjusted to generate a lighter file, thus facilitating the creation of the visualization. By altering the time step, it reduced the file size while still providing meaningful information for visualization purposes.

View Position of Vehicles

Visualization of position of vehicles in a specified time period.

Results

Porto Result 1

Time

54000

View Position of Vehicles

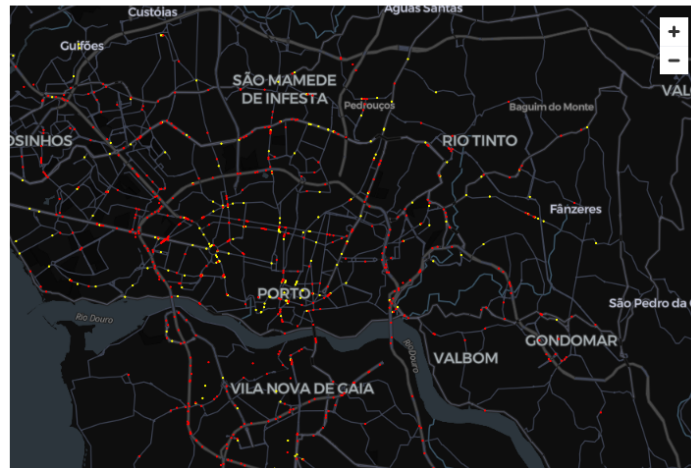


Figure 7.11: Visualization of the Vehicle Position webpage.

To enhance the vehicle position visualization, an additional component was developed that provided information specifically related to the selected time step. This component aimed to provide users with a better understanding of the ongoing events in the simulation, including the number of vehicles currently in operation and the number of vehicles that have already completed their trips. This supplementary information further enriched the user's comprehension of the simulation dynamics.

7.3.7 Simulated Public Transportation Updates

During the implementation of the simulated public transportation updates, various new components were developed to enhance the understanding of GTFS Real-Time functionality. These components aimed to explore how different organizations utilize message feeds, aiming to achieve a comprehensive understanding of the Google standard's usage.

The first component created was a Feed Message reader, which receives a URL and extracts the feed message to be displayed to the user. Initially, this component was designed to read Open-Source GTFS Real-Time data from other cities. However, after the implementation of the simulated updates, it now serves as both a proof of concept display and a means to read and showcase the generated updates.

Simulated Public Transportation Updates

Web page that shows GTFS Real-Time and near real time simulated in the format of FeedMessages, created by Google

Visualization of GTFS Real-Time

Trip Update

Vehicle Update

Alerts Update

Number Of Entities

 - +

Figure 7.12: Visualization of the first component of Simulated Public Transportation Updates webpage.

The second component on the webpage establishes a simple connection to the endpoint responsible for generating the simulated public transportation updates. This component utilizes the results obtained from a specific simulation and the corresponding GTFS data used in creating the simulation. These results serve to update the model, enabling the display of trip updates and vehicle positions in the first component.

7.3.8 Simulation Scenarios

To show all the functionalities that were created, a scenarios page was implemented. This page serves as the primary webpage, designed to provide a simplified and straightforward presentation of the various scenarios tested within the scope of this project.

The implementation of these scenarios took place in Porto, a prominent city in Portugal, with a partnership with Ubiwhere across multiple projects and its extensive collection of available datasets. Porto's significant advantages include a comprehensive roadmap and a high volume of daily vehicles, enabling the generation of dynamic results and rigorous testing of the API with large datasets and intensive simulations.

The following scenarios were implemented:

- **Baseline Scenario:** This scenario serves as the default scenario, acting as the foundation upon which all other scenarios were built. It includes the road network of Porto City, incorporates public transportation data from STPC, and simulates the presence of 500,000 private vehicles.

- **Occurrences Scenario:** Similar to the baseline scenario in terms of configuration, this scenario introduces specific incidents occurring in Ponte Arrábida. These incidents aim to significantly disrupt the daily lives of citizens, highlighting the potential impact of unexpected events.
- **Road Restriction Scenario:** In this scenario, access to certain roads is restricted exclusively to private vehicles, promoting the usage of public transportation as the primary mode of transportation. This scenario emphasizes the importance of encouraging sustainable and efficient transportation options.
- **Reserved Route for CCAM Scenario:** This scenario focuses on designating a scenic area exclusively for autonomous vehicles, restricting access to other types of vehicles. By showcasing this reserved route, the scenario demonstrates the potential of autonomous vehicle integration in enhancing specific areas or routes.

These scenarios effectively illustrate the diverse applications of the SUMO API in managing a large city like Porto. They offer visualizations of various information types and serve as practical demonstrations of the capabilities and potential benefits of leveraging the SUMO API in urban management.

In order to facilitate a comprehensive analysis of the KPIs that were mentioned in section 3.3, a dedicated component was developed. This component introduces a range of features that allow users to view important averages derived from the simulation data.

Among the key performance indicators provided are the average speed and average speed per vehicle, providing insights into the overall velocity maintained throughout the simulation. This metric helps assess the efficiency and smoothness of vehicle movements within the simulated environment.

Additionally, the component enables the observation of average emissions generated during the simulation. This information proves vital in evaluating the environmental impact of the simulated scenarios, aiding in the assessment of sustainability and air quality considerations.

Furthermore, the component provides valuable statistics on stop time, which refers to the period during which vehicles have scheduled stops, and waiting time, which represents the period during which vehicles make unscheduled stops. This data sheds light on irregularities or delays encountered during the simulation, offering valuable information for optimizing routes, reducing congestion, and improving overall efficiency.

passenger	17.3212
bus	6.3104
speed	16.6824
km_traveled	10,455.0373
stop_time	35.1593
wait_time	81.6719
wait_count	4.0787
co2	4,483,903.2652
pm	376.5103
nox	14,590.5517

Figure 7.13: Example of the Key Performance Indicator display component.

By incorporating these diverse averages into its functionality, the component offers users a simplified and accessible view of the essential KPIs mentioned in section 3.3. This user-friendly interface facilitates comprehensive analysis, aiding in decision-making processes and providing valuable insights for enhancing the performance and sustainability of the simulated system.

7.4 Simulation Efficiency

In order to ensure that the aforementioned scenarios can be calculated within a reasonable time frame for prospective clients, it is necessary to expedite the simulation process or enable asynchronous execution. This enables users to request simulations and carry on with their other tasks without having to wait for the results.

To address this need, an internal server has been set up to handle the execution of these scenarios. As part of a proof of concept, certain frameworks, namely Redis and Celery, have been incorporated into the project. These frameworks provide essential functionality for efficient task management and distributed computing.

Redis is a high-performance in-memory data structure store that serves as a key-value database, caching system, and message broker. By utilizing Redis, the project gains the ability to store and retrieve simulation data quickly, enabling faster computation and response times.

Celery, on the other hand, is a distributed task queue system that allows for the asynchronous execution of tasks. With Celery, simulations can be dispatched to worker processes, which operate independently from the main application. This enables concurrent processing and ensures that users can request simulations without experiencing delays, as the tasks can be distributed across multiple workers.

By combining the benefits of Redis and Celery, the project aims to streamline the computation of scenarios, delivering faster results and an improved user experience. The addition of these frameworks serves as a proof of concept, demonstrating the feasibility and potential for enhanced performance in handling simulation

requests.

To illustrate the concept, a fundamental worker function was created with the purpose of generating simulated updates for Google’s standard trips and vehicles. This function serves as an example to showcase how the idea works in practice and how the system can be utilized. By developing this worker function, it becomes possible to generate simulated data that mimics the updates typically seen in Google’s trip and vehicle information. This demonstration helps to provide a clearer understanding of the underlying idea and its potential applications.

7.5 Scalability

Scalability refers to the capability of a system to handle increased or decreased workloads effectively and efficiently. It measures how well a system can adapt to changes in application and processing demands without sacrificing performance or significantly increasing costs.

To evaluate the scalability of the project, several popular monitoring and observability tools were incorporated. Prometheus, Grafana, and cAdvisor are commonly used tools in the field of IT and system administration. Each tool serves a specific purpose:

- Prometheus ²: Prometheus is an open-source monitoring and alerting system. It is designed to collect metrics from various sources, such as applications, servers, and databases, and store them in a time-series database. Prometheus provides a flexible query language for analyzing and querying metrics data. It also offers powerful alerting capabilities, allowing users to define rules and receive notifications when certain conditions are met. Prometheus is widely used for monitoring and troubleshooting the performance of systems and applications in real-time.

cAdvisor (1/1 up) View					
Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://cAdvisor:8080/metrics	UP	group="cAdvisor" instance="cAdvisor-8080" job="cAdvisor"	19.970s ago	484.935ms	
Prometheus (1/1 up) View					
Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://django:8000/prometheus-xyzabc/metrics	UP	instance="django:8000" job="prometheus"	12.343s ago	4.536ms	

Figure 7.14: Display of Prometheus running in communication with Django and cAdvisor.

- Grafana ³: Grafana is an open-source data visualization and monitoring tool. It integrates with various data sources, including Prometheus, to create visually appealing dashboards and graphs for monitoring and analyzing metrics data. Grafana offers a user-friendly interface for designing customizable dashboards, which can display metrics, logs, and other data in

²Prometheus documentation: <https://prometheus.io/docs/introduction/overview/>

³Grafana documentation: <https://grafana.com/docs/>

real time. It supports a wide range of data visualization options, including charts, graphs, tables, and heat maps. Grafana is often used alongside Prometheus to create informative and visually appealing monitoring dashboards.



Figure 7.15: Display of Grafana Dashboard, analysing multiple metrics from Prometheus.

- cAdvisor ⁴: cAdvisor, short for Container Advisor, is an open-source tool specifically designed for container monitoring. It collects resource usage and performance metrics from running containers and provides insights into their resource consumption, such as CPU usage, memory usage, and network statistics. cAdvisor automatically discovers containers running on a host and provides real-time metrics for each container. It is commonly used to monitor the health and performance of containerized applications.

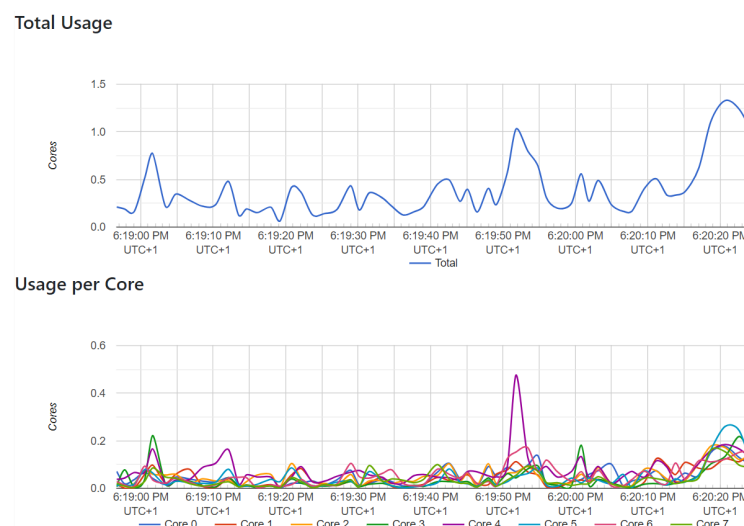


Figure 7.16: Display of cAdvisor CPU metrics, such as Total Usage and Usage per Core, from Prometheus.

Prometheus, Grafana, and cAdvisor form a powerful stack for monitoring, collecting, storing, querying, visualizing, and analyzing metrics and performance data. Together, they enable developers to gain valuable insights into the health, performance, and behavior of systems, applications, and containers.

⁴cAdvisor documentation: <https://github.com/google/cadvisor/blob/master/docs/overview.md>

The project was integrated into a private server provided by Ubiwhere, allowing the backend to be hosted. This ensured consistent request duration and execution times, facilitating easier comparisons between different scenarios.

7.6 Summary

In conclusion, the API Prototype chapter highlighted the iterative design process, the selection of Django as the framework, the development of various models, the challenges faced in implementing autonomous vehicles and public transportation schedules, and the visualization and real-time update capabilities of the SUMO API. These efforts laid the foundation for a powerful and flexible API that can simulate and analyze traffic scenarios effectively.

Chapter 8

Results

The results chapter presents a comprehensive analysis of the outcomes obtained from the conducted experiments and simulations. This chapter aims to showcase the multiple phases of the simulations and what possible analysis can be done, by presenting multiple scenarios created.

These scenarios encompass a wide range of functionalities, incorporating various implemented models. They span from creating road networks and vehicles to analyzing diverse results provided by SUMO.

It is important to note that all scenarios were implemented in Porto, a city that has partnered with Ubiwhere in multiple projects. This collaboration facilitated access to essential resources and support for the project. Moreover, Porto was chosen because it is a Portuguese city with publicly available public transportation data, provided online by Sociedade de Transportes Coletivos do Porto ¹.

Through the exploration of these scenarios, the practical application and effectiveness of the implemented tool in a real-world context are demonstrated. The scenarios offer insights into numerous actions and analyses, contributing to the overall objectives of the thesis project.

8.1 Baseline Scenario

The first scenario implemented in the thesis project focuses on simulating a normal day in the city of Porto. This scenario aimed to provide a simplified representation of daily activities without incorporating any specific events or incidents.

To accurately depict the road network of the city, publicly available data from OpenStreetMaps was utilized. This data, provided by Ubiwhere, served as a reliable source for creating a comprehensive road network model.

Next, the scenario involved generating a fleet of private vehicles and distributing them across the city throughout the day. This step aimed to mimic the typical

¹Sociedade de Transportes Coletivos do Porto: <http://www.stcp.pt>

movement patterns of private vehicles in Porto. The number of vehicles created for this scenario was 500,000, and they were uniformly distributed over the day.

In addition to private vehicles, the simulation also accounted for public transportation vehicles, such as buses. Open data was utilized to include these vehicles in the simulation, ensuring a more realistic representation of the city's transportation system.

Once all the necessary elements were in place, a simulation model was constructed. This model determined the methods and models to be used for obtaining the final results. It allowed for organized data analysis based on vehicles, and roads, as well as general simulation metrics like the simulation duration and total distance covered.

The simulation model was executed, generating results that were then prepared for visualization and analysis. These results provide valuable insights into the behaviors and patterns observed during a typical day in Porto. They contribute to the overall understanding of the city's transportation dynamics and help evaluate the effectiveness of the implemented tool.

In the subsequent sections, we will delve deeper into the specific outcomes and observations obtained from this scenario, offering a comprehensive analysis of a normal day in Porto's transportation system.

8.1.1 Porto Roadmap

The purpose of this display is to demonstrate the potential for visualizing the road map of Porto. Users can use it to determine if the network of roads has been accurately constructed based on the provided bounding box. Additionally, they can explore the database to examine other stored road maps and assess if any previous models can be reused for the current project.

Additionally, this example is also important to view the identification of all the roads. This capability is essential for effectively incorporating and representing various incidents or events that may take place along these identified roads within the road map.



Figure 8.1: Display of Porto road map used for the baseline scenario.

The image shown above provides a visual representation of the Porto road map used in the baseline scenario. This map serves as a fundamental tool for analyzing and interpreting the data, allowing researchers and transportation professionals to explore the intricacies of the road network and its associated features. The accurate depiction of the road map is vital for studying traffic patterns, identifying congestion hotspots, and devising effective strategies for improving traffic flow and overall transportation management.

8.1.2 Porto Bus Stops

The objective of this display is to showcase the translation of bus stops from GTFS to SUMO via the API. Through this analysis, users can validate the accuracy of public transportation stops and even compare them with the bus stops defined in the GTFS data. This feature provides users with a means to ensure the correct representation of bus stops within the SUMO simulation and verify their alignment with the corresponding GTFS data.



Figure 8.2: Analysis of Bus Stops (a) in GTFS data and (b) in SUMO

Figure 8.2 showcases the analysis of bus stops, illustrating a side-by-side comparison between the GTFS data (a) and the corresponding representation in SUMO (b). This comparison allows users to assess the consistency and fidelity of the bus stops between the two systems. By visually inspecting the alignment and attributes of the bus stops, users can ensure the accuracy of the translation process and confirm that the bus stops in SUMO faithfully reflect the information provided in the GTFS dataset.

8.1.3 Traffic Congestion Analysis

To effectively visualize the extent of traffic congestion in the city of Porto, numerous parameters provided by SUMO can be utilized. These parameters include density, vehicles per kilometer, and occupancy of the edge, expressed as a percentage. These metrics offer valuable insights into the level of congestion experienced on various roads within the city.

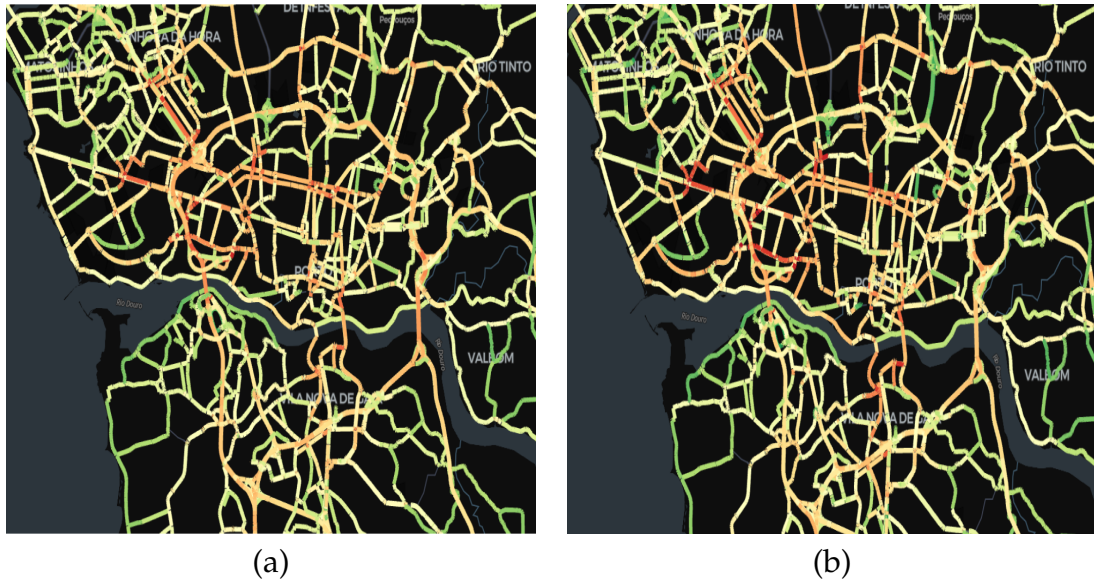


Figure 8.3: Visualization of density and occupancy in Porto, between 14h and 17h

From these visualizations, it is evident that certain zones within Porto experience higher levels of traffic congestion, particularly in the city center and within the inner ring road known as the Porto VCI. Conversely, residential areas tend to exhibit lower traffic congestion during the day, aligning with expectations.

An intriguing observation is that certain streets, despite having a red indicator for occupancy, indicating a high percentage of vehicles on the road, may not stand out as significant in the density figure. This discrepancy can be attributed to the influence of vehicle size. For instance, a bus would have a greater impact on the occupancy value of a road compared to a private vehicle, whereas in density calculations, both a bus and a private vehicle are treated equally.

These nuanced differences give rise to an interesting comparison between the two parameters, emphasizing the importance of visualizing and considering both when analyzing the city's traffic congestion. By examining density and occupancy, a more comprehensive understanding of traffic patterns and congestion hotspots can be achieved, enabling effective strategies and interventions to alleviate congestion and enhance traffic management in Porto.

8.1.4 Speed Study

Another crucial parameter that can be assessed through the API is the analysis of speed within the city. Each road is assigned a maximum speed limit, determined by a responsible organization based on various factors. Generally, humans tend to adhere to these speed limits and road guidelines [43]. By leveraging this understanding, it becomes possible to gauge congestion levels on specific roads by examining the average speed of vehicles traveling along them.

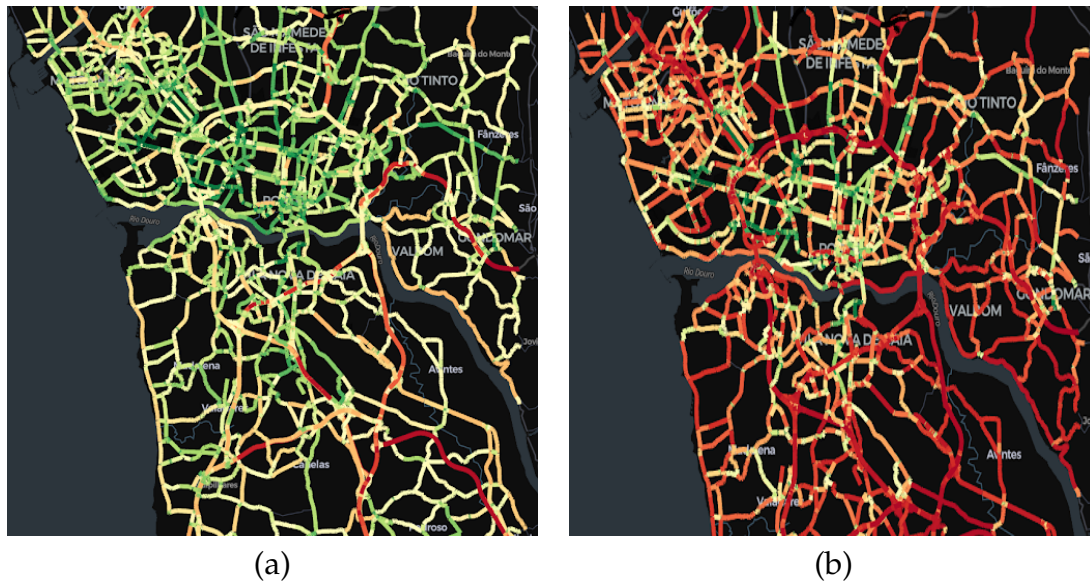


Figure 8.4: Breakdown of average speed and relative average speed in Porto, between 14h and 17h

However, relying solely on the average speed of a road provides limited information, mainly indicating fluctuations in average speeds along its length. To gain deeper insights into the impact of congestion, it is more effective to consider the relative average speed of the road, which compares the actual average speed with the maximum speed limit. This approach allows for a clearer understanding of areas where vehicles were unable to reach higher speeds, not due to imposed speed limits, but as a result of traffic congestion.

By analyzing the relative average speed, it becomes easier to identify sections of roads that experience significant congestion, facilitating targeted interventions and strategies to alleviate traffic bottlenecks and improve overall traffic flow.

8.1.5 Environmental Impact

Three crucial key performance indicators were provided, enabling the visualization of vehicle emissions during the simulation: CO₂ (Carbon Dioxide), PM (particulate matter), and NO_x (Nitrogen oxide). Consequently, features were developed to depict the total emissions, emissions normalized by time and road length and even emissions per vehicle.

These visualizations serve not only to analyze CO₂, PM, and NO_x emissions but also to assess CO (carbon monoxide), HC (hydrocarbons), fuel and electricity consumption.

To streamline the presentation in this section, the following simplified displays are presented: the total CO₂ emissions, PM values based on hourly and edge length measurements, and NO_x values per vehicle.

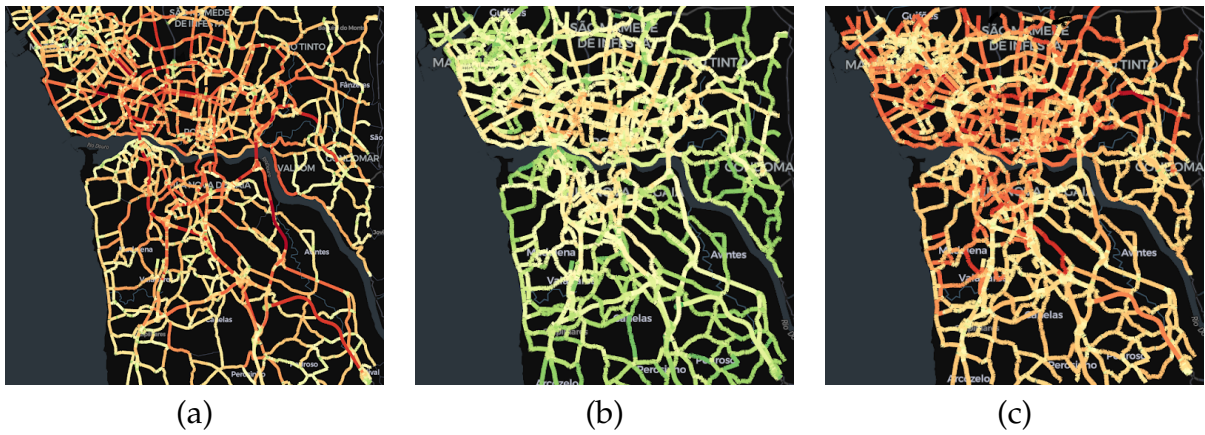


Figure 8.5: Visualization of (a) total emissions of CO₂, (b) emissions of PM_x per hour per road length, and (c) emissions of NO_x by vehicle. City of Porto, between 14h and 17h

The availability of these visualizations plays a crucial role in assisting governmental entities in comprehending and addressing critical pollution hot spots within the city. By utilizing these features, decision-makers gain valuable insights into areas with high emissions of CO₂, PM, and NO_x, allowing them to prioritize targeted interventions and implement effective measures to improve the overall quality of life for residents.

8.1.6 Result Overall Statistics

The simulation results provide valuable insights into various aspects, including performance metrics and general trip information. By analyzing the simulation data, users can access a wealth of information about the system's performance, such as travel times and duration of the simulation. Additionally, they can gather overall trip details, such as the number of trips made, and distances traveled. This comprehensive information allows for a comprehensive understanding of the transportation system's efficiency and aids in making informed decisions for future improvements or optimizations.

	title	value
0	clockBegin	1687183163.81
1	clockEnd	1687200324.28
2	clockDuration	17160.47
3	traciDuration	2766.55
4	realTimeFactor	5.33
5	vehicleUpdatesPerSecond	22620.43
6	personUpdatesPerSecond	0.00
7	begin	0.00
8	end	91514.00
9	duration	91514.00

Figure 8.6: Example of overall statistics that can be gathered from the baseline simulation scenario.

8.1.7 Vehicle Trip information

The availability of vehicle-grouped information enables the analysis of extensive data sets. To enhance the user experience and facilitate data exploration, a dedicated component was developed to plot these parameters against each other within the application. This feature serves the purpose of maximizing the amount of information that users can extract from the application, empowering them to gain valuable insights and draw meaningful conclusions from the plotted data.

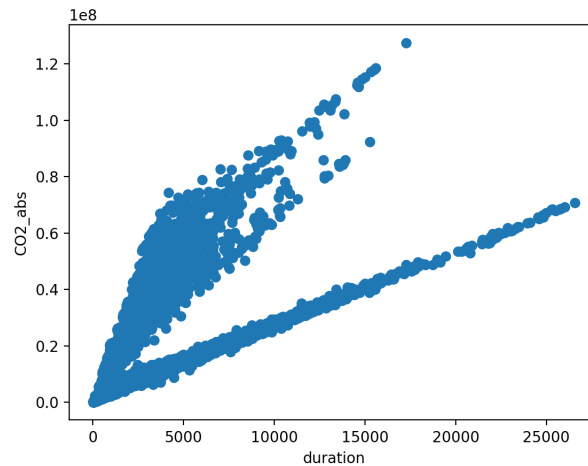


Figure 8.7: Example of a plot created with the duration of trip and CO2 emitted by the vehicle. Each point corresponds to a vehicle.

By examining the plotted data, a clear division between two distinct groups of vehicles becomes apparent. The first group exhibits a linear relationship between the duration of the trips and the amount of CO2 emissions. As the duration increases, so does the CO2 emission in a proportionate manner. On the other hand, the second group of vehicles does not adhere to any discernible pattern. It emits significantly higher levels of CO2 compared to the vehicles in the linear group.

One potential explanation for this discrepancy is the impact of traffic congestion and stoppage time. When vehicles experience prolonged stops in traffic, they tend to emit higher amounts of CO2 emissions due to idling and inefficient engine performance. This factor could contribute to the elevated CO2 levels observed in the second group.

To further investigate this phenomenon, a second plot was generated, this time depicting the relationship between waiting time and CO2 emissions. The resulting plot revealed a similar pattern of two distinct vehicle groups. This correlation suggests that the waiting time in traffic is not a significant contributing factor to the increased CO2 emissions observed in the second group of vehicles.

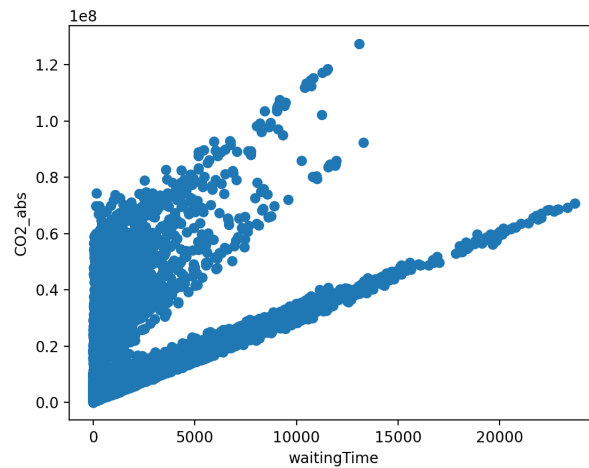


Figure 8.8: Plot created with the waiting time and CO2 emitted by the vehicle. Each point corresponds to a vehicle.

Through these plotted data analyses, users can gain deeper insights into the relationships between various parameters and the respective outcomes or variables of interest. By examining the patterns and anomalies within the data, researchers and professionals in the transportation field can uncover influential factors, identify opportunities for improvement, and develop targeted interventions to enhance environmental sustainability and overall system performance.

8.1.8 Simulation Time Series

To enhance the user's understanding of the simulation results, a dedicated component was developed to provide a summary of the information using time series analysis. This component serves the purpose of presenting a concise and comprehensive overview of the simulation's progression over time.

Using time series plots, the component showcases important metrics and variables that evolve throughout the simulation. It allows users to track changes, trends, and patterns over time, enabling them to gain deeper insights into the dynamics of the system being simulated.

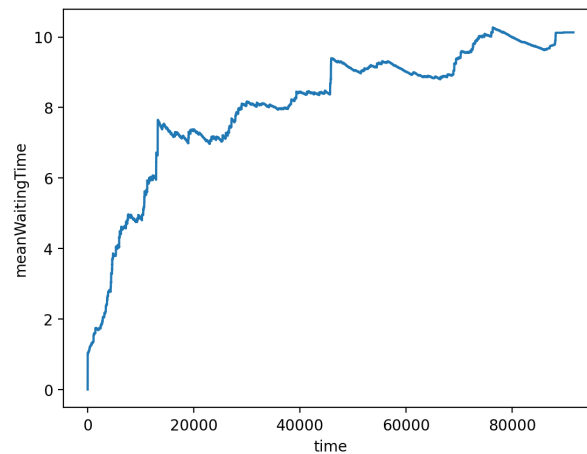


Figure 8.9: Example of a plot created using the time series component to visualize the mean waiting time (seconds) of the simulation.

The visualization component plays a crucial role in helping users comprehend the occurrence of traffic congestion peaks within the city. By presenting the data in an intuitive and visually appealing manner, it allows users to easily identify and understand the timing of these congestion spikes.

8.1.9 Vehicle Position and Statistics

A fascinating visualization option is the ability to observe the vehicle positions at a specific time interval. This feature provides valuable insights into the origins and extent of traffic congestion, as well as its duration when combined with data from other intervals.

Unfortunately, due to certain limitations in Streamlit, it was not feasible to generate a GIF-like showcasing of the behavior of each vehicle throughout the simulation over a period of time. Nonetheless, the feature implemented does differentiate between private vehicles and public transportation by representing personal vehicles with red points and buses with yellow points.

For this particular analysis, the chosen time interval corresponds to approximately 15:00, allowing for a focused examination of traffic conditions during that specific period.

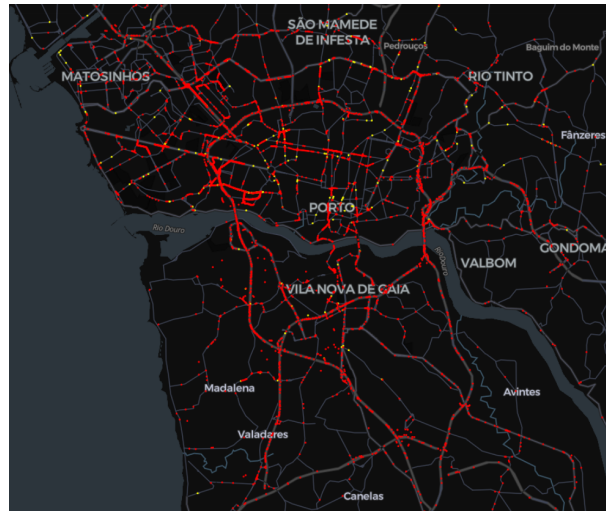


Figure 8.10: Visualization of vehicle position in Porto at 15h. Red points correspond to private vehicles and yellow points to public transportation vehicles.

By examining this figure, it becomes apparent that the simulation within the Porto VCI area depicts a higher density of vehicles. Furthermore, it enables us to pinpoint potential areas where traffic congestion occurs due to this increased vehicular density, particularly concentrated on the left side of the inner ring.

When visualizing the vehicle position, one can access comprehensive data regarding the simulation's general aspects, using the same timestamp. This includes information such as the total number of vehicles that have been loaded, the current count of active running vehicles, the occurrences of errors such as teleports or collisions, the average speed of vehicles, waiting times, and numerous other relevant metrics.

time	54000.00
loaded	300422
inserted	297768
running	4894
waiting	53
ended	292874
arrived	292874
collisions	22
teleports	6473
halting	1379
stopped	56
meanWaitingTime	9.15
meanTravelTime	803.87
meanSpeed	12.29
meanSpeedRelative	0.51
duration	216

Figure 8.11: Visualization of statistical information about the simulation of Porto at 15h.

This data table provides valuable insights into the simulation's progress and per-

formance, enabling analysis of traffic density, the efficiency of traffic control measures, identification of issues or errors, and informed decision-making for traffic management and planning.

8.1.10 Simulated Trip Updates for Public Transportation

One of the main functionalities created for the SUMO API is the usage of the simulation results to simulate trip updates using the Google standard, GTFS RT. Using the baseline scenario it is possible to create and visualize trip updates and vehicle position.

```
header {
  gtfs_realtime_version: "2.0"
  incrementality: FULL_DATASET
  timestamp: 1687429555
}
entity {
  id: "106_1_U_13"
  trip_update {
    trip {
      trip_id: "106_1_U_13"
      start_time: "37200"
      start_date: "20230622"
      schedule_relationship: SCHEDULED
      route_id: "106"
      direction_id: 1
    }
    stop_time_update {
      stop_sequence: 4
      departure {
        delay: 4
        time: 37339
      }
      stop_id: "ACBL2"
      schedule_relationship: SCHEDULED
    }
    # stop_time_updates removed to simplify visualization
    vehicle {
      id: "106_1_U_13.0"
    }
  }
}
```

Listing 8.1: Example of a simulated trip update, with header and entity of a trip update. This trip update is composed of a trip, stop time updates, and vehicle entities

To simplify the presentation of the feed message, the next stop times were removed, as it was only to demonstrate the multiple entities that are created.

```
header {
  gtfs_realtime_version: "2.0"
  incrementality: FULL_DATASET
  timestamp: 1687429555
}
entity {
  id: "106_1_U_13.0"
  vehicle {
    trip {
      trip_id: "106_1_U_13"
      start_time: "37200"
      start_date: "20230622"
      schedule_relationship: SCHEDULED
      route_id: "106"
      direction_id: 1
    }
    position {
      latitude: 41.081092834472656
      longitude: -8.655570983886719
    }
    vehicle {
      id: "106_1_U_13.0"
    }
  }
}
```

Listing 8.2: Example of a simulated vehicle position, with header and entity of a vehicle entity. This vehicle entity is composed of a trip entity, position, and vehicle identifier

Creating and visualizing GTFS-RT feed messages have the potential to be instrumental in validating the values of a SUMO simulation. This validation process involves comparing the simulated feeds with feeds generated from GPS sensors. However, it's important to note that in the case of Porto, there might not be readily available open-source feed messages that can be utilized for such comparisons and validation. Therefore, the possibility of conducting this validation remains more of an afterthought due to the unavailability of suitable data sources for comparison.

8.1.11 Key Performance Indicators

To enhance the comprehension and evaluation of the simulation, a thoughtful approach was taken to group the key performance indicators (KPIs) within a table. This strategic arrangement allows for a more systematic analysis of the simulation results and offers a comprehensive view of the performance metrics.

By presenting the KPIs in a structured format, it becomes easier to quantify and measure the performance objectively. Each KPI represents a specific aspect of

performance that is crucial for assessing the effectiveness and progress toward the desired objectives.

In this particular simulation, the vehicles employed were powered by conventional fuels, resulting in the exclusion of electricity-related metrics, as none of the vehicles were electric. Additionally, the consideration of noise was eliminated from the evaluation process. While it is feasible to measure the total noise level per lane, creating an average for noise proved impractical due to the non-linear metric employed in the SUMO simulation framework.

KPIs	Values
Average Speed	57.096 km/h
Average Speed (private vehicle)	57.528 km/h
Average Speed (public transport)	20.952 km/h
Vehicle km traveled	10630 m
Vehicle stops	6.373 s
Non-scheduled stops	183.926 s
CO2 emissions	$3.69 * 10^6$ mg
PM emissions	146.85 mg
NOx emissions	4504.53 mg

Table 8.1: Table of KPIs obtained from the baseline scenario in Porto.

These values provide a clear depiction of the KPIs that can be utilized for comparison with other simulations. Furthermore, they can serve as a means to validate the accuracy of the simulations, as these values can be obtained directly from sensors and cameras.

By using these indicators, it also becomes feasible to compare subsequent scenarios and, ideally, gain a more comprehensive perspective on the results.

8.2 Occurrences Scenario

The second scenario created had the objective of presenting different existing options implemented to simulate random events and occurrences during a day in Porto.

Similar to what has been done in the previous scenario, the creation of the network is crucial for the execution, as well as the vehicles and public transportation. With the objective of better comparison between this scenario and the previous implementation, the same models of roads and vehicles were used. This choice of modulation allows for more compact and smaller models, that can be used in multiple simulations without the need of creating a new network or different trips every time.

With the objective of simulating these events and occurrences, some models were created. The first event created was an occurrence in Ponte da Arrábida, in the left-most lane in the direction Gaia-Porto, for a duration of 3 hours, between 15h and 18h, resulting in a blocked lane. This is one of the most important connections

between the two cities, creating a big disturbance in the movement of vehicles.

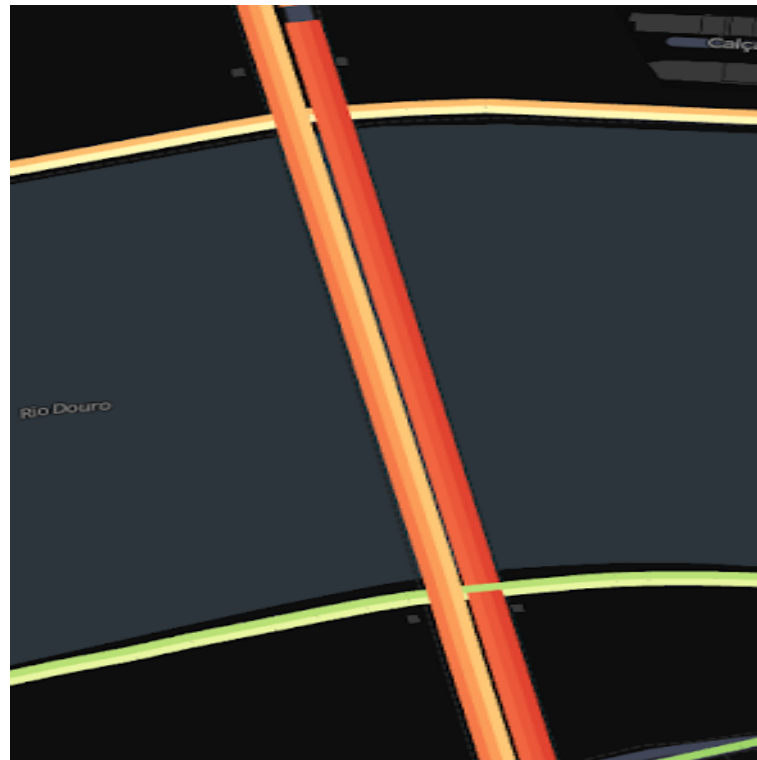


Figure 8.12: Visualization of Porto traffic congestion between 15h and 18h in the baseline scenario.

In order to make this occurrence more impactful, during the Y hours of the blocked lane, the middle lane in the same direction, had an impact in the maximum speed allowed, going from 22.2 meters per second, or approximately 80 kilometers per hour, to 13.8 meters per second, or approximately 50 kilometers per hour.

This double event allows us to have a better visualization of the changes in the movement of vehicles, enabling the possible study of secondary routes that these choose to reach their final destination.

8.2.1 Road Block and Speed Restriction

A display of the occurrences created to simulate this scenario was developed, emphasizing the highlighted models of the roads.

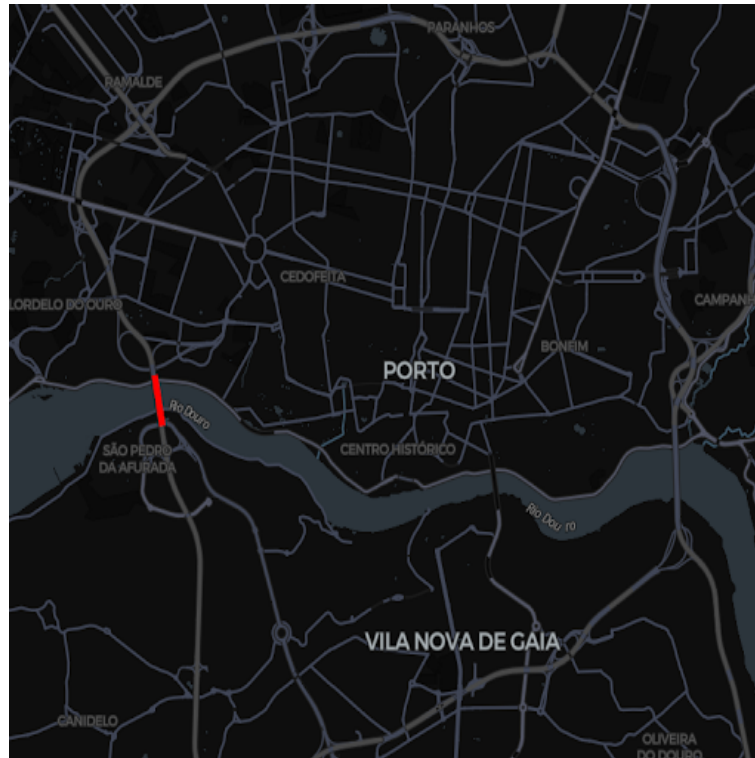


Figure 8.13: Analysis of Porto highlighted lane blocked in Ponte da Arrábida, created for the occurrences scenario.

Figure 8.13 illustrates the visualization of a lane blocked on Ponte da Arrábida in Porto, specifically designed for the occurrences scenario. The purpose of this roadblock is to restrict traffic flow on one of the heavily utilized bridges connecting Porto and Gaia. The intention is to generate significant effects that differ from the baseline scenario, enhancing the visualization and facilitating the comparison of results.

8.2.2 Comparison of Traffic Congestion with Baseline Scenario

To assess the impact of the occurrence on the road, a comparative visualization was developed. This feature normalizes the values from the two results and compares them, generating a simple interval. The interval is then converted into colors using a Green-White-Red color mapping scheme. In this scheme, the color green represents less traffic congestion in the Occurrences Scenario compared to the Baseline Scenario, while the color red indicates higher traffic congestion in the Occurrences Scenario. By employing this color mapping approach, the visualization provides a clear and intuitive representation of the traffic congestion levels resulting from the occurrence, facilitating a quick and comprehensive analysis.

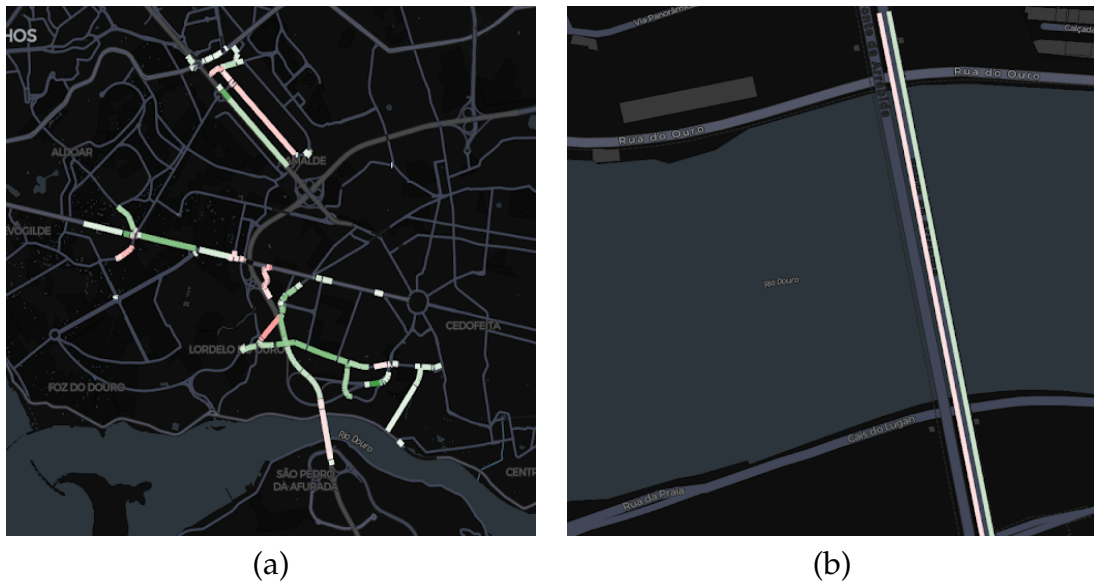


Figure 8.14: Visualization of comparison of density between the occurrence scenario with the baseline scenario.

To enhance the clarity of the figure, routes that exhibited insignificant differences between the two scenarios were excluded, allowing for a more concise representation of the crucial data.

In the visualization, green lanes indicate that the occurrence scenario experienced lower traffic congestion compared to the baseline scenario, while red lanes indicate higher traffic congestion in the occurrence scenario relative to the baseline. The intensity of the color corresponds to the magnitude of the difference observed.

From these results, we can observe the immediate impact in the lanes where the occurrence took place, which leads to a shift of vehicles towards the leftmost lane. Additionally, there are noticeable repercussions along the continuation of the highway, potentially caused by the occurrence, resulting in increased spacing between vehicles.

Furthermore, there are certain areas within the city where disparities were observed, but a direct correlation to the occurrence could not be established.

8.2.3 Comparison of KPIs with Baseline Scenario

Upon conducting a visual analysis of the results using a map visualization, the next step was to compare the key performance indicators (KPIs) to potentially uncover new insights. The comparison was based on the values provided in the table below:

KPIs	Baseline Scenario	Occurrences Scenario
Average Speed	57.096 km/h	57.024 km/h
Average Speed (private vehicle)	57.528 km/h	57.492 km/h
Average Speed (public transport)	20.952 km/h	20.938 km/h
Vehicle km traveled	10630 m	10631 m
Vehicle stops	6.373 s	6.393 s
Non-scheduled stops	183.926 s	180.129 s
CO2 emissions	$3.69 * 10^6$ mg	$3.68 * 10^6$ mg
PM emissions	146.85 mg	146.18 mg
NOx emissions	4504.53 mg	4486.09 mg

Table 8.2: Table of KPIs obtained from the Occurrences scenario in Porto and compared with Baseline Scenario.

After conducting a comprehensive analysis of the given values, it becomes apparent that there are no observable variations or significant trends between the baseline scenario and the occurrences scenario. Consequently, this suggests that the closure of a lane on the bridge between 15:00 and 18:00 had a limited impact on the overall city. It indicates that only a small portion of the city was affected by this particular intervention.

8.3 Road Restriction Scenario

One interesting scenario implemented in the project involved the limitation of certain vehicles to specific lanes or areas in the city. The purpose was to improve the quality of life in those areas and enhance the standard of public transportation within the city of Porto.

This scenario aimed to showcase the creation of this functionality and its potential benefits. Specifically, the focus was on limiting the access of vehicles to the right-most lane on important roads in Porto exclusively for public transport vehicles.

By implementing this method, the scenario demonstrated how such limitations can be utilized to optimize public transportation services and enhance the overall urban experience. The objective was to create a more efficient and reliable system that benefits both residents and the city as a whole.

Through this scenario, the project aimed to highlight the possibilities and advantages of strategically restricting vehicle access in specific lanes or areas. By exploring the implementation process and potential outcomes, this scenario showcased the transformative power of strategically limiting vehicle access in specific lanes or areas, leading to improved urban mobility and elevated public transportation standards.

In order to assess the effects of limiting private vehicle access and evaluate its impact on surrounding areas of the city, a decision was made to block Rua do Campo Alegre, a heavily congested street in Arrábida. This measure aimed to analyze the consequences of such restrictions and observe how they would influ-

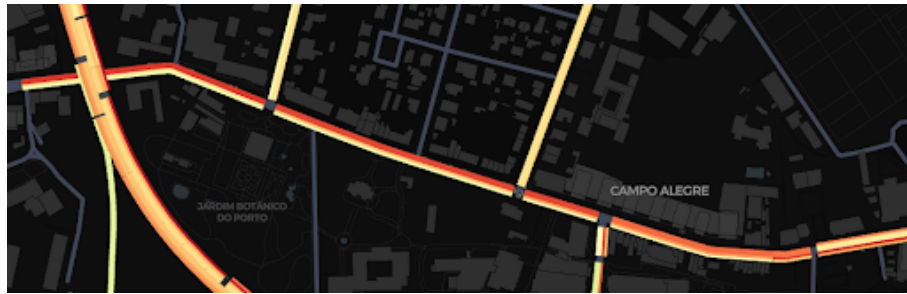


Figure 8.15: Visualization of Porto traffic congestion in Rua do Campo Alegre between 15h and 18h

ence neighboring zones.

8.3.1 Road Restriction

Similar to the roadblock visualization, the road restriction visualization was created to emphasize the highlighted roads where the occurrence happens.

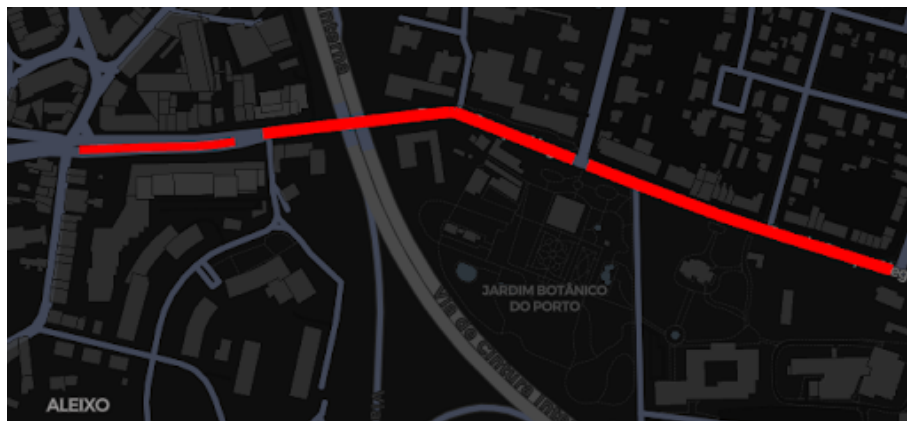


Figure 8.16: Visualization of a Porto street limited access to private vehicles.

The primary objective of implementing this road restriction to exclusively allow public transportation is to effectively manage and control the flow of traffic on particular roads located in the city center. By limiting access to only public transportation vehicles, the aim is to mitigate congestion and potentially reduce emissions in the area.

This road restriction serves as a means to analyze and assess the impact of such measures on various aspects, including traffic congestion and environmental factors.

8.3.2 Comparison of Traffic Congestion with Baseline Scenario

A comparison was created to evaluate the impact of road restrictions. This analysis aims to visualize the consequences on neighboring zones where private ve-

hicles are not permitted, investigate whether alternative routes are chosen, and observe the resulting behavior of individuals affected by the restrictions.



Figure 8.17: Visualization of Porto traffic congestion comparison in Rua do Campo Alegre between 15h and 18h with the baseline scenario.

One notable outcome that can be discerned from this comparison is the identification of alterations in the connections to Rua do Campo Alegre. The presence of traffic congestion often leads to dynamic changes in the road network as drivers seek alternative routes to avoid the congested area. Consequently, this comparative analysis allows us to detect modifications in the connectivity patterns and identify how traffic is diverted or redirected as a result of the congestion.

Moreover, this examination sheds light on the emergence of increased traffic congestion on alternative routes. As drivers navigate away from the congested Rua do Campo Alegre, there is a consequential redistribution of traffic along alternative paths. By carefully examining the comparison data, it is possible to observe the patterns of congestion intensifying on these alternative routes, providing a comprehensive understanding of the broader traffic dynamics influenced by the congestion incident.

However, it is crucial to recognize that the visualization alone does not provide a comprehensive explanation for the observed variations in traffic congestion levels in areas further away from the congestion incident. While the comparison allows for the identification of potential rerouting of vehicles, it is essential to consider additional factors that contribute to the observed disparities.

The impact of traffic congestion can propagate through the network, causing cas-

cading effects that go beyond the immediate vicinity of the incident. Delays, disruptions, and congestion in one area can lead to congestion spillover to adjacent roads or even affect distant parts of the network. Consequently, the observed disparities in congestion levels in areas further away may be a result of these cascading effects. However, without further investigation or data, it remains speculative to attribute the observed disparities solely to these ripple effects

8.3.3 Comparison of KPIs with Baseline Scenario

After conducting a visual examination of the results through a map visualization, the subsequent stage involved comparing the key performance indicators (KPIs) in order to potentially discover fresh insights. This comparison was carried out using the values presented in the table provided below:

KPIs	Baseline Scenario	Road Restriction Scenario
Average Speed	57.096 km/h	56.412 km/h
Average Speed (private vehicle)	57.528 km/h	56.844 km/h
Average Speed (public transport)	20.952 km/h	20.304 km/h
Vehicle km traveled	10630 m	10728 m
Vehicle stops	6.373 s	6.239 s
Non-scheduled stops	183.926 s	210.553 s
CO2 emissions	$3.69 * 10^6$ mg	$3.81 * 10^6$ mg
PM emissions	146.85 mg	153.58 mg
NOx emissions	4504.53 mg	4684.19 mg

Table 8.3: Table of KPIs obtained from the Restricted Access scenario in Porto and compared with Baseline Scenario.

After conducting a comprehensive analysis of the given values, a notable observation emerges. The average distance traveled by vehicles is found to be nearly 100 meters greater, which can be attributed to a considerable number of vehicles altering their route to avoid the modified street. This change in route selection indicates that drivers were actively adapting their travel paths in response to the altered conditions.

Furthermore, a noticeable difference can be observed in the number of non-scheduled stops. This increase in unscheduled stops suggests that congestion levels were higher in the scenario where the street was altered. The higher congestion can be attributed to the rerouting of vehicles, resulting in slower travel times and a greater number of stops along the way.

These findings not only highlight the impact of the alteration on the flow of traffic but also indicate the significance of the change for the city of Porto. The observed increase in congestion and changes in travel patterns emphasize the importance of carefully considering and evaluating the effects of such alterations on overall transportation efficiency and urban mobility.

8.4 Reserved Route for CCAM Scenario

One of the primary objectives of this thesis is to address the needs of IN2CCAM (insert full form of IN2CCAM). As part of this objective, a simulation scenario was specifically created to simulate connected and cooperative autonomous vehicles. To achieve this, two distinct methods were implemented to replicate the behavior of autonomous vehicles.

The first method, which is more straightforward, involves modifying the existing algorithm used to create normal vehicles in previous scenarios. By adjusting parameters such as vehicle type, speed, class, and most importantly, the decision-making model, the vehicles were transformed into autonomous vehicles. For this purpose, the CACC (Cooperative Adaptive Cruise Control) model, already available in SUMO, was employed. It is important to note that this method, although simple, does not entirely reflect reality, as there are regulations in Portugal and Europe that restrict the circulation of autonomous vehicles. Consequently, a dedicated route had to be created exclusively for the operation of these vehicles.

Taking these constraints into consideration, a second method was devised for the creation of autonomous vehicles. In this approach, the user is required to specify the lanes that will be designated for the autonomous vehicle route. These lanes are then effectively blocked off for other vehicles, ensuring the exclusive circulation of autonomous vehicles. Multiple autonomous vehicles are then simulated in a loop, maintaining equal distances between them. This setup simulates a specialized form of autonomous public transportation.

By implementing these two methods, the thesis project successfully emulates the behavior of connected and cooperative autonomous vehicles. The first method provides a simplified representation, while the second method introduces a more realistic approach by creating a dedicated route and simulating a specialized autonomous transport system.

8.4.1 Autonomous Road Restriction

Similar to the road restriction analysis, the road restriction visualization was created to emphasize the highlighted roads where the road will be limited to autonomous vehicles.



Figure 8.18: Visualization of Porto highlighted CCAM route for autonomous vehicles.

The goal of implementing this road restriction for public transportation is to regulate the traffic flow on particular roads within the city center by exclusively permitting access to autonomous vehicles and analyzing its impact on the traffic congestion of the city.

8.4.2 Comparison of Traffic Congestion with Baseline Scenario

A comparison was created to evaluate the impact of road restrictions on autonomous vehicles circulating. This analysis aims to visualize the consequences on neighboring zones where private vehicles are not permitted, investigate whether alternative routes are chosen, and observe the resulting behavior of individuals affected by the restrictions.



Figure 8.19: Visualization of Porto comparison density of autonomous route created and the baseline scenario.

The density comparison allows us to observe the influence of the traffic density on the roads adjacent to the riverside, indicating a decrease in traffic congestion

on the closed route designated for autonomous vehicles. However, when examining other roads in Porto, we notice disparities that cannot be directly attributed to the closed route alone. It is plausible that the impact of the closure might have propagated to other areas of the city, leading to the observed differences. Nevertheless, it is important to acknowledge the possibility of errors or other contributing factors that could have influenced the results.

While it provides a valuable starting point for understanding the impact of the closed route on traffic congestion, further investigation and analysis are necessary to establish a comprehensive understanding of the underlying causes and implications. This may involve incorporating additional data sources, conducting statistical analyses, and considering alternative scenarios to account for the complexities and uncertainties inherent in the dynamics of urban traffic.

8.4.3 Comparison of KPIs with Baseline Scenario

Following a visual examination of the comparison results using a map visualization, the subsequent phase entails a comparative analysis of the key performance indicators (KPIs) in order to potentially reveal novel insights. This comparative assessment relied on the data values presented in the table provided below:

KPIs	Baseline Scenario	Reserved Route for CCAM
Average Speed	57.096 km/h	56.268 km/h
Average Speed (private vehicle)	57.528 km/h	56.700 km/h
Average Speed (public transport)	20.952 km/h	21.018 km/h
Average Speed (autonomous vehicle)	-	17.856 km/h
Single vehicle km traveled	10630 m	10741 m
Vehicle stops	6.373 s	6.170 s
Non-scheduled stops	183.926 s	224.171 s
CO2 emissions	$3.69 * 10^6$ mg	$3.85 * 10^6$ mg
PM emissions	146.85 mg	154.95 mg
NOx emissions	4504.53 mg	4715.04 mg

Table 8.4: Table of KPIs obtained from the baseline scenario in Porto.

Similar to the previous scenario, the data presented in this table unveils a notable increase in both the average distance traveled and the number of non-scheduled stops by vehicles. These findings can be attributed to specific factors that have influenced transportation dynamics.

Firstly, the observed increase in average distance traveled can be explained by a route restriction that limits access exclusively to autonomous vehicles. This restriction forces non-autonomous vehicles to alter their routes, resulting in longer distances traveled on average. The introduction of this restriction has likely prompted a change in the overall route preferences of drivers, leading to the observed increase.

Secondly, the rise in the number of non-scheduled stops by vehicles can potentially be linked to the influx of vehicles changing their routes and subsequently

perturbing traffic congestion on other roads. The altered routes taken by a greater number of vehicles can disrupt the flow of traffic, leading to additional stops and delays. This, in turn, increases the number of non-scheduled stops recorded in the data.

These findings highlight a distinct shift in transportation patterns and dynamics compared to previous observations. By delving into these metrics, we can gain valuable insights into the evolving trends and factors that contribute to the observed variations in vehicle behavior. Understanding these factors is crucial for optimizing traffic management strategies, enhancing overall transportation efficiency, and mitigating congestion-related issues.

Chapter 9

Conclusion

This final chapter brings together the key findings and insights from the preceding chapters, culminating in a comprehensive conclusion to the research presented in this thesis. Additionally, it explores potential avenues for future work that can build upon the current study and the utilization of the API. This chapter aims to provide a sense of closure to the research and implementation undertaken while also stimulating further inquiry and exploration.

9.1 Study Findings

Innovations utilizing cutting-edge technology and data analysis are being implemented in smart cities to enhance the well-being of citizens and make the environment more sustainable for living. This work addresses this goal by implementing a traffic simulation integrated system that is fed data from Ubiwhere's solution, Urban Platform, to help the municipalities visualize the results of anomalies in the road network, such as changes to the number of lanes, or changes in the public transportation routes and schedules.

To create this integrated system, a set of requirements was gathered with a group of Ubiwhere staff members. The requirements were then categorized using various classification methods, such as SMART Objectives and MoSCoW methodology.

Subsequently, the architecture of the integrated system was created, based on the C4 model, going from a high-level interpretation of the architecture to a more in-depth explanation of the application.

After understanding which was the objective of the project and which was the purpose, research was done on the subjects of traffic flow, the simulation tools that are used by the community of traffic simulation, and what capabilities each has implemented. A comparative analysis of the frameworks was conducted, to choose the best simulation tool for the gathered requirements and for the architecture that is to be implemented. In the end, it was agreed that the best traffic simulation, according to the collected requirements, the SUMO framework would be

the best tool for the integrated system, and therefore, the programming language would be Python.

In the meantime, the planning for the first and second semesters was performed, where the expectations of what would be possible to achieve are delineated. A Risk assessment analysis was made with the objective of creating awareness of the risks and implementing mitigation plans for each risk.

The second semester marked the initiation of the development of the integrated system, which began with the creation of a proof of concept using Flask. The primary objective of this initial step was to gain valuable insights into the specific requirements and limitations associated with the SUMO framework and its corresponding libraries.

Upon the completion of the proof of concept, a crucial decision was made to transition the development process to Django. This shift was primarily motivated by the numerous advantages offered by Django's Object-Relational Mapping (ORM) capabilities. Subsequently, multiple models and relations were meticulously implemented, and a diverse range of endpoints were meticulously designed and developed. These endpoints played a pivotal role in facilitating seamless operations such as data creation, modification, and previewing within the application.

As the API implementation progressed, a new framework was introduced to showcase the extensive functionalities that had been incorporated. This entailed the creation of various endpoints, which allowed users to effortlessly create models, execute simulations, and access the results in multiple formats. The overarching aim was to empower end-users to conduct comprehensive analyses and make well-informed decisions based on the data provided.

Following the successful implementation of the API, the next step involved deploying it on an internal server at Ubiwhere. To ensure optimal performance, scalability, and efficiency, monitoring containers were strategically employed. This meticulous setup enabled a thorough assessment of the system's performance and laid the foundation for future growth and expansion, providing reliable infrastructure for ongoing evaluation and enhancement.

9.2 Future Work

In future research, it will be crucial to assess the validity of the findings by conducting evaluations that involve comparing the simulated values with real-world traffic data from a city. Specifically, the density values obtained from the simulation should be compared to the actual traffic density observed on the roads. Furthermore, an interesting avenue for investigation would be to compare the simulated GTFS (General Transit Feed Specification) real-time values with real sensor data gathered from public transportation systems.

To comprehensively understand the capabilities and limitations of the simulation, multiple scalability tests should be performed. These tests would explore various aspects such as the size of the road network, the number of vehicles sim-

ulated, and even the size of the resultant files generated by the simulation. By conducting these tests, researchers can gain insights into the simulation's ability to handle larger and more complex scenarios and identify any potential bottlenecks or limitations.

Additionally, the SUMO platform provides a variety of supplementary functionalities that can be incorporated into the API. This integration adds an extra layer of complexity to the simulation, thereby offering enhanced value. These functionalities can enhance the simulation's capabilities by incorporating factors such as traffic signal control, advanced routing algorithms, or even specific vehicle behaviors.

Exploring these possibilities would allow for a more accurate representation of real-world traffic scenarios and provide a more comprehensive simulation environment.

In terms of integration with Ubiwhere, there are exciting opportunities to delve into the automation of occurrence creation in cities. By harnessing the power of Porto Open Data on traffic conditions, we can leverage condition types, observations, and polygon geometry to facilitate the selection of specific incidents among the multitude of possibilities. This integration enables the automatic generation of occurrences based on the available data, enhancing the efficiency and accuracy of incident management systems.

Furthermore, another avenue for integration involves utilizing the rich pool of traffic data to simulate the behavior and movement of vehicles within the city. By incorporating this traffic data into the simulation, a more realistic and immersive experience can be achieved, allowing for better analysis, planning, and decision-making in urban environments.

To facilitate these data-intensive processes, an asynchronous approach can be employed using Celery, a powerful distributed task queue system. By offloading the processing tasks to separate workers, the system can efficiently handle the creation of new models specific to each day. This asynchronous processing enables scalability, flexibility, and better resource management, ensuring smooth and timely execution of the data processes.

By exploring these integration possibilities and employing asynchronous data processing techniques with Celery, Ubiwhere can unlock new levels of efficiency, accuracy, and realism in urban incident management and simulation scenarios.

References

- [1] European Commission, *Smart Cities in Europe: Best Practices and the Way Forward*. Brussels, Belgium: European Union, 2014. [Online]. Available: https://smart-cities-marketplace.ec.europa.eu/sites/default/files/2021-04/the_making_of_a_smart_city_-_best_practices_across_europe.pdf
- [2] —, “Smart Cities,” 2023, accessed on July 3, 2023. [Online]. Available: https://commission.europa.eu/eu-regional-and-urban-development/topics/cities-and-urban-development/city-initiatives/smart-cities_en
- [3] T. E. E. COMMUNICATION FROM THE COMMISSION TO THE EUROPEAN PARLIAMENT, THE COUNCIL, S. COMMITTEE, and T. C. O. T. REGIONS, “Sustainable and smart mobility strategy – putting european transport on track for the future SWD(2020) 331 final,” 2020.
- [4] E. Union, “Horizon europe (horizon),” 2021.
- [5] E. C. D. C, “Horizon 2020 - work programme 2014 – 2015,” 2015.
- [6] D. J. Drucker, *Fertility Technology*, ser. The MIT Press Essential Knowledge series. London, England: MIT Press, Mar. 2023.
- [7] S. Brown, “The c4 model for visualising software architecture,” *C4 Model*. URL: <https://c4model.com/>(visited on Mar. 13, 2021), 2011.
- [8] twi global, “What is simulation? what does it mean?” last accessed 1 October 2022. [Online]. Available: <https://www.twi-global.com/technical-knowledge/faqs/faq-what-is-simulation#WhatDoesitMean>
- [9] W. Schwarting, J. Alonso-Mora, and D. Rus, “Planning and decision-making for autonomous vehicles,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 187–210, 2018. [Online]. Available: <https://doi.org/10.1146/annurev-control-060117-105157>
- [10] Q. Chao, H. Bi, W. Li, T. Mao, Z. Wang, M. C. Lin, and Z. Deng, “A survey on visual traffic simulation: Models, evaluations, and applications in autonomous driving,” *Computer Graphics Forum*, vol. 39, no. 1, pp. 287–308, 2020. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13803>
- [11] S. S. S. M. Qadri, M. A. Gökçe, and E. Öner, “State-of-art review of traffic signal control methods: challenges and opportunities,” *European Transport Research Review*, vol. 12, no. 1, p. 55, Oct 2020. [Online]. Available: <https://doi.org/10.1186/s12544-020-00439-1>

- [12] B. Himite, "Simulating traffic flow in python," 2021, last accessed 4 October 2022. [Online]. Available: <https://towardsdatascience.com/simulating-traffic-flow-in-python-ee1eab4dd20f>
- [13] M. H. F. K. C. N. S. S. Bhargav Naidu Matcha, Satesh Narayana Namasiyayam and S. Y. E. Noum, "An overview of microscopic and macroscopic traffic models," 2020.
- [14] S. J. Jang, "Vehicle traffic simulation using 2d cellular automata in mathematica," in *Review of Computational Science and Engineering Volume 4, Issue 1*, 2018. [Online]. Available: <https://complexityeducation.org/wp-content/uploads/2018/08/RCSE-07.2018-FINAL.pdf#page=14>
- [15] S. Dorokhin, A. Artemov, D. Likhachev, A. Novikov, and E. Starkov, "Traffic simulation: an analytical review," vol. 918, no. 1. IOP Publishing, sep 2020, p. 012058. [Online]. Available: <https://doi.org/10.1088/1757-899x/918/1/012058>
- [16] H.-T. Zhao, X.-R. Liu, X.-X. Chen, and J.-C. Lu, "Cellular automata model for traffic flow at intersections in internet of vehicles," *Physica A: Statistical Mechanics and its Applications*, vol. 494, pp. 40–51, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0378437117312335>
- [17] D.-F. Xie, Z.-Z. Fang, B. Jia, and Z. He, "A data-driven lane-changing model based on deep learning," *Transportation Research Part C: Emerging Technologies*, vol. 106, pp. 41–60, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0968090X18315651>
- [18] M. Vechione, E. Balal, and R. L. Cheu, "Comparisons of mandatory and discretionary lane changing behavior on freeways," *International Journal of Transportation Science and Technology*, vol. 7, no. 2, pp. 124–136, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2046043017300886>
- [19] A. Kesting, "Mobil : General lane-changing model for car-following models," 2007.
- [20] S. Mohammadian, Z. Zheng, M. M. Haque, and A. Bhaskar, "Performance of continuum models for realworld traffic flows: Comprehensive benchmarking," *Transportation Research Part B: Methodological*, vol. 147, pp. 132–167, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0191261521000527>
- [21] Y. Xue, Y. Zhang, D. Fan, P. Zhang, and H.-d. He, "An extended macroscopic model for traffic flow on curved road and its numerical simulation," *Nonlinear Dynamics*, vol. 95, no. 4, pp. 3295–3307, Mar 2019. [Online]. Available: <https://doi.org/10.1007/s11071-018-04756-y>
- [22] F. de Souza, O. Verbas, and J. Auld, "Mesoscopic traffic flow model for agent-based simulation," *Procedia Computer Science*, vol. 151, pp. 858–863, 2019, the 10th International Conference on Ambient Systems, Networks and Technologies (ANT 2019) / The 2nd International Conference on

- Emerging Data and Industry 4.0 (EDI40 2019) / Affiliated Workshops. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050919305824>
- [23] J. Nguyen, S. T. Powers, N. Urquhart, T. Farrenkopf, and M. Guckert, "An overview of agent-based traffic simulators," *Transportation Research Interdisciplinary Perspectives*, vol. 12, p. 100486, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2590198221001913>
- [24] M. Ziemka-Osuch and D. Osuch, "Modeling the assessment of intersections with traffic lights and the significance level of the number of pedestrians in microsimulation models based on the ptv vissim tool," *Sustainability*, vol. 14, no. 14, 2022. [Online]. Available: <https://www.mdpi.com/2071-1050/14/14/8945>
- [25] F. Vrbanić, D. Čakija, K. Kušić, and E. Ivanjko, *Traffic Flow Simulators with Connected and Autonomous Vehicles: A Short Review*. Cham: Springer International Publishing, 2021, pp. 15–30. [Online]. Available: https://doi.org/10.1007/978-3-030-66464-0_2
- [26] M. Muchlisin, I. Tajudin, and W. Widodo, "Optimization model of unsignalized intersection to signalized intersection using ptv. vissim: Study case in imogiri barat and tritunggal intersection, yogyakarta, indonesia," vol. VOL.11NO.9, p. 011–025, 01 2019.
- [27] T. Kučera and J. Chocholáč, "Design of the city logistics simulation model using ptv vissim software," *Transportation Research Procedia*, vol. 53, pp. 258–265, 2021, international Scientific Conference "Horizons of Railway Transport 2020". [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2352146521001939>
- [28] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "Sumo – simulation of urban mobility: An overview," in *SIMUL 2011*, S. . U. of Oslo Aida Omerovic, R. I. R. T. P. D. A. Simoni, and R. I. R. T. P. G. Bobashev, Eds. ThinkMind, October 2011. [Online]. Available: <https://elib.dlr.de/71460/>
- [29] P. Barbecho Bautista, L. Urquiza Aguiar, and M. Aguilar Igartua, "How does the traffic behavior change by using sumo traffic generation tools," *Computer Communications*, vol. 181, pp. 1–13, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0140366421003601>
- [30] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wiessner, "Microscopic traffic simulation using sumo," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2575–2582.
- [31] Aimsun, *Aimsun Next 22 User's Manual*, aimsun next 22.0.1 ed., Barcelona, Spain, Accessed on: Month, Day, Year 2022. [Online]. [Online]. Available: <https://docs.aimsun.com/next/22.0.1/>
- [32] K. W. A. Andreas Horni, Kai Nagel, "The multi-agent transport simulation matsim," 2022. [Online]. Available: <https://matsim.org/docs/userguide/>

- [33] H. Triebke, M. Kromer, and P. Vortisch, "Bridging the gap between mesoscopic transport planning and microscopic traffic simulation: An analytical and numerical analysis of traffic dynamics," *Transportation Research Record*, vol. 0, no. 0, p. 03611981221128284, 0. [Online]. Available: <https://doi.org/10.1177/03611981221128284>
- [34] *The Multi-Agent Transport Simulation MATSim*. Ubiquity Press, 2016. [Online]. Available: <http://www.jstor.org/stable/j.ctv3t5r7p>
- [35] D. Ziemke, I. Kaddoura, and K. Nagel, "The matsim open berlin scenario: A multimodal agent-based transport simulation scenario based on synthetic demand modeling and open data," *Procedia Computer Science*, vol. 151, pp. 870–877, 2019, the 10th International Conference on Ambient Systems, Networks and Technologies (ANT 2019) / The 2nd International Conference on Emerging Data and Industry 4.0 (EDI40 2019) / Affiliated Workshops. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050919305848>
- [36] matsim tools, "matsim-tools." [Online]. Available: <https://github.com/matsim-vsp/matsim-python-tools>
- [37] A. N. Laboratory.
- [38] B. D. M. J. K. F. K. M. M. T. S. J. Jedlicka, K., "Traffic modelling for the smart city of pilsen," *The 8th International Conference on Cartography and GIS 2020*, 2020. [Online]. Available: [https://iccgis2020.cartography-gis.com/8ICCGIS-Vol1/8ICCGIS_Proceedings_Vol1_\(56\).pdf](https://iccgis2020.cartography-gis.com/8ICCGIS-Vol1/8ICCGIS_Proceedings_Vol1_(56).pdf)
- [39] F. Kolovský, "Tramod pilsen." [Online]. Available: <https://plzen.trafficmodeller.com/>
- [40] J. M. Carew, "reinforcement learning." [Online]. Available: <https://www.techtarget.com/searchenterpriseai/definition/reinforcement-learning>
- [41] C. Wu, A. Kreidieh, K. Parvate, E. Vinitzky, and A. M. Bayen, "Flow: Architecture and benchmarking for reinforcement learning in traffic control," *CoRR*, vol. abs/1710.05465, 2017. [Online]. Available: <http://arxiv.org/abs/1710.05465>
- [42] S. Ngoma and P. Shamambo, "Transit mapping in lusaka, zambia. the first of its kind in lusaka." Ph.D. dissertation, 11 2016.
- [43] M. A. Elliott, C. J. Armitage, and C. J. Baughan, "Drivers' compliance with speed limits: An application of the theory of planned behavior," *Journal of Applied Psychology*, vol. 88, no. 5, pp. 964–972, 2003.