Report : Graduation Project Report

Mohsen Zouari

November 2020

Dedications

First and foremost, I thank "Allah" for guiding me throughout ....

I dedicate this work to my dearest Family, words can never express the deepest gratitude I have for them.

You have been there for me my whole life and I love you so much for it.

Thanks for always being there for me

Acknowledgments

I would like to express my sincere gratitude to the people who have allowed me to carry out my work through their sincere collaborations.

I would like to give special thanks to my honourable supervisor **Mr. Amine Kchiche** and **Mrs. Hajer taktak** for their proper direction, supporting assistance and guidelines in coordinating during the whole project. I am really grateful for their patience and the trust they have put on me.

I would also like to acknowledge all the jury members for their presence today for their collaboration during the examination of this work. Last but not least, my personal thanks go to all my professors for their efforts and the quality of the education during my study.

Contents

1 Context of the project **3**

1.1 Presentation of the context 4
1.2 Problematic 4
1.3 Nowadays Routing Criteria 5
1.4 Presentation of the project 5
1.5 State of the art 6
1.5.1 Transportation and environmental impact 6
1.5.2 ITS (INTELLIGENT TRANSPORTATION SYSTEMS) 6
1.5.3 Routing models 9
1.5.4 Eco-routing models 11
1.5.5 Simulation of transportation systems 13

2 SUMO architecture and routing modules **16**

2.1 SUMO STRUCTURE 17
2.2 Actors Identifications 18
2.3 SUMO's architectural components 19
2.4 General use case diagram 20
2.5 Use case : generate an abstract road network 23
2.6 Use case : import and convert a road networks 24
2.7 Use case : create and modify SUMO networks 27
2.8 Use case : convert of O/D matrices to vehicle trips 29
2.9 Use case : compute and build vehicle routes from induction loop conts 31
2.10 Use case : compute and build vehicle routes from demand definition using turning percentages at junctions 32
2.11 Use case : generate a set of random trips for a given network 33
2.12 Use case : Simulate a defined scenario without/with a graphical user interface . 34
2.13 Use case : compute and build vehicle routes from demand definition using shortest path computation 38
2.13.1 Route planning primitive and models 38
2.13.2 Static User Assignment 40
2.13.3 Dynamic User Assignment (Iterative Routing) 44
2.13.4 Dynamic User Assignment (Automatic Routing) 45

3 Description of the Eco-routing model **48**

3.1 Fuel and CO_2 emission model 49
3.2 Impacts of acceleration 53
3.3 Eco-routing model description 54
3.3.1 Network modeling and formulation 54
3.3.2 Eco-routing problem 55
3.3.3 Eco-routing model(solution) formulation 55
3.4 Numerical experiments for Eco-routing model 57
3.4.1 First experiment 57
3.4.2 Second experiment 58

4 Implementation of the Eco-routing model in SUMO 61
4.1 Requirement analysis and specification 62
4.1.1 Actors Identifications 62
4.1.2 Functional requirements 62
4.1.3 Non-Functional requirements 63
4.1.4 duarouter's use case diagram with our extended eco-routing model . . . 64
4.1.5 Use case : Perform routing with eco-routing, distance and travel time models 65
4.2 Conceptual design 72
4.2.1 Static modeling: class diagrams 72
4.2.2 dynamic modeling: sequence diagrams 74
4.2.3 Sequence diagram for " Load the road network" 74
4.2.4 Sequence diagram for "Load trips" 75
4.2.5 Sequence diagram for "Activation of routing algorithm" 77
4.2.6 Sequence diagram for "Routing with Ecological model" 78
4.3 Realisation 80
4.3.1 The Integrated objectives functions and parameters 80
4.3.2 Hardware environment 83
4.3.3 Software environment 83
4.3.4 Technologies used 85

5 Simulations and results 87
5.1 Simulation scenario 88
5.2 Evaluation 98

List of Figures

1 Summary of route planning metrics. 5
2 Overview of ITS based Communication. 7
3 V2V Communication Scenario [6]. 8
4 V2I Communication Scenario [6]. 8
5 Traffic assignment components. 11

6 SUMO's architectural components diagram. 20
7 General use case diagram. 22
8 Description of netgenerate application. 23
9 Road network-(Cities-map.net.xml). 24
10 Description of netconvert application. 27
11 Description of netedit application. 28
12 netedit user interface. 28
13 Description of od2trips application. 30
14 Description of jtrrouter application. 32
15 Description of dfrouter application. 33
16 Description of sumo/sumo-gui applications. 37
17 Sumo-gui user interface. 37
18 Description of Duarouter application. 41
19 duarouter's dependency components diagram. 42
20 duarouter's use case diagram. 44

21 Network topology and movement properties of the grid network[28]. 58

22 Extented duarouter's use case diagram. 65
23 Perform routing with eco-routing, distance and travel time models use case diagram. 66
24 Extented duarouter's class diagram. 74
25 Sequence diagram of "Load the road network". 75
26 Sequence diagram of "Load trips". 76
27 Sequence diagram of "Activation of routing algorithm". 77
28 Sequence diagram of "Routing with Ecological model". 79

29 Our OSM grand tunis street map boundries and views. 91
30 Our road network and additional information opened in netedit. 91
31 Sumo-gui with our scenario of simulation. 93

32 Flowchart of our road network and trips generation process. 94
33 Flowchart of our routes generation process. 95
34 Flowchart of our simulation with travel time routing process. 96
35 Flowchart of our simulation with distance routing process. 97
36 Flowchart of our simulation with eco-routing process. 98

List of Tables

1 GENERAL USE CASE DESCRIPTION. 21
2 Examples of netgenerate outputs 24
3 duarouter's use case description. 43
4 Description of parameters used in calculating fuel emission rates [28]. 52
5 Description of parameters used in calculating CO_2 emissions [28]. 53
6 Total costs of all paths (all in monetary value except for 'Emission')[28]. [1] Time cost in Case 1, [2] Fuel cost in Case 1 60
7 Use case Description of "Activate a routing model". 67
8 Use case Description of "Activate a routing algorithm". 67
9 Use case Description of "Command to load all routes in one step or go through them stepwise". 68
10 Use case Description of "Add file as SUMO-network format to route on". . . . 69
11 Use case Description of "Add file as SUMO-network format to route on". . . . 69
12 Use case Description of "Activate verbose's additional information output". . . 70
13 Use case Description of "Command to write all log messages into file". 70
14 Use case Description of "Add the begin time". 71
15 Use case Description of "Add the end time". 71
16 Use case Description of "Ask for help". 72
17 Extented duarouter's class diagram description 73
18 Description of parameters used in movement cost [28]. 81
19 Description of parameters used in fuel consumption [28]. 82
20 Description of software 85
21 Our used passenger type car's characteristics 92
22 SUMO routing models and the average cost metric values for all simulated trips 100

Listings

2.1 node file (Cities.nod.xml) 25
2.2 edge file (Roads.edg.xml) 25
2.3 type file (Roads-type.type.xml) 26
2.4 od2trips application command input 30
2.5 od2trips application trips definitions file output 30
2.6 traffic assignment zone (TAZ) file input 30
2.7 OD-matrice file input 31
2.8 A input detector file (detector1.xml) 33
2.9 Route file (My-routes.rou.xml) 33
2.10 Sumo/sumo-gui Configuration file (My-simulation.sumocfg) 36
2.11 Command line of Sumo simulation 36
2.12 Simulation output (My-summary.xml) 37
2.13 Command line to generate routes using duarouter 40
2.14 Command line to perform dynamic user equilibrium using duaIterate.py 45
2.15 Command line to perform automatic routing using one-shot.py 47
5.1 Sumo/sumo-gui Configuration file: "simulation-time-routing-based.sumo.cfg.xml" 96
5.2 Sumo/sumo-gui Configuration file: "simulation-distance-routing-based.sumo.cfg.xml" 97
5.3 Sumo/sumo-gui Configuration file: simulation-eco-routing-based.sumo.cfg.xml 98

General Introduction

Eco-routing is a vehicle navigation method that chooses those paths that minimize fuel consumption, energy consumption or pollutant emissions for a trip to given destination. Road transportation has numerous adverse effects on the environment. It is the largest contributor to global warming through emissions of CO_2 [1] and is responsible for the deterioration of air quality in areas with dense road networks through emissions of particulate matter (PM) and other pollutants. For example, the particulate matter (PM) is a dangerous pollutant: the study by Raaschou Nielsen, 2013 [2] revealed that for every increase of 10 $\mu g/m^3$ of particulate matter in the air, the lung cancer rate in the area rose by 22%. Apart of this environmental dimension, there is also an economical side to the problem. The monetary cost associated with energy (fuel) consumption is considerable. This motivates techniques such as eco-routing that aim to lower pollutant emissions, or energy (fuel) consumption, or both.

Different authors consider different variants of eco-routing. Some aim to lower pollutant emissions while others aim to lower consumption. In the former case the term "eco-routing" stands for “ecologic routing” while in the latter case it might be both "ecologic" and"economic" (Due that CO_2 emissions are reduced together with consumption).

Although renewable sources of transportation fuels(substituting the fossil fuel with cleaner energy sources) and alternative fuel vehicles(improving the technology of vehicles) may be a long-term solution, reducing the emission of the existing transportation fleet is an important and effective measure in the near term, and likely remains to be so for years to come.

In the last two decades , and because of increasingly severe environmental issues, the environmental and economical impacts of the transportation sector have attracted the attention of scholarly communities, who devoted much research effort towards sustainable mobility, to save fuel consumption and emissions.

In this context, this present project aims to implement and integrate an eco-routing techniques in the SUMO (Simulation for Urban MObility).

Simulation of Urban MObility, (SUMO) [3], is an open source, microscopic, multi modal traffic simulator. It allows the user to simulate how a specified traffic demand performs on a given road network. It is microscopic, which means each vehicle is modelled explicitly, it has its own route and moves individually through the network. The German Aerospace Center began developing SUMO in 2001 and since then it has been improved and has evolved into a suite of traffic modelling utilities which includes a road network capable of reading different source formats, demand generation and routing utilities. SUMO was developed as an open source simulator, it is licensed under the Eclipse Public License” [3] , aiming that its prospective users will suggest and implement improvements to the simulator helping to build a better and more realistic model

This document, composed of six chapters, summarizes the work carried out during this project.

The first chapter, entitled "Context of the project" presents and puts the subject in its context. Then, we will draw up a state of the art of the Traffic simulation.

The second chapter "SUMO architecture and routing modules" presents the architecture and the different modules of SUMO Simulation for Urban MObility [3], in particular, those which manage and control the choice of vehicle paths.

The third chapter "Description of the Eco-routing model" describes the eco-routing model to be implemented, in particular the parameters relating to this model and the algorithms and state changes, which govern their operation.

The fourth chapter "Implementation of the Eco-routing model in SUMO", deals with the implementation of the proposed model in a modular way respecting the architecture of the simulator and making it easier for the user to activate the desired model.

the fifth and last chapter "Simulations and results" validates our implementation through simulation of scenario using real maps and the generation of evaluation metrics including average time, average distance and estimated average consumption for the simulated trips.

1. Context of the project

Contents

1.1 Presentation of the context **4**

1.2 Problematic **4**

1.3 Nowadays Routing Criteria **5**

1.4 Presentation of the project **5**

1.5 State of the art **6**

- 1.5.1 Transportation and environmental impact 6
- 1.5.2 ITS (INTELLIGENT TRANSPORTATION SYSTEMS) 6
- 1.5.3 Routing models 9
- 1.5.4 Eco-routing models 11
- 1.5.5 Simulation of transportation systems 13

Introduction

In this section, First we present and put the subject in its context and then we draw up a state of the art of the Traffic simulation or the simulation of transportation systems and the different research effort towards sustainable mobility, to reduce fuel consumption and emissions for a better ecosystem. In state of the art, at first, we draw attention about transportation and environmental impact. Then, we give a brief review about intelligent transport System. Next, we survey the routing and eco-routing models and techniques. Finally, we discuss the traffic simulation from importance to tools.

1.1 Presentation of the context

This project will be developed as part of the final project in order to obtain the national diploma in computer engineering from the private higher school of technology and engineering in Tunisia (TEK-UP). The internship is taking place over a period of six months within Riadi Laboratory-Tunisia [4]. It is supervised by Ms. Hajer TAKTAK, my university teacher, and by Mr. Amine KCHICHE my professional supervisor.

1.2 Problematic

Road transportation has numerous adverse effects on the environment. It is the largest contributor to global warming through emissions of CO_2 [1] and is responsible for the deterioration of air quality in areas with dense road networks through emissions of particulate matter and other pollutants. Although renewable sources of transportation fuels(substituting the fossil fuel with cleaner energy sources) and alternative fuel vehicles(improving the technology of vehicles) may be a long-term solution, reducing the emission of the existing transportation fleet is an important and effective measure in the near term, and likely remains to be so for years to come.

In the last two decades , and because of increasingly severe environmental issues, the environmental and economical impacts of the transportation sector have attracted the attention of scholarly communities, who devoted much research effort towards sustainable mobility, to save fuel consumption and emissions.

Eco-routing solutions will address precisely the above issues. It is a vehicle navigation method that chooses those paths that minimize fuel consumption, energy consumption or pollutant emissions for a trip to given destination.

1.3 Nowadays Routing Criteria

In present studies, we show that travel impedance for the transportation network are travel time, travel cost, travel distance. However, due to congestion and transportation environmental pollution, route choice not only depends on the shortest path but also it depends on ecological parameters of link of network. Therefore, planner should focus on the optimal path rather than the shortest path. Optimal paths may be defined as the paths having minimum travel time or cost along with higher safety and ecological factors respect.

Cost estimation of road segment is key to route planning. This estimation gives wide range of impact, including monetary costs, state of the traffic in the road network, safety and pollution and climate change. A summary of some of the routing metrics is provided in Figure 1.

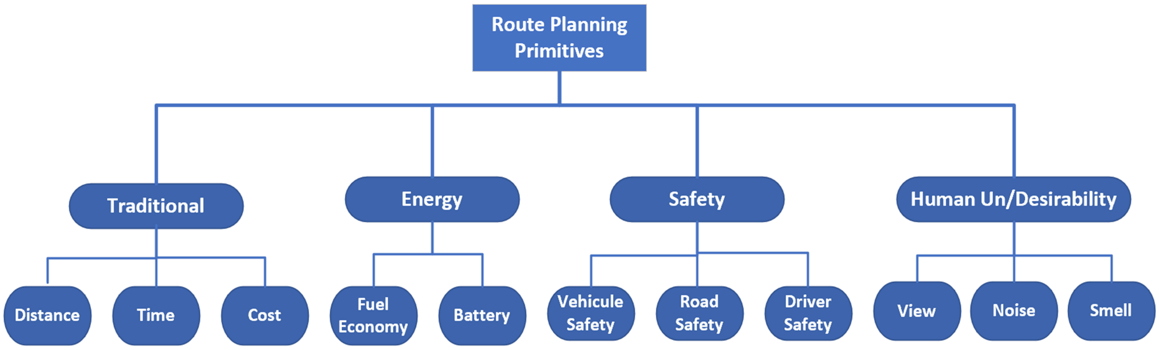


Figure 1. Summary of route planning metrics.

1.4 Presentation of the project

In this contexe, this project aims to implement an eco-routing techniques in the SUMO (Simulation for Urban MObility). In this document i am going to:

- Describe the architecture and the different modules of SUMO Simulation for Urban MObility, in particular, those which manage and control the choice of vehicle paths.
- Clearly describe the eco-routing model to be implemented, in particular the parameters relating to this model and the algorithms and state changes, which govern their operation.
- Implement the proposed model in a modular way respecting the architecture of the simulator and making it easier for the user to activate the desired model.
- Validate the implementation through simulation of scenarios using real maps and the generation of evaluation metrics including average time, average distance and estimated average consumption for the simulated trips.

1.5 State of the art

1.5.1 Transportation and environmental impact

Transportation plays an important role in climate change. Transportation consumes over 27.8% of all energy, of which 92.8% comes from petroleum use; Transportation also contributes roughly 32.7% of total greenhouse gas (GHG) emissions [5].

Although renewable sources of transportation fuels(substituting the fossil fuel with cleaner energy sources) and alternative fuel vehicles(improving the technology of vehicles) may be a long-term solution, reducing the emission of the existing transportation fleet is an important and effective measure in the near term, and likely remains to be so for years to come.

In the last two decades, and because of increasingly severe environmental issues, the environmental and economical impacts of the transportation sector have attracted the attention of scholarly communities, who devoted much research effort towards sustainable mobility, to save fuel consumption and emissions.

1.5.2 ITS (INTELLIGENT TRANSPORTATION SYSTEMS)

Intelligent transport System

Mobility has had an increasing impact on worldwide economical and social development over the last decades. current road traffic yields in high congestion and pollution and both will increase with the predicted increase of road traffic amount. On the other hand, the applications of intelligent transportation systems are showing promising results for fuel saving and emission reduction.

Among them, eco-routing navigation systems are one of such successful applications, which enables users to reduce their fuel consumption and CO_2 emissions at the route choice level. Intelligent transportation system (ITS) solution is a form of improvement in information and communication technology (ICT), it can control several indicators, such as speed, traffic signals, and route guidance to minimize the negative impact on the environment [6].

Figure 2 shows the overview of ITS based data sharing.

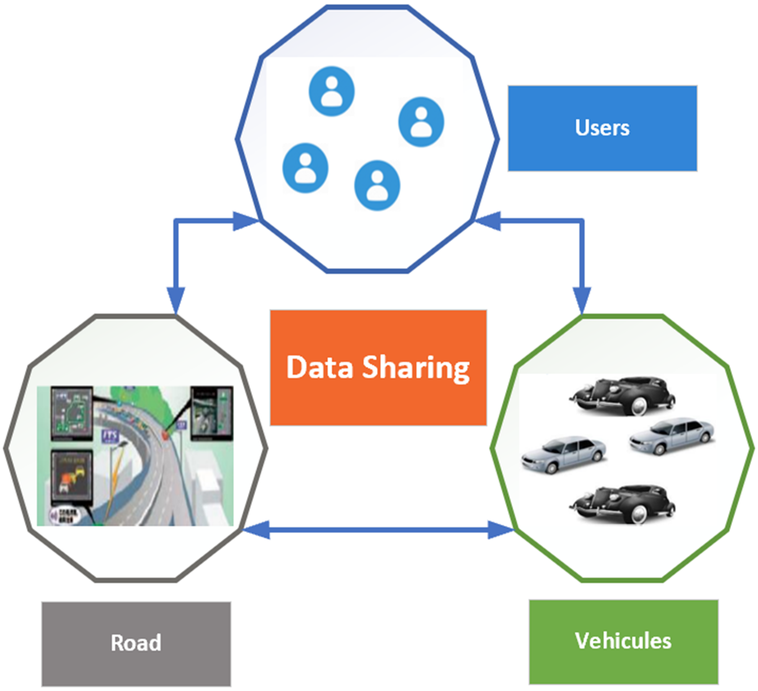


Figure 2. Overview of ITS based Communication.

ITS Advantages

Among the advantages of the applications of intelligent transportation systems we have:

- Quick response to incidents like road accidents/medical emergencies/rescue
- Real-time vehicle location monitoring
- Safety/Traffic Jam alerts for travellers
- Real-time data transmission support
- On demand responsive application support
- Data sharing support for multiple geographical regions
- Real-time vehicle deficiency detection

ITS deployment and communication strategies

ITS can be deployed under the constraints of Urban or Rural topologies. ITS supports the following communication types: Vehicle-to-Vehicle (V2V): In this kind of correspondence, vehicles can specifically connect with one another, without the help of the external network. The following Figure 3 shows the V2V communication scenario. Due to high mobility environment and short radio range, this type of communication suffers from the following issues:

- **Message Flooding:** Large scale vehicle density can flash the messages over the network and these may be data messages or control messages.
- **Frequent Link Breaks:** Vehicles can move in/out from the network at any stage which may cause interruption in already established paths.

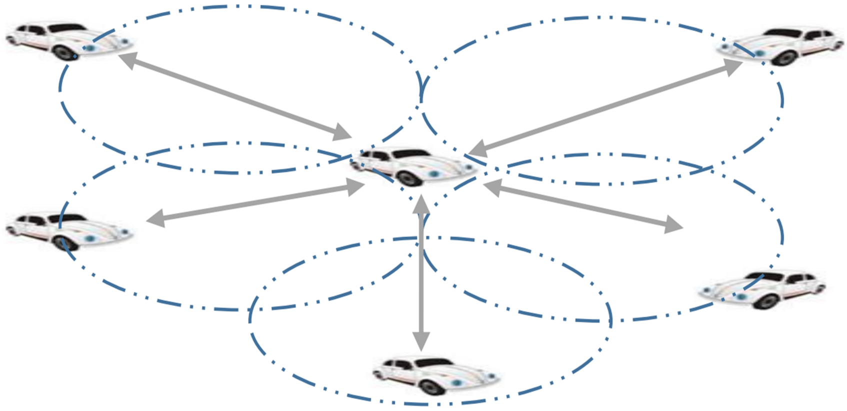


Figure 3. V2V Communication Scenario [6].

Vehicle-to-Infrastructure (V2I): In this type of communication, vehicles can interact with road side units (RSU) to share the information with other vehicles as shown in Figure 4. This type of communication suffers from the compatibility issues of homogeneous networks.

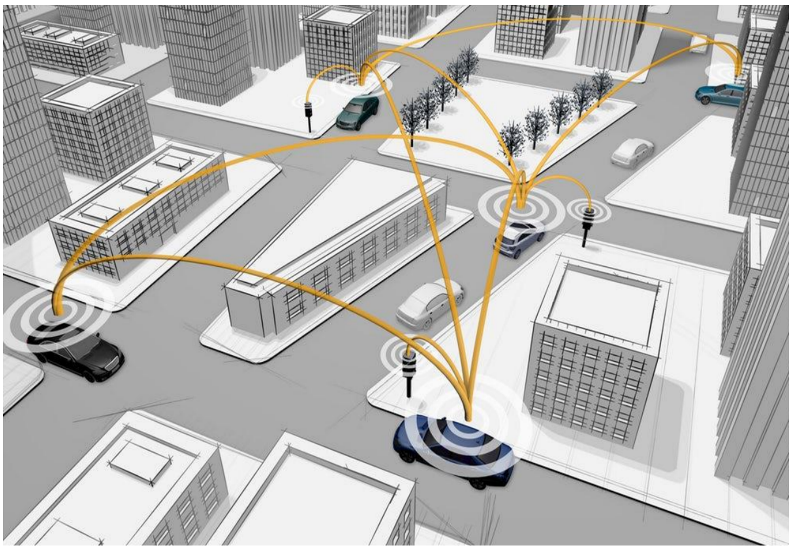


Figure 4. V2I Communication Scenario [6].

1.5.3 Routing models

Routing is guiding vehicles to their destinations based on a single criterion, such as travel time, distance, emissions, and fuel, or any combination of them.

The concept of vehicle routing has been used by public road agencies for decades. The main tool used by the agencies is the roadside variable message signs (VMSs). Nevertheless, with the ICT advancements, route guidance services by the private parties have ballooned due to the commercial introduction and affordability of the standalone personal navigation devices and smartphones [7].

More recently, environmental variables, like CO_2 emissions and fuel consumption/type, were taken into account, and eco-routing was introduced to replace the conventional routing concept that only aimed to minimize travel time. Other synonym terms of eco-routing can be pollution routing[8] or green routing [9].

Traffic assignment is the transportation process that focuses mainly on the choice of the path from an origin to a destination. The route choice is based on the objectives set by drivers or by governments [10].

Traffic assignment models are classified into two major categories: the static (STA) and the dynamic traffic assignment (DTA).

- The STA models do not represent the congestion phenomenon and consider equal in- and outflow from a link. The main outputs of STA models are average speed, traffic volume, traffic composition, and to test the level of service that are used to estimate the weights of traffic characteristics to define the routes [11].
- On the other hand, DTA models are represents the real situation more efficiently in which it considers the changes in the traffic flow with time. DTA is an iterative process that examines the progress of achieving either user equilibrium (UE) or system optimal (SO) assignment [12].
- Achieving UE or SO is associated with weights of the different traffic characteristics taken into account (travel time, distance, emissions, and fuel) for every link. As presented in 5, both STA and DTA consist of two components which are traffic flow model and travel choice principle. Traffic flow models can take three forms: microscopic, mesoscopic, and macroscopic [13]. The microscopic model considers the detailed temporal characteristics of every vehicular agent in the network. It requires models that account for the behavioural aspect of the drivers. The outputs can include the position, speed, and acceleration of

every vehicle at each time step. The mesoscopic model goes between the microscopic and macroscopic flow models. It represents the vehicle flow in aggregate terms, and the behaviour rules are captured in detail. Finally, the macroscopic requires aggregated information about the vehicular dynamics. The main drawback here is that it cannot capture reality and certain traffic incidents, such as queues.

- Wardrop formulated two basic travel choice principles for traffic assignment[14] :
  - The first principle is the user equilibrium (UE): "The journey cost on all the routes actually used are equal, and less than those which would be experienced by a single vehicle on any unused route",
  - while the other being the system optimal (SO): "The average journey cost is a minimum for all routes in a network". Compared to an assignment based on this principle, a lower average travel cost cannot be achieved by any other assignment. SO assignment is what authorities seek to achieve as the total travel cost in a network is minimized. This type of traffic assignment means that some users may experience a higher travel cost to achieve SO assignment. In other words, it may be unfair to certain users when the global benefit is achieved.

Achieving UE or SO is associated with weights of the different traffic characteristics taken into account (travel time, distance, emissions, and fuel) for every link. As presented in 5, both STA and DTA consist of two components which are traffic flow model and travel choice principle. Traffic flow models can take three forms: microscopic, mesoscopic, and macroscopic [13].

- The microscopic model considers the detailed temporal characteristics of every vehicular agent in the network. It requires models that account for the behavioural aspect of the drivers. The outputs can include the position, speed, and acceleration of every vehicle at each time step.
- The mesoscopic model goes between the microscopic and macroscopic flow models. It represents the vehicle flow in aggregate terms, and the behaviour rules are captured in detail.
- Finally, the macroscopic requires aggregated information about the vehicular dynamics. The main drawback here is that it cannot capture reality and certain traffic incidents, such as queues.

Wardrop formulated two basic travel choice principles for traffic assignment[14] :

- The first principle is the user equilibrium (UE): "The journey cost on all the routes actually used are equal, and less than those which would be experienced by a single vehicle on any unused route”,

- while the other being the system optimal (SO): “The average journey cost is a minimum for all routes in a network”. Compared to an assignment based on this principle, a lower average travel cost cannot be achieved by any other assignment. SO assignment is what authorities seek to achieve as the total travel cost in a network is minimized. This type of traffic assignment means that some users may experience a higher travel cost to achieve SO assignment. In other words, it may be unfair to certain users when the global benefit is achieved.

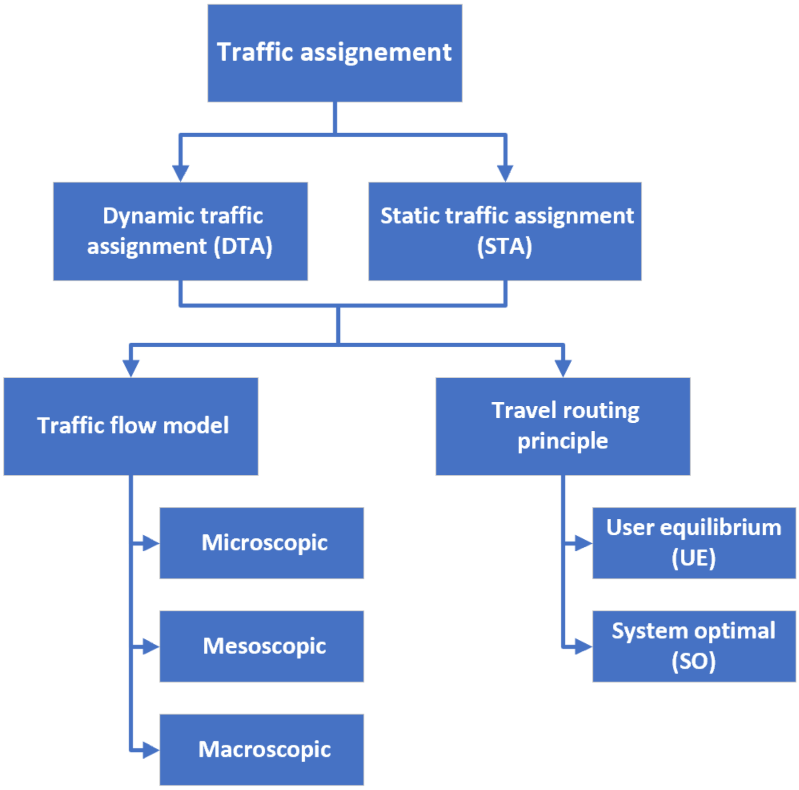


Figure 5. Traffic assignment components.

1.5.4 Eco-routing models

There are diverse publications on Eco-routing. We can classify them into three categories:

- Case studies publication that let Some authors to analyse and report their results and to recommend a solution.
- Method publication, here, some authors propose a method and evaluate it in a simulation.
- A publication with a fully described Eco-routing systems deployed in real-life applications.

In this present work, we have selected a well cited method publication that have not implemented yet in any transport simulator. we have reviewed and resume a lot of other valuable Eco-routing study's publication. The reviews below follow a similar structure, and it is the same structure that we have conducted in our project: first, the aim of the method is introduced, then the consumption model followed by the routing method, and at the end, results are discussed (if the method was evaluated).

Different authors evaluate their results in different ways. Many evaluate them in terms of savings, where the consumption (or pollutant emissions) on the eco-route is compared to the consumption or pollutant emissions on some reference path. The most common reference paths are the fastest paths and the shortest paths:

- The first eco-routing method described in the literature is proposed by Barth et al., 2007 [15] "Environmentally friendly navigation".
  The authors use CMEM microscopic consumption and pollutant emissions model, as in our selected method case, on a set of pre-recorded trips to estimate pollutant emissions and fuel consumption on individual roads in the network. This is used to build a regression model for pollutant emissions and fuel consumption where average vehicle speed and road grade are explanatory variables. The road grade parameters are assumed known for every road in the road network. The average vehicle speed parameter can be based on historical observations or real-time data.
  The routing was done with the Dijkstra's algorithm.
  The authors show that considerable savings in energy and emissions can be obtained if the right path is taken in the right situation and conclude that congestion can cause the change of the minimal energy path.
- De Nunzio et al., 2016 [16] proposed an eco-routing method for electric vehicles. The method uses Bellman-Ford algorithm instead of the Dijkstra's algorithm for routing. While Dijkstra's algorithm is asymptotically faster, it also assumes the routing costs are non-negative. This is a problem for electric vehicles since they can recuperate energy, which can result in negative consumption (meaning negative cost).
  It assumes the vehicle speed is constant and it also considers the changes in speed between two roads (or road segments) with different average speeds. When the vehicle passes on a boundary between two roads with different average speeds the transition is modeled with constant acceleration from original speed to the new speed. The authors compute the energy associated with the change in speed using the same framework like in our selected method case.
  Another novelty of this method is what authors call "adjoint graph". The authors conjecture that the usually used model for the road network is a directed graph where nodes represent intersections and edges represent roads between them. They propose to use the adjoint

graph, where nodes represent roads and edges represent the connections on intersections. The authors have evaluated the validity of eco-route model. The authors have chosen a thousand origin-destination pairs at random and plotted the eco-routes, the shortest paths, and the fastest paths. The results show 6% average savings when compared to the shortest path and 10% average savings when compared to the fastest path.

- Yao and Song, 2013[17]. The authors propose a pollutant emission and consumption model based on vehicle-specific power (VSP; Jimenez-Palacios, 1998[18]).
  They consider an empirical formula with fuel consumption and pollutant emission factors in grams per kilometer as a function of average speed.
  The routing was implemented with a time-dependent variant of the Dijkstra's algorithm.
  The authors conducted a proof-of-concept case study with a field trial between a single origin and destination. They observed 5.3% fuel savings on a fuel-minimizing eco-route and 5.9% CO2 emission reduction on a CO2-minimizing eco-route when compared to the fastest path.
  They also simulated eco-routing between 200.000 randomly generated origin-destination pairs in Beijing area. They computed the eco-routes for all and estimated that their method cuts carbon dioxide emissions by 4.7% and lower consumption by 4% in heavy traffic when compared to the fastest paths.

1.5.5 Simulation of transportation systems

Traffic simulation or the simulation of transportation systems is the mathematical modeling of transportation systems (e.g., freeway junctions, arterial routes, roundabouts, downtown grid systems, etc.) through the application of computer software to better help plan, design, and operate transportation systems.

Various national and local transportation agencies, academic institutions and consulting firms use simulation to aid in their management of transportation networks.

Simulation in transportation is important because it can study models too complicated for analytical or numerical treatment, can be used for experimental studies, can study detailed relations that might be lost in analytical or numerical treatment and can produce attractive visual demonstrations of present and future scenarios.

Software of transportation simulation can be classified into microscopic, mesoscopic, macroscopic:

- microscopic simulators track individual vehicle movements on a second or subsecond basis. Each road is divided into cells with the speeds of vehicle movement being quantified and dependent on the cell granularity.

- Macroscopic simulators use equations derived from fluid dynamics to calculate the flow of vehicles.
- Mesoscopic simulators, similar to microscopic simulators, simulate individual vehicles but in a simplified manner, without separation gap or lane-changing behaviour.

In Transportation Simulation we have Software like:

- Aimsun Live (Commercial),
- TSIS-CORSIM, CORridor SIMulation (Commercial)
- Paramics, PARAllel MICroscopic Simulation (Commercial),
- PTV Vissim, (Commercial),
- Synchro + SimTraffic, (Commercial)
- MATSim, Multi-Agent Transport Simulation , (Open Source)
- Transims, TRansportation ANalysis SIMulation System (NASA Open Source)
- SUMO Simulation of Urban Mobility, (DLR, Open Source)

The most of Transportation Simulation software are commercial software, as open source we have MATSim, Transims and SUMO. Among all those open source software, we have sumo that is more advanced up to now.

Simulation for Urban MObility (SUMO) is a mature and robust open source set of traffic simulation tools capable of importing road network layout data in various formats, modelling emissions, noise, driver behaviour, intermodal transportation and vehicle models validation [19].

One of most interesting features of SUMO is the capability to simulate V2V and V2I using Traffic and Network Simulation Environment (TraNS) to simulate the network communication. iTETRIS is an extension to the SUMO-TraNS coupling featuring cooperative traffic management [19].

In our work, we are going to implement an eco-routing techniques in this mature and robust Open Source SUMO (Simulation of Urban MObility) in a modular way and respecting its architecture.

Conclusion

This chapter presented and putted the subject in its context, then it surveyed a state of the art of the Traffic simulation. It drew attention about transportation and environmental impact. The

routing and eco-routing models and techniques was widely discussed in this chapter since this step, it is the basic step for traffic planning solutions.
In the next chapter, we will present the architecture and the different modules of SUMO Simulation for Urban MObility [3], in particular, those which manage and control the choice of vehicle paths.

2. SUMO architecture and routing modules

Contents

2.1 SUMO STRUCTURE **17**

2.2 Actors Identifications **18**

2.3 SUMO's architectural components **19**

2.4 General use case diagram **20**

2.5 Use case : generate an abstract road network **23**

2.6 Use case : import and convert a road networks **24**

2.7 Use case : create and modify SUMO networks **27**

2.8 Use case : convert of O/D matrices to vehicle trips **29**

2.9 Use case : compute and build vehicle routes from induction loop conts . . . **31**

2.10 Use case : compute and build vehicle routes from demand definition using turning percentages at junctions **32**

2.11 Use case : generate a set of random trips for a given network **33**

2.12 Use case : Simulate a defined scenario without/with a graphical user interface **34**

2.13 Use case : compute and build vehicle routes from demand definition using shortest path computation **38**

2.13.1 Route planning primitive and models 38

2.13.2 Static User Assignment 40

2.13.3 Dynamic User Assignment (Iterative Routing) 44

2.13.4 Dynamic User Assignment (Automatic Routing) 45

Introduction

In this chapter, we will present the architecture and the different modules of SUMO Simulation for Urban MObility [3], in particular, those which manage and control the choice of vehicle paths. First, we will discuss the different SUMO's possible areas of interest. Second, we will identify the actors that are concerned by using the powerful SUMO traffic simulator package. Next, we will extract the general architectural components of SUMO package and the relations between them using the structure of SUMO with the doxygen documentation [20]. Then, We will tackle the most important cases use of SUMO traffic simulator package using flowcharts, a use case diagrams and an illustrative scenario example to describe the most important modules of SUMO and especially the routing one and also to highlight the importance of each module for SUMO's users.

2.1 SUMO STRUCTURE

SUMO (Simulation of Urban MObility) [3] is an open source traffic simulation package. It was created by the German Aerospace Center (DLR). The development of SUMO started back in 2001. SUMO is not a traffic simulator only, but a suite of applications that allow the user to create/import a road network and define its corresponding traffic demand. It uses ”netconvert” to import a network from Open Street Map or from other traffic simulators such as VISUM, MATsim or VISSIM. Once a road network is imported and converted to the appropriate format, traffic demand, and routes for each vehicle should be created. ”DUAROUTER” will compute our routes. These routing tool take the network and trips as arguments and produce a route file that contains the routing information for each vehicle defined in the network. We can use random trips (which is what has been used in this report), or we can manually create a demand using OD (Origin-Destination) matrices or even by supplying various parameters for the specified network. These parameters include the population and the land class usage definition, among others.

The researches which cite or use SUMO come from all over the world, SUMO's possible areas of interest include:

- Microscopic traffic simulation: It is used to estimate the movement of vehicles on roads, in the case that available information is limited or a large amount of factors are involved, such as intersections with traffic lights and accidents.
- E-Mobility: since 2009 e-mobility has become very much in vogue. The central idea is that the electric car will contribute to sustainable transport development.

- Mobility Modelling: It is one of the important aspects in Vehicular Network. It can be commonly classified into the following categories: Macroscopic models, Mesoscopic models and Microscopic models.
- Autonomous Driving: It is a multi-agent setting where users do not have to drive at all. So, the host vehicle must apply sophisticated negotiation skills with other road users when overtaking, giving way, merging, taking left and right turns and while pushing ahead in unstructured urban roadways
- Intermodal Transport: The conceptual center of intermodal traffic is the individual person. This person needs to undertake a series of trips where each may be taken with a different mode of transport such as personal car, public bus or walking.
- Logistics Simulation: It's an interdisciplinary research field, covering aspects of operations research, mathematics, statistics, computer science, and engineering. Most of these disciplines consider to some extent questions of verification and validation of their applications, techniques, and models. It's implemented in sumo by containers and container stops.
- Traffic Management Solutions: they are increasingly called for to address problems of transport and mobility. In particular, introducing and enforcing variable speed limits, installing local-express lanes and coordinated traffic lights, imposing differentiated road pricing or optimizing traffic signal timing.
- Open Tools and Open Data: The use of open tools (such as SUMO and Python) and open data for transport modeling in general has attracted considerable research interests in recent years.
- Vehicular Communication: The SUMO suite has a huge number of application with modeling traffic within research on V2X (vehicle-to-vehicle and vehicle-to-infrastructure) communication.

2.2 Actors Identifications

In this subsection, we will identify the actors that are concerned by using SUMO traffic simulator package. At first, SUMO is not a traffic simulator only, but a suite of applications that allow the user to create/import a road network and define its corresponding traffic demand.

Since 2002, SUMO is in use at many institutions. It's also of immense importance for researchers as well as practitioners in the field of transportation.

SUMO is a powerful simulator for all kind of users, for:

- researchers,

- developers,
- academic institutions,
- national and local transportation agencies,
- consulting firms,
- and enthusiasts.

SUMO was developed as an open source simulator aiming that its prospective users will suggest and implement improvements to the simulator helping to build a better and more realistic model.

2.3 SUMO's architectural components

In this subsection, we will extract the different components of SUMO's package and precise the relations between them using the structure of SUMO with the doxygen documentation [20].
For this purpose, we have used UML Component diagrams. UML Component diagrams are used in modeling the physical aspects of object-oriented systems that are used for visualizing, specifying, and documenting component-based systems and also for constructing executable systems through forward and reverse engineering. Component diagrams are essentially class diagrams that focus on a system's components that often used to model the static implementation view of a system.
The purpose of a component diagram is to show the relationship between different components in a system. The term "component" refers to a module of classes that represent independent systems or subsystems with the ability to interface with the rest of the system.
A component diagram in UML gives a bird's-eye view of the software system. Component diagrams can describe software systems that are implemented in any programming language or style.

Figure 6 is the component diagram describes the SUMO's architectural components and the relationships between them.

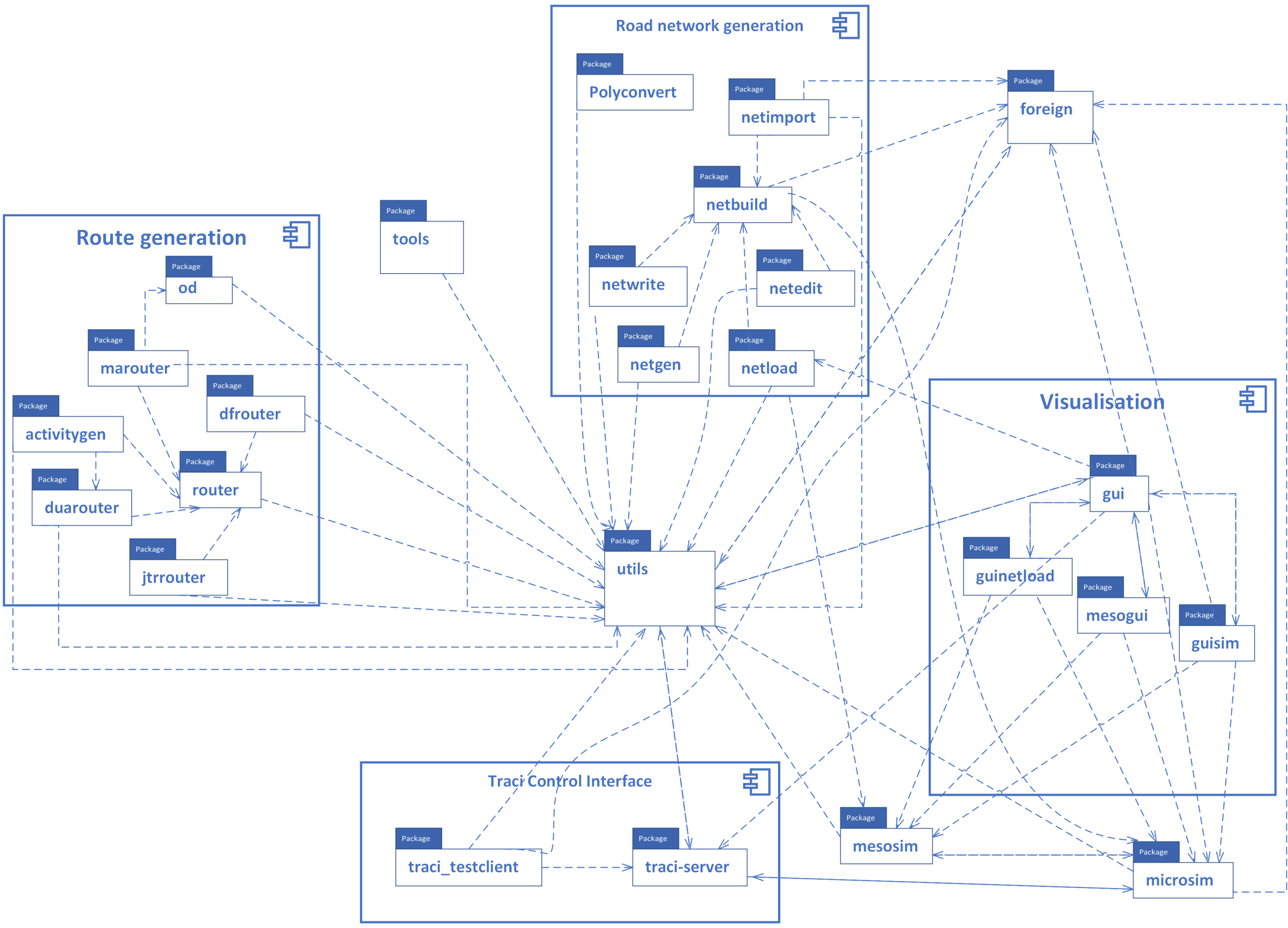


Figure 6. SUMO's architectural components diagram.

2.4 General use case diagram

In this subsection, we have extracted the different cases use of SUMO traffic simulator package, and the different functionalities that user can have while using this powerful package, see the Figure 7 the general use case diagram of SUMO traffic simulator package.
Then, we have given a brief description for each user's functionality and indicating the module that is responsible for achieving this functionality, see the Table 1 for the general use case descriptions.

Functionality	Description
generate an abstract road network	SUMO allows user to generate an abstract road network using netgenerate application.
import and convert a road networks	SUMO allows user to import and convert a road networks using netconvert application.
create and modify SUMO networks	SUMO allows user to create and modify SUMO networks using netedit application.
import geometrical shapes from different sources, convert them to a representation that may be visualized using sumo-gui.	SUMO allows user to import geometrical shapes (polygons or points of interest) from different sources, to convert them to a representation that may be visualized by sumo-gui using polyconvert application.
convert of O/D matrices to vehicle trips	SUMO allows user to convert of O/D matrices to vehicle trips using od2trips application.
generate a set of random trips for a given network	SUMO allows user to generate a set of random trips for a given network using randomTrips.py tool.
compute and build vehicle routes from demand definition using shortest path computation	SUMO allows user to compute and build vehicle routes from demand definition using shortest path computation.
compute and build vehicle routes from demand definition using turning percentages at junctions	SUMO allows user to compute and build vehicle routes from demand definition using turning percentages at junctions by jtrrouter application.
compute and build vehicle routes from induction loop conts and detector data	SUMO allows user to compute and build vehicle routes from induction loop conts and detector data using dfrouter application.
compute and build a macroscopic vehicle routes from various inputs (Origin-Destination-Matrices, trip files).	SUMO allows user to compute and build a macroscopic vehicle routes from various inputs (Origin-Destination-Matrices, trip files) using marouter application.
Simulate a defined scenario.	SUMO allows user to Simulate a defined scenario using sumo application.
Simulate a defined scenario with a graphical user interface.	SUMO allows user to Simulate a defined scenario with a graphical user interface using sumo-gui application.

Table 1. GENERAL USE CASE DESCRIPTION.

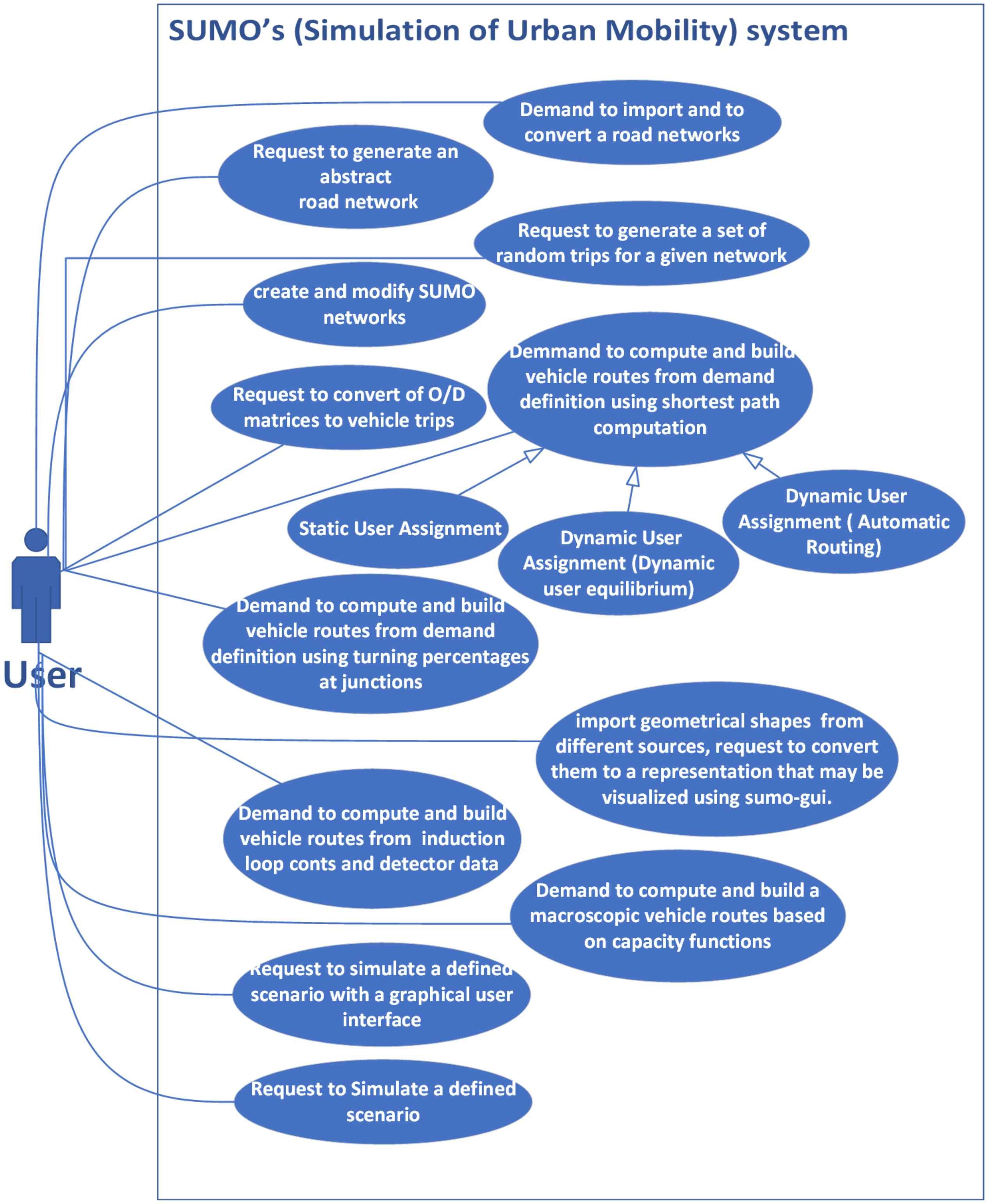


Figure 7. General use case diagram.

2.5 Use case : generate an abstract road network

With SUMO traffic simulator [3], road network can be created in three ways:

- Manually by creating our own node, edge, route, connection files.
- Using netgenerate application command.
- Importing road network from external sources such as OSM, VISSIM, VISUM etc.

In this subsection, we will focus on the second one, the generation of an abstract road network using netgenerate application command. The advantage of this solution that is so fast and we can generate a network and build a simulation scenario quickly. But this solution is not recommended, because the topology of road network and the parameters are generated randomly with no reality. Also we will touch, in the simulation and results chapter 5, the third solution for our simulation scenario to validate our implemented Eco-routing model. This solution lets us to perform a real scenario's context with a real road network map.
For those reason, In the next subsection 2.6, we will create manually the road network in our illustrative example scenario of simulation, to cover a more solutions in our report.

netgenerate application description

As they are described in the flowchart Figure 8. Netgenerate is a command line application. It requires a command line parameter to generate a different kind of road networks(Grid Network, Spider Network and Random Network), see Table 2.

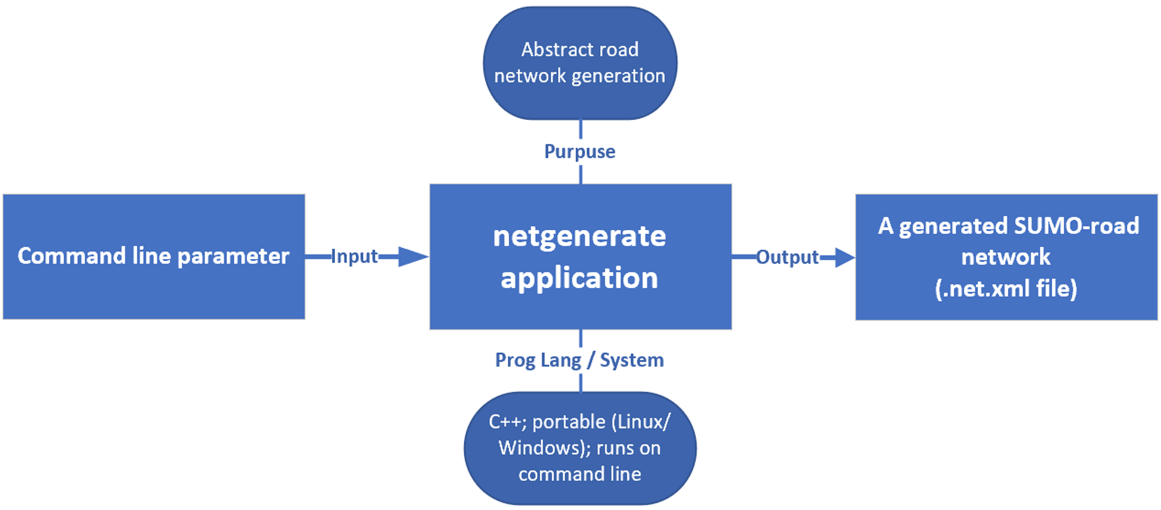


Figure 8. Description of netgenerate application.

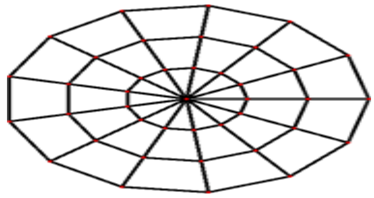
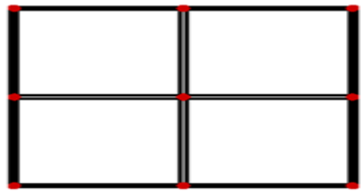
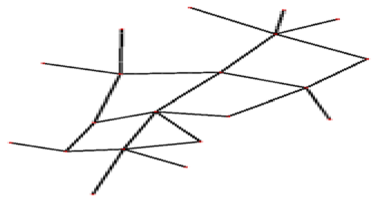
Input command line	netgenerate output
`netgenerate --grid --grid.x-number 3 --grid .y-number 3 --output-file my-grid-network. net.xml`	
`netgenerate --spider --spider.circle-number 3 --output-file my-spider-network.net.xml`	
`// --rand.iterations refers to number of nodes; default: 2000` `netgenerate --rand --rand.iterations 20 --output-file my-random-network.net.xml`	

Table 2. Examples of netgenerate outputs

2.6 Use case : import and convert a road networks

In this subsection we will start to build our example scenario. First we will create our manually road network. Figure 9 is the road network we want to build manually. The road network consists of nodes (junctions) and edges (i.e. roads that connect various junctions with each other). Each node has a coordinate x, y and we are going to use those coordinate in order to build our nodes.

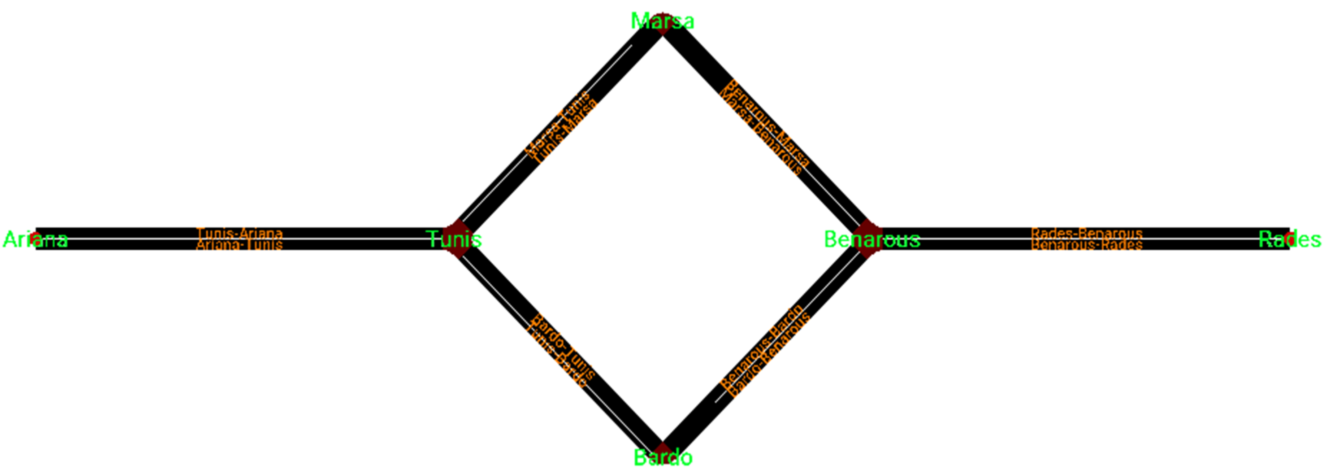


Figure 9. Road network-(Cities-map.net.xml).

The steps to create our network are :

- Step 1: Create our node file with the extension (.nod.xml)
  Listing 2.1 is the xml script code that we use to build our node file. Each node has an id(the cities name in our example). x and y represent the coordinate of each node or city.

```
        <nodes>
          <node id="Ariana" x = "-375" y="0"/>
          <node id="Tunis" x = "-125" y="0"/>
          <node id="Marsa" x = "0" y="125"/>
          <node id="Bardo" x = "0" y="-125"/>
          <node id="Benarous" x = "125" y="0"/>
          <node id="Rades" x = "375" y="0"/>
        </nodes>

```

Listing 2.1. node file (Cities.nod.xml)

- Step 2: Create the edge file with the extension (.edg.xml)
  Listing 2.2 is the xml script code that define the connect nodes/cities together to form links/roads. Each link has an id (the road name) and origine-destination nodes. We Should also put a type parameter to add more characteristics to each roads. The specification of this characteristics is our next step.

```
        <edges>
          <edge from="Ariana" to="Tunis" id="Ariana-Tunis" type="Type-
Ariana-Tunis"/>
          <edge from="Tunis" to="Ariana" id="Tunis-Ariana" type="Type-
Tunis-Ariana"/>
          <edge from="Tunis" to="Marsa" id="Tunis-Marsa" type="Type-
Tunis-Marsa"/>
          <edge from="Marsa" to="Tunis" id="Marsa-Tunis" type="Type-
Marsa-Tunis"/>
          <edge from="Tunis" to="Bardo" id="Tunis-Bardo" type="Type-
Tunis-Bardo"/>
          <edge from="Bardo" to="Tunis" id="Bardo-Tunis" type="Type-
Bardo-Tunis"/>
          <edge from="Marsa" to="Benarous" id="Marsa-Benarous" type="
Type-Marsa-Benarous"/>
          <edge from="Benarous" to="Marsa" id="Benarous-Marsa" type="
Type-Benarous-Marsa"/>
          <edge from="Bardo" to="Benarous" id="Bardo-Benarous" type="
Type-Bardo-Benarous"/>
          <edge from="Benarous" to="Bardo" id="Benarous-Bardo" type="
Type-Benarous-Bardo"/>
          <edge from="Benarous" to="Rades" id="Benarous-Rades" type="
```

```
Type-Benarous-Rades"/>
        <edge from="Rades" to="Benarous" id="Rades-Benarous" type="
Type-Rades-Benarous"/>
      </edges>

```

Listing 2.2. edge file (Roads.edg.xml)

- Step 3: Create the edge type file with the extension (.type.xml)

  Listing 2.3 is the xml script code that describes the road priority, the number of lanes, speed limit, type of vehicles allow, etc. the -1 value in priority parameter means that the priority in this edge/road is not defined.

```
      <types>
        <type id="Type-Ariana-Tunis" priority="-1" numLanes="2"
speed="120"/>
        <type id="Type-Tunis-Ariana" priority="-1" numLanes="2"
speed="120"/>
        <type id="Type-Tunis-Marsa" priority="-1" numLanes="2" speed
="120"/>
        <type id="Type-Marsa-Tunis" priority="-1" numLanes="2" speed
="120"/>
        <type id="Type-Tunis-Bardo" priority="-1" numLanes="2" speed
="120"/>
        <type id="Type-Bardo-Tunis" priority="-1" numLanes="2" speed
="120"/>
        <type id="Type-Marsa-Benarous" priority="-1" numLanes="2"
speed="120"/>
        <type id="Type-Benarous-Marsa" priority="-1" numLanes="2"
speed="120"/>
        <type id="Type-Bardo-Benarous" priority="-1" numLanes="2"
speed="120"/>
        <type id="Type-Benarous-Bardo" priority="-1" numLanes="2"
speed="120"/>
        <type id="Type-Benarous-Rades" priority="-1" numLanes="2"
speed="120"/>
        <type id="Type-Rades-Benarous" priority="-1" numLanes="2"
speed="120"/>
      </types>

```

Listing 2.3. type file (Roads-type.type.xml)

- Step 4: Generate the road network using netconvert application command.

netconvert application description

As they are described in the flowchart Figure 10. Netconvert is a command line application. It imports digital road networks from different sources and generates road networks that can be used by other tools from the SUMO package.
netconvert is able to import road networks from the following formats:

- "SUMO native" XML descriptions (.edg.xml, .nod.xml, .type.xml)
- OpenStreetMap (.osm.xml)
- Vissim
- MATsim
- etc.

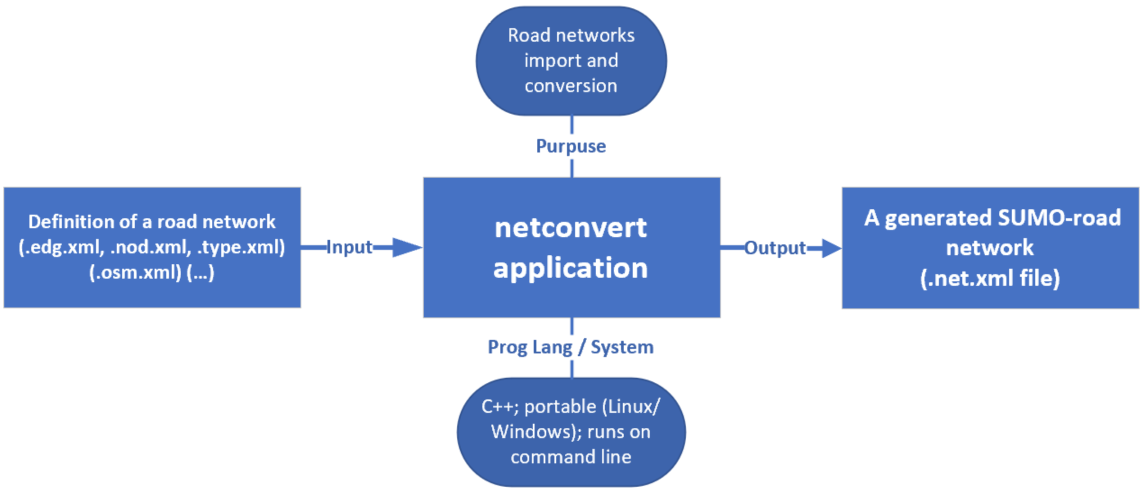


Figure 10. Description of netconvert application.

Now, netconvert is going to assemble our node, edge and type files, named respectively Cities.nod.xml, Roads.edg.xml and Roads-type.type.xml together, and it will generate our road network file with .net.xml extension and a complete name Cities-map.net.xml.
The bellow command is the command that we are going to use in order to generate our road network file:

```
netconvert --node-files Cities.nod.xml  --edge-files Roads.edg.xml
  --type-files Roads-type.type.xml  --output-file Cities-map.net.
xml
```

2.7 Use case : create and modify SUMO networks

In order to visualise our road network, we must use netedit application, the powerful graphical network editor for SUMO package:

netedit application description

As they are described in the flowchart Figure 11. netedit is a visual network editor, Figure 12 represents the netedit user interface. netedit can be used to create networks from scratch and to modify all aspects of existing networks. With a powerful selection and highlighting interface it can also be used to debug network attributes. It is built on top of netconvert. netedit has unlimited undo/redo capabilities and thus allows editing mistakes to be quickly corrected.

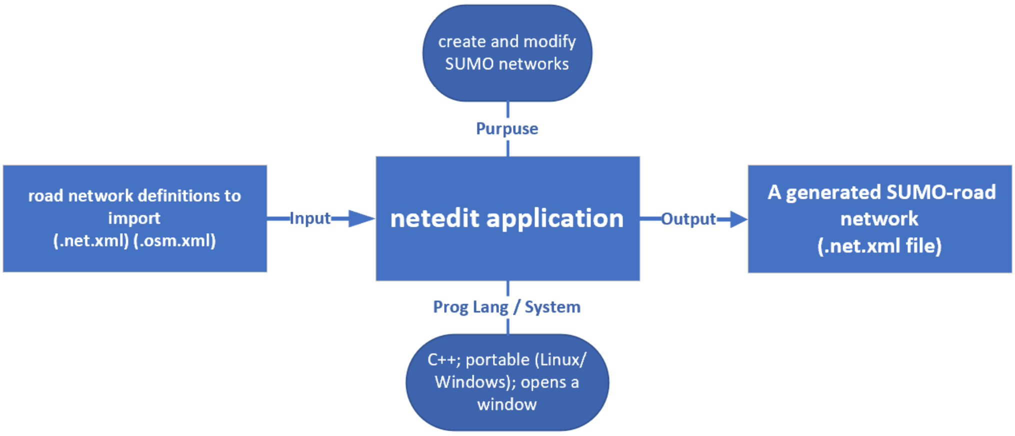


Figure 11. Description of netedit application.

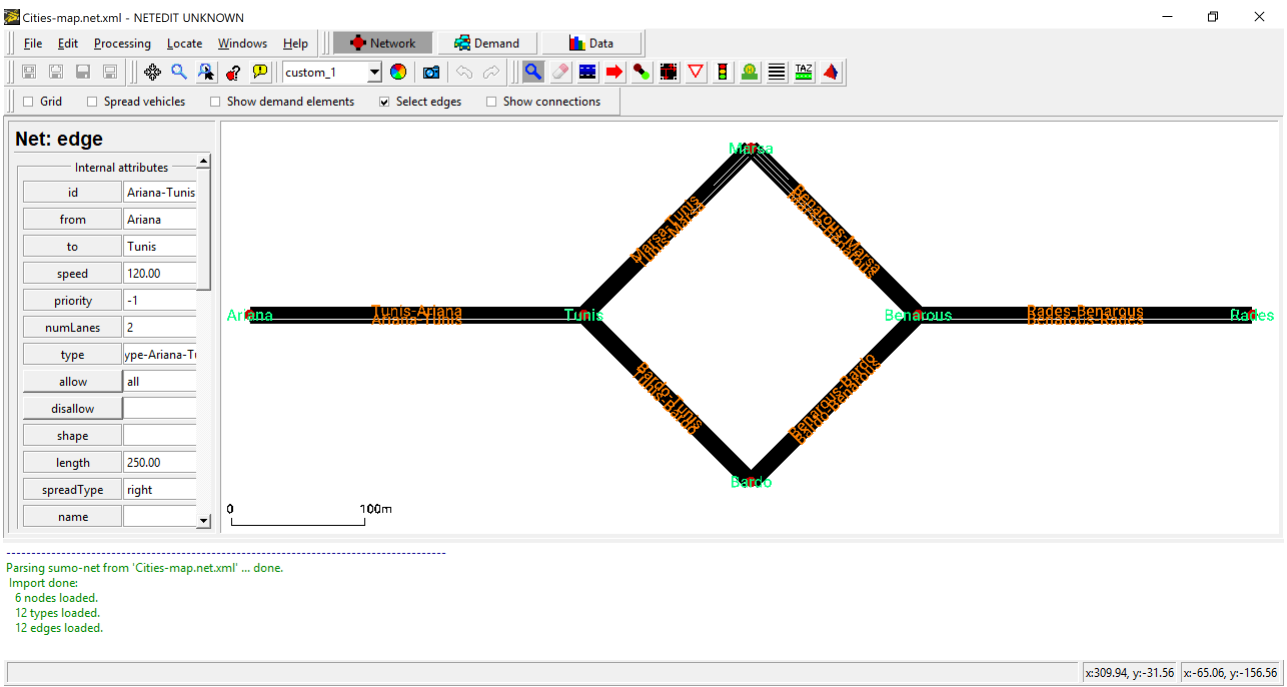


Figure 12. netedit user interface.

2.8 Use case : convert of O/D matrices to vehicle trips

After having generated a network, we need to generate a traffic demand. sumo and sumo-gui need routes as input for vehicle movements. For further illustration, a trip is a vehicle movement from one place to another defined by the starting edge(street), the destination edge, and the departure time. A route is an expanded trip, that means, that a route definition contains not only the first and the last edge, but all edges the vehicle will pass. There are several ways to generate routes for SUMO. The choice depends on our available input data:

- Using OD-matrice and trip definitions
- Using flow definitions and turning ratios
- Using detector data
- Using Randomization
- Manually

od2trips application description

Origin-Destination-Matrice (or OD-matrice) are often available from traffic authorities. They have to be converted to trips definitions using od2trips. After, we may use duarouter to turn our trips into routes, we will deeply focus on duarouter in our next Subsection 2.13.
O-D matrices describe traffic demand by dividing a given area into so-called "traffic assignment zones" (TAZs). For every TAZ at which traffic participants start (the demand origins), the number of participants that approach a destination TAZ is given. O-D matrix estimation methods based on traffic counts have been developed over the last 30 years [21].
As they are described in the flowchart Figure 13, od2trips application imports O/D-matrices and splits them into single vehicle trips, see Listing 2.6.
As they are indicated in the command line in Listing 2.4. od2trips requires a mapping of TAZ to edges file, see Listing 2.6. TAZ is described by its id (an arbitrary name) and lists of source and destination edges. Also od2trips requires a O-format OD-matrice, see Listing 2.7. The OD-matrice describe and precise the amount of trips(traffic) in the form FROM TO NUMVEHICLES.

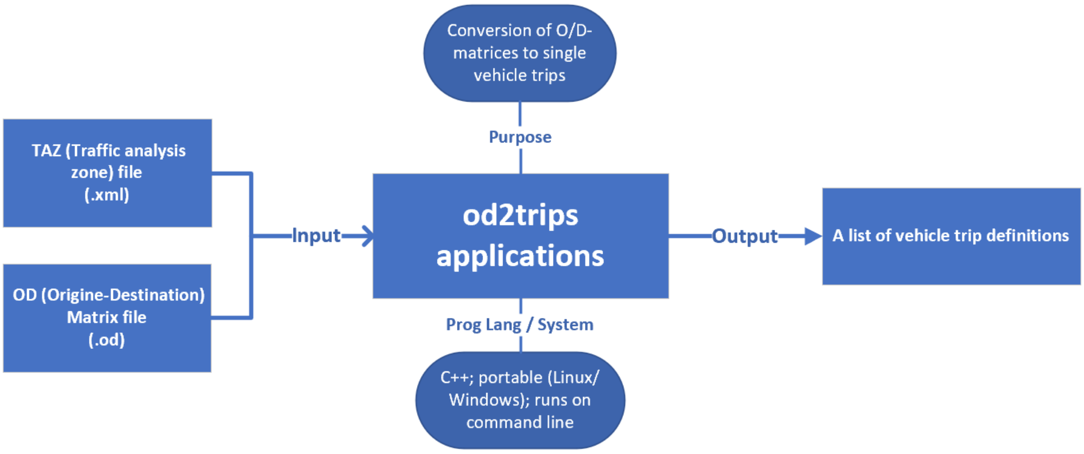


Figure 13. Description of od2trips application.

```
od2trips --taz-files TAZ_file.taz.xml --od-matrix-files OD_file.od.txt --output-file od_file.odtrips.xml
```

Listing 2.4. od2trips application command input

```
<routes>
    <trip id="5" depart="55.65" from="Ariana-Tunis" to="Benarous-Rades" fromTaz="1" toTaz="3"/>
    <trip id="1" depart="476.94" from="Ariana-Tunis" to="Tunis-Bardo" fromTaz="1" toTaz="2"/>
    <trip id="3" depart="654.29" from="Ariana-Tunis" to="Benarous-Rades" fromTaz="1" toTaz="3"/>
    <trip id="2" depart="1517.65" from="Ariana-Tunis" to="Benarous-Rades" fromTaz="1" toTaz="3"/>
    <trip id="0" depart="2488.08" from="Ariana-Tunis" to="Tunis-Bardo" fromTaz="1" toTaz="2"/>
    <trip id="4" depart="2950.16" from="Ariana-Tunis" to="Benarous-Rades" fromTaz="1" toTaz="3"/>
</routes>
```

Listing 2.5. od2trips application trips definitions file output

```
<additional>
    <tazs>
    <taz id="1" edges="Ariana-Tunis">
    </taz>
    <taz id="2" edges="Tunis-Bardo">
    </taz>
    <taz id="3" edges="Benarous-Rades">
```

```
        </taz>
        </tazs>
    </additional>

```

Listing 2.6. traffic assignment zone (TAZ) file input

```
    $OR
    * From-Time To-Time
    0.00 1.00
    * Factor
    1.00
    *
    1 3 4
    1 2 2

```

Listing 2.7. OD-matrice file input

2.9 Use case : compute and build vehicle routes from induction loop conts

jtrrouter application description

As they are described in the flowchart Figure 13. The jtrrouter is a routing applications which uses flows and turning percentages at junctions as input(routing based on turn-ratios). The following parameter must be supplied

- the network to route the vehicles through with file name extension (.net.xml),
- the description of the turning ratios for the junctions with file name extension (.turns.xml), and
- the descriptions of the flows with file name extension (.flows.xml).

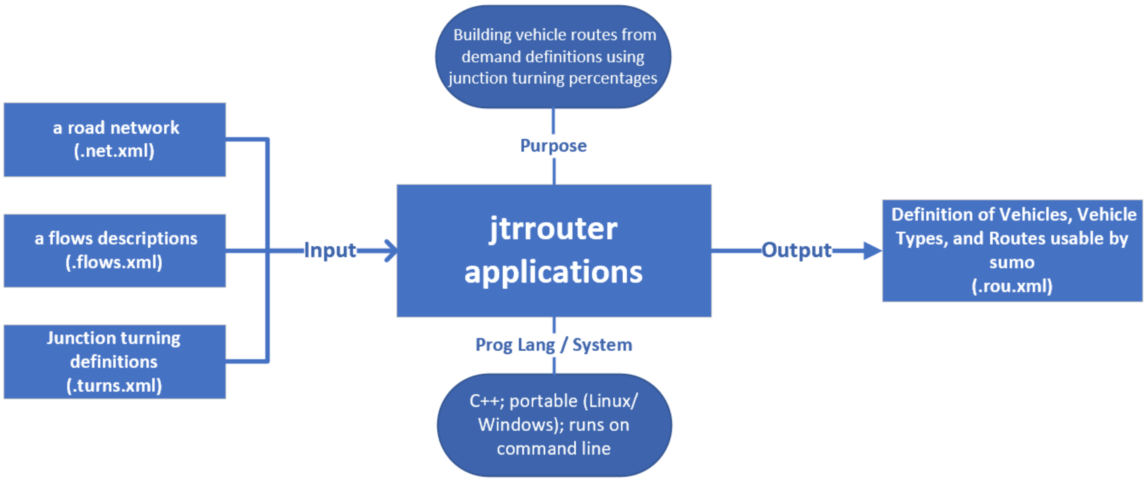


Figure 14. Description of jtrrouter application.

2.10 Use case : compute and build vehicle routes from demand definition using turning percentages at junctions

dfrouter application description

The idea behind this router is that nowadays, most highways are well equipped with induction loops, measuring each of the highways' entering and leaving flows. Given this information one may assume that the flows on the highway are completely known[21]. As they are described in the flowchart Figure 15. The dfrouter uses directly the information collected from induction loops to computer and rebuild the vehicle routes that may be used by sumo. the input detector file should look as Listing 2.8 with *pos* parameter refers to The position on the lane the detector shall be laid on in meters.

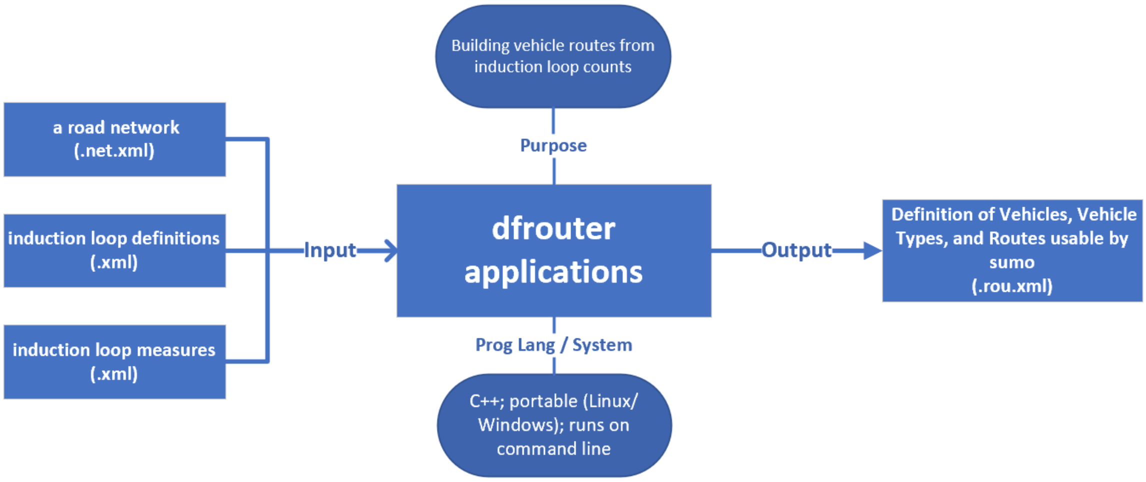


Figure 15. Description of dfrouter application.

```
<detectors>
    <detectorDefinition id="detector1" lane=":Ariana_0_0" pos="4"/>
... further detectors ...
</detectors>

```

Listing 2.8. A input detector file (detector1.xml)

2.11 Use case : generate a set of random trips for a given network

randomTrips.py SUMO tool description

"randomTrips.py" generates a set of random trips for a given network(option –net-file). It does so by choosing source and destination edge randomly. The resulting trips are stored in an XML file (option –output-file, default trips.trips.xml) and they are suitable for duarouter, see Listing 18. This trips are distributed evenly in an interval defined by begin (option -b, default 0) and end time (option -e, default 3600) in seconds. The number of trips is defined by the repetition rate (option -p, default 1) in seconds.

Listing 2.9 is the xml script code that presents our demand routes. In our illustrative example scenario we defined our three routes manually. Each vehicle must have a departure time, characterised by a specific type and affected to a specific route.

```
    <routes>
      <vType accel="1.0" decel="5.0" id="Car" length="4.0" maxSpeed="
100.0"/>
```

```
        <vType accel="1.0" decel="5.0" id="Bus" length="9.0" maxSpeed="80.0"/>

        <route id="route1" edges="Ariana-Tunis Tunis-Marsa Marsa-Benarous Benarous-Rades"/>
        <vehicle depart="0" id="Opel-Astra" route="route1" type="Car" />
        <vehicle depart="5" id="Urbanway-18M" route="route1" type="Bus" />

        <route id="route2" edges="Ariana-Tunis Tunis-Bardo Bardo-Benarous Benarous-Rades"/>
        <vehicle depart="40" id="Honda-Fit" route="route2" type="Car" />
        <vehicle depart="45" id="CapaCity" route="route2" type="Bus" />

        <route id="route3" edges="Ariana-Tunis Tunis-Bardo"/>
        <vehicle depart="80" id="Ford-Fiesta" route="route3" type="Car" />
        <vehicle depart="85" id="A-537" route="route3" type="Bus" />
    </routes>

```

Listing 2.9. Route file (My-routes.rou.xml)

2.12 Use case : Simulate a defined scenario without/with a graphical user interface

sumo/sumo-gui applications description

As they are indicated in the flowchart Figure . sumo and sumo-gui applications are the simulations themselves. sumo application is a microscopic, space-continuous, and time-discrete traffic flow simulation. sumo-gui is basically the same application as sumo, just extended by a graphical user interface, Figure 17 represents the sumo-gui user interface.
The Figure 16 is a flowchart to describe the functionality of sumo and sumo-gui applications, they need some inputs to create a traffic scenario, then the simulation starts. and we get finally a valuable output results.
The inputs needed by the simulation modules sumo and sumo-gui are:

- Road Network: For a simulation, a SUMO Road Network must be given using the option --net-file Cities-map.net.xml. The network is normally built using netconvert or netgenerate.
- Traffic Demand (Routes): routes are normally given to the simulation modules using the option - -route-files My-routes.rou.xml. We can use more than one route file within a

single simulation run. To be able sumo to simulate large road networks with up to millions of routes, it requires that the routes must be sorted by departure time of their vehicles include. Starting at the begin time step, new routes are loaded every n time steps for the next n time steps. n may be controlled using the - -route-steps <int> where ≤ 0 forces sumo/sumo-gui to load the file completely. The order of loading is as follows:

1. The network is read.
2. The route files are opened and the first n steps are read.
3. Each n time steps, the routes for the next n time steps are read.

- Defining the Time Period to Simulate: each simulation requires the definition about the time period to be simulated. a time period must be given using the option - -begin <time> and - -end <time>. The simulation starts at the time given in - -begin, which defaults to 0. All vehicles with a departure time (depart) lower than the begin time are discarded. The simulation performs each time step one-by-one. The simulation ends in the following cases:
  1. The final time step is given using - -end and this time step is reached.
  2. No value for - -end has been given and all vehicles have been simulated.
- Defining the Time Step Length: sumo/sumo-gui use a time step of one second per default. We may override this using the - -step-length <time> option. Giving - -step-length 0.01 will run the simulation using time steps of 10ms. Defining the time step length implies that vehicles perform calculations for the adaption of accelerations or lane-change maneuvers within every simulations step.

As outputs sumo may generate:

- Raw vehicle positions dump: the network dump(My-network-dump.xml), as it is defined in Listing 2.10, is a xml-file containing for each time step every edge of the network with every lane of this edge with all vehicles on this lane. For each vehicle, its name, speed and position on its respective lane are written.
- Trip information: the trip information(My-trip-info.xml), as it is defined in Listing 2.10, is a xml-file containing the information about each vehicle's departure time, the time the vehicle wanted to start at (which may be lower than the real departure time) and the time the vehicle has arrived. The information is generated for each vehicle as soon as the vehicle arrived at its destination and is removed from the network.
- Simulation state statistics: This output(My-summary.xml), as it is defined in Listing 2.10, is a xml-file containing for each time step the simulation-wide number of vehicles that are loaded, inserted, running, waiting to be inserted, have reached their destination and how long they needed to finish the route. The last value is normalised over all vehicles that have reached their destination so far. The output file looks like in Listing 2.12.

- emission output: SUMO allows users to track environmental statistics and emission values of all vehicles for every simulation step according to the Handbook of Emission Factors for Road Transport (HBEFA) standard[22].
- Aggregated Traffic Measures: When setting the option - -duration-log.statistics during the simulation, see 2.11, the following statistics will be printed:
  - Vehicle Counts: number vehicles that were Inserted, Running, Waiting and teleported.
  - Simulation timing data
  - The following averages for all vehicle trips
    * RouteLength: average route length
    * Duration: average trip duration
    * WaitingTime: average time spent standing
    * TimeLoss: average time lost due to driving slower than desired(desired speed= min(maxSpeed, edgespeed)) includes WaitingTime
    * DepartDelay: average time vehicle departures were delayed due to lack of road space

To add ecological output statistics for each of the vehicle used during the trip, we must add the option - -device.emissions.probability "1" in our sumo configuration file, see 2.10. In our work we will use the fuel consumption ecological output factor with ml unit measures.

```
<configuration>
  <input>
  <net-file value="Cities-map.net.xml"/>
  <route-files value="My-routes.rou.xml"/>
  </input>
  <output>
  <netstate-dump value="my-network-dump.xml"/>
  <tripinfo-output value="my-trip-info.xml"/>
  <summary value="my-summary.xml"/>
  <device.emissions.probability value="1"/>
  </output>
  <time>
  <begin value="0"/>
  <end value="2000"/>
  </time>
</configuration>
```

Listing 2.10. Sumo/sumo-gui Configuration file (My-simulation.sumocfg)

```
sumo -c  My-simulation.sumocfg  --duration-log.statistics
```

Listing 2.11. Command line of Sumo simulation

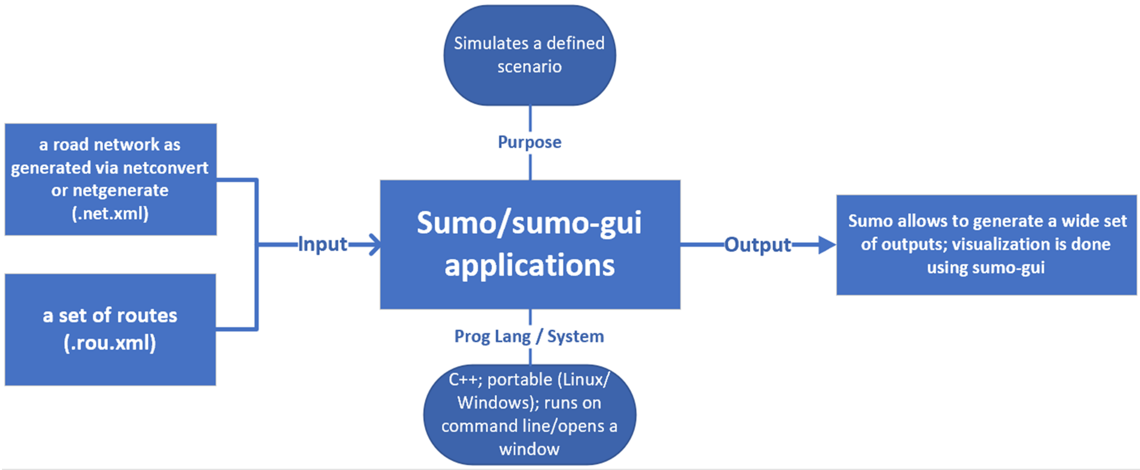


Figure 16. Description of sumo/sumo-gui applications.

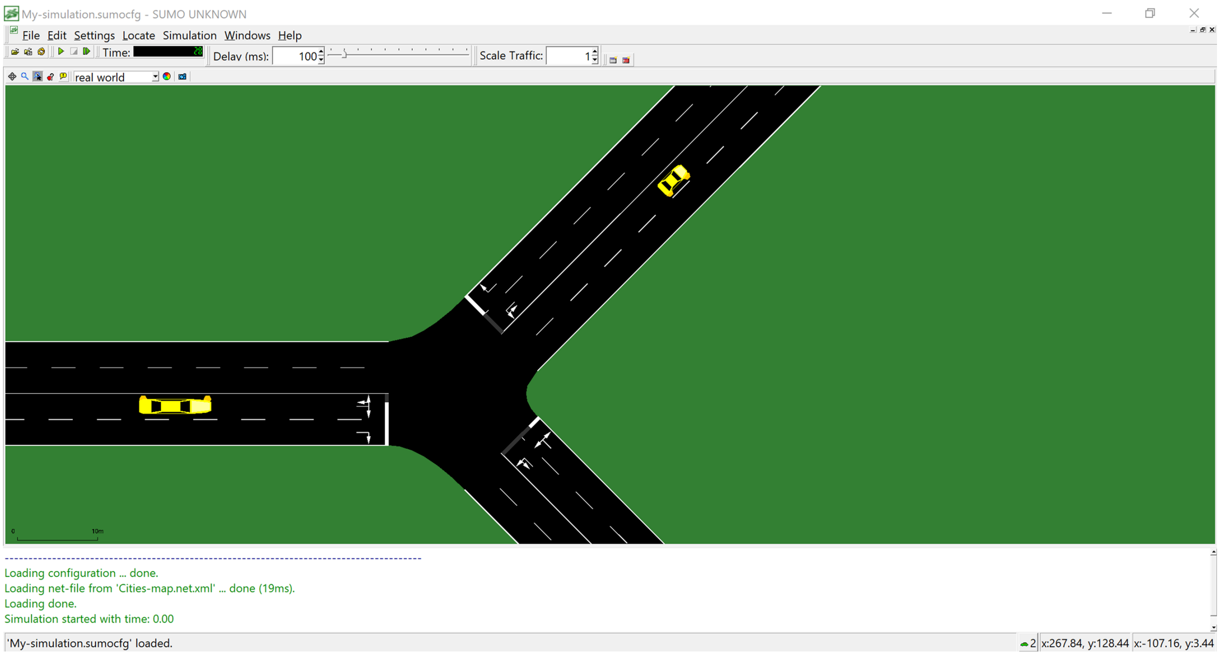


Figure 17. Sumo-gui user interface.

```
    <summary>
        <step time="0.00"
                loaded="2"
                inserted="1"
                running="1"
                waiting="0"
                ended="0"
                meanWaitingTime="0.00"
                meanTravelTime="-1.00"/>
```

```

        ... further time steps ...

    </summary>

```

Listing 2.12. Simulation output (My-summary.xml)

2.13 Use case : compute and build vehicle routes from demand definition using shortest path computation

For a given set of vehicles with of origin-destination relations (trips), the simulation must determine routes through the network (list of edges) that are used to reach the destination from the origin edge. The simplest method to find these routes is by computing shortest or fastest routes through the network using a routing algorithm such as Dijkstra or A*.

Basically, traffic assignment methods can be broadly classified into two categories as enlisted below :

- Static User Assignment Method
- Dynamic User Assignment Method

See the Routing models Subsection 1.5.3 for more information .

Static User Assignment methods (time-invariant variables) have been applied for a long time and for planning purposes, and are considered to be an important tool for strategic policy decisions mainly due to its simplicity and computational efficiency [23], whereas Dynamic User Assignment methods focus on the traffic control and management along with transportation planning and are used to generate a time varying traffic flow of the transport network which illustrates how congestion levels vary with time. [24].

2.13.1 Route planning primitive and models

Traffic assignment is the process of allocating given set of origin-destination pair to the existing suitable road network based on specific traveller's route choice criteria. The route choice criteria is the travel impedance of the transportation network be minimized for a given origin-destination pair.

SUMO Routing by Traveltime and Edge Priority

The objective of the routing algorithms is to minimize the travel time between origin and destination. The traveltime can either be computed from the maximum velocity allowed on the edge and the edge's length, it can be estimated at runtime from the simulation state.
SUMO can guide route search with additional information while still taking travel times into account. For this use case the option –WEIGHTS.PRIORITY-FACTOR FLOAT can be used with SUMO routing tools.
When this option is set, the priority value of each edge is factored into the routing decision so that low-priority edges receive a penalty (they appear to be slower) whereas high-priority edges receive little or no penalty. For the option value PRIORITYFACTOR, the penalty is computed thus:

```
 "MinEdgePriority" : minimum priority value of all edges
 "MaxEdgePriority" : maximum priority value of all edges
 "EdgePriorityRange" = MaxEdgePriority - MinEdgePriority

 "relativeInversePrio" = 1 - ((edgePriority - MinEdgePriority) /
EdgePriorityRange)
 "effort" =  traveltime * (1 + relativeInversePrio * PriorityFactor)
```

As a consequence:

- the highest priority edge will get no penalty
- the traveltime of the lowest priority edge is multiplied with 1+PriorityFactor,
- edges with in-between priorities will get a scaled penalty

SUMO Routing by distance

By default, vehicles in SUMO adhere to the maximum speed defined for the lane they are driving on (if the maximumSpeed of their vehicle type allows it).
A simpler solution In SUMO to find the shortest route rather than the fastests can be achieved by setting the speed for all network edges to the same value, and to define a vehicles type with the same maxspeed.
This solution is Unreasonable solution. To make it more real and authentic, we have implemented an independent routing by distance model. This model will help us, in chapter 5, to compare and to validate our implemented Ecological model.

2.13.2 Static User Assignment

Model description

Static User Assignment has been widely used for the problems of transportation planning and demand management policies decisions with respect to infrastructure investment. It is still preferred tool for the strategic transportation planning due to its simplicity and computation efficiency, and also very fast and can handle very large networks [23].

SUMO Routing algorithms(by default, Dijkstra), in Static User Assignment mode, search for each vehicle the fastest route through the empty network. It uses travel times of the edges which are computed from the maximum velocity allowed on the edge and the edge's length. Changes in the travel times due to previously routed vehicles are not regarded.

Duarouter application description

Duarouter application of SUMO can perform the Static User Assignment using the bellow command line, see Listing 2.13. As they are noticed in Figure 18, Duarouter imports different demand definitions and a road network to compute vehicle routes that may be used by sumo using shortest path computation.
The primary output of duarouter is a .rou.xml file. Additionally a .rou.alt.xml file with the same name prefix as the .rou.xml file will be generated. This route alternative file holds a routeDistribution for every vehicle. Such a routeDistribution is used during dynamic user assignment (DUA).

```
 duarouter --net-file Network.net.xml  --route-files Trips.trips.xml
--output-file Routes.rou.xml
```

Listing 2.13. Command line to generate routes using duarouter

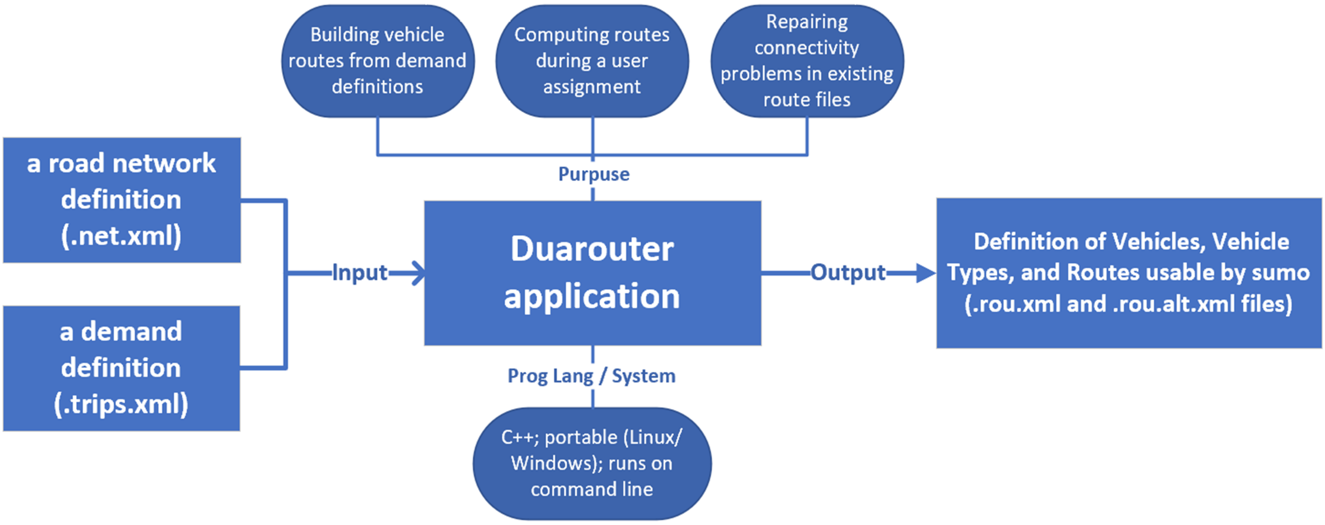


Figure 18. Description of Duarouter application.

duarouter : Dijkstra route planning algorithm

duarouter by default use Dijkstra algorithm, Dijkstra algorithm is a process of finding the path with the lowest cost (i.e. usually refers to the shortest path) from one node to all nodes in a city map. Its computation complexity is O(n^2) with n being the number of nodes in network [25]. Dijkstra is one of the optimal algorithms based on labeling method. In addition, other labeling algorithms like Bellman-Ford-Moore, incremental Graph, threshold, topological ordering, etc. are also used to find shortest path. F. Benjamin states that for finding the shortest path from one-to-one problem, it is worthwhile to consider Dijkstra algorithm since this algorithm is terminated as soon as the destination node is labeled, which also means that the shortest path is found [26].

duarouter's dependency components diagram

Figure 2 is a component diagram to show duarouter's dependency components and the relationships between them.

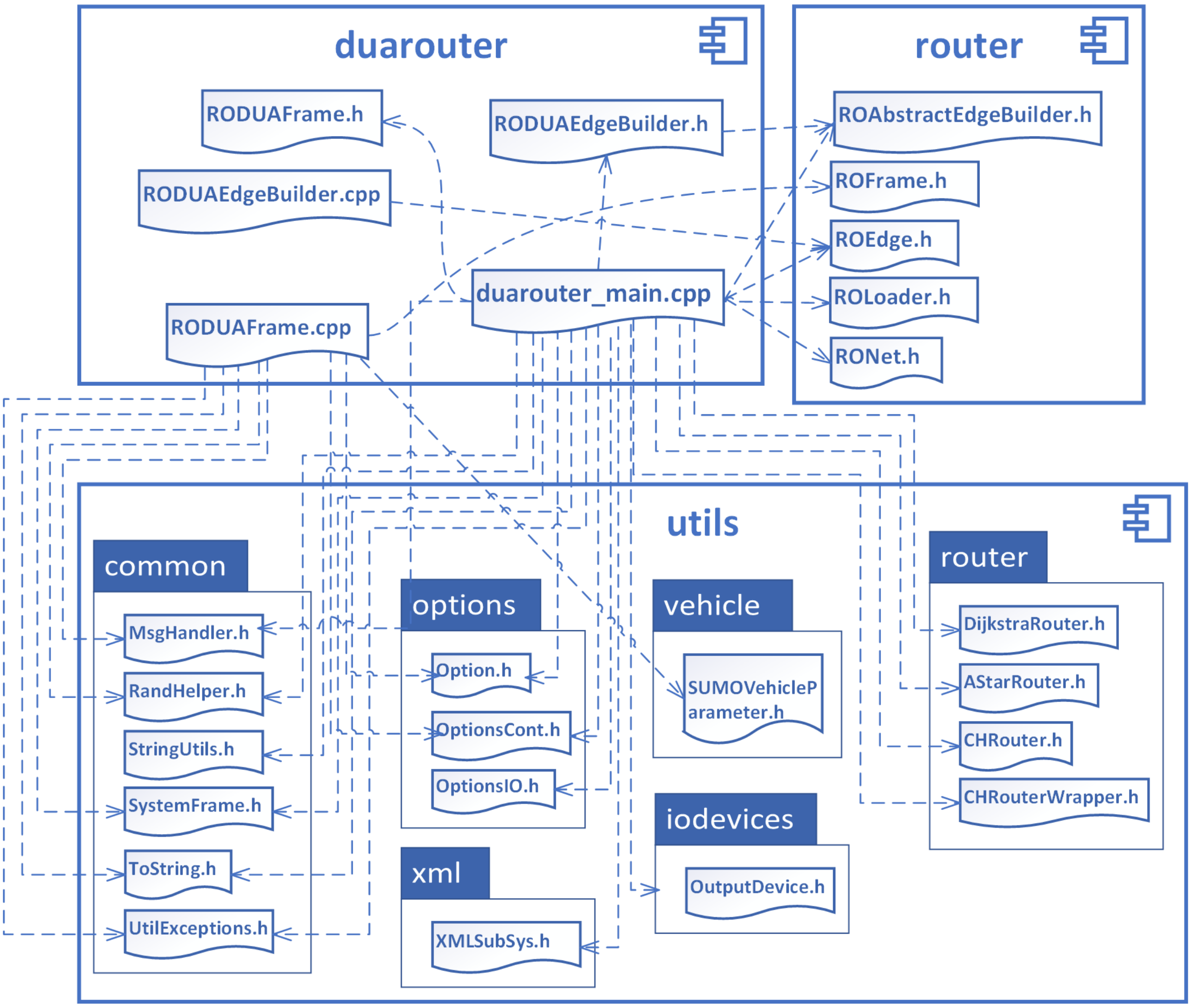


Figure 19. duarouter's dependency components diagram.

duarouter's use case diagram

Figure 20 represents the duarouter's use case diagram. Table 3 describes each functionality and giving the necessary configuration option to activate each use case

Functionality	Description
Command to repair connectivity problems for an existing route files	duarouter application allows user to repair connectivity problems in his existing route files using " $--repair$" configuration option; default: false.
Activate the user assignment (routing) with travel time model using shortest path computation	duarouter application allows user to perform a user assignment (routing) using shortest path computation.
Activate the dynamic user assignment (DUA) with travel time model using shortest path computation	duarouter application allows user to perform a dynamic user assignment (DUA) using shortest path computation by calling duarouter iteratively. This is facilitated by the tool duaiterate.py which converges to an equilibrium state (DUE).
Command to validate an existing trips	duarouter application allows user to validate her trips by rewriting the validating trips input using " $--write-trips$" configuration option; default: false.

Table 3. duarouter's use case description.

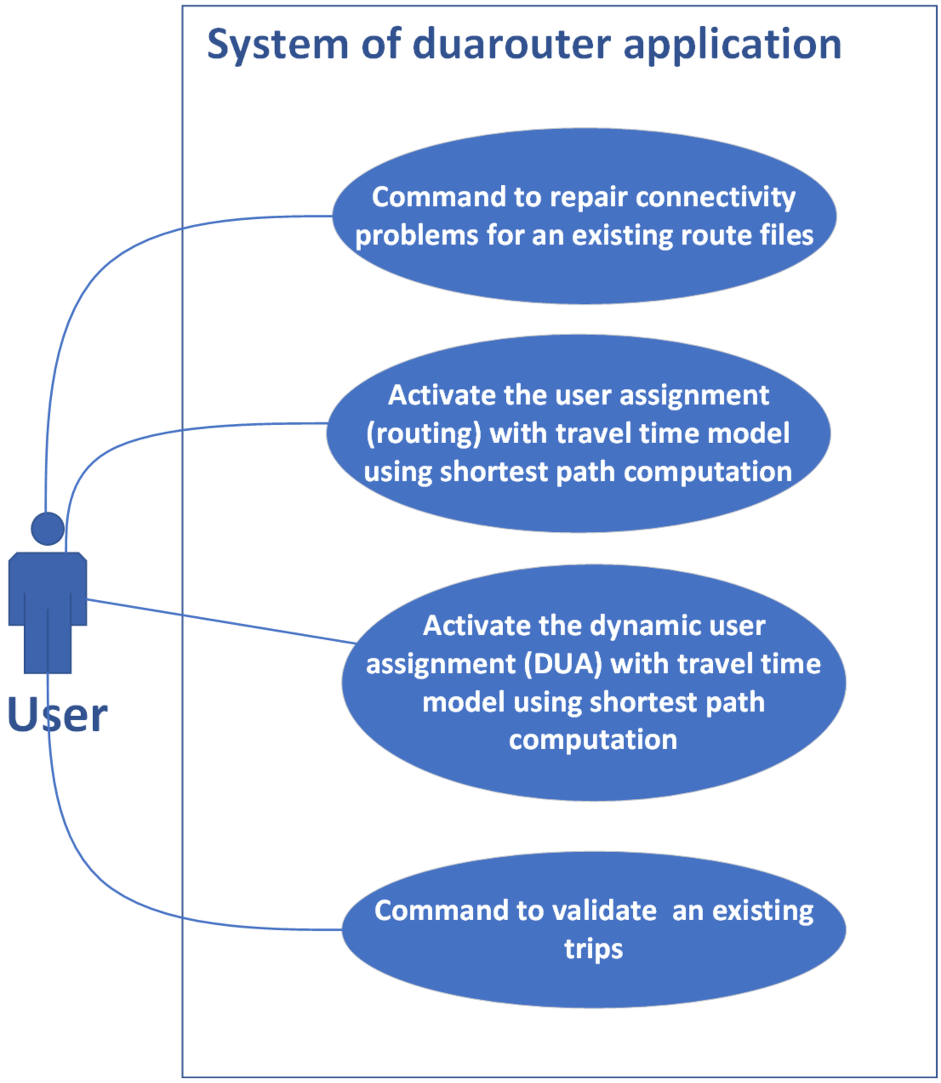


Figure 20. duarouter's use case diagram.

2.13.3 Dynamic User Assignment (Iterative Routing)

Model description of Duarouter's for dynamic user equilibrium

We can simply perform(approximate) the dynamic user equilibrium by calling iteratively the duarouter application. duarouter is integrated by a dynamic user assignment algorithm developed by C. Gawron (DUA-Gawron)[27]. This model is a microscopic approach meaning that the routes through a network to simulate are computed for every vehicle individually. The basic procedure is as follows:

- Step 1: Initialize the process by computing the fastest route through the empty network for each simulated vehicle. Set the usage probability for this route to 1

- Step 2: Perform the simulation using the current routes in order to obtain the edges' travel times over simulation time.
- Step 3: Compare the mean travel times to the last run (if any) and quit if the algorithm converges, i.e. if the mean travel time reduction falls below a given threshold.
- Step 4: Compute new routes for vehicles using the current travel times within the network. Then, continue with step 2.

Model description of duaiterate.py tool for dynamic user equilibrium

The tool DUAITERATE.PY facilitates the computation(approximation) of dynamic user equilibrium.

```
python duaIterate.py --net-file Network.net.xml  --route-files Trips.
trips.xml -l <nr-of-iterations>
```

Listing 2.14. Command line to perform dynamic user equilibrium using duaIterate.py

The command line script in Listing 2.14 tries to calculate a user equilibrium. It tries to find a route for each vehicle (each trip from the trip-file above) such that each vehicle cannot reduce its travel cost (usually the travel time) by using a different route. It does so iteratively by

1. calling duarouter to route the vehicles in a network with the last known edge costs (starting with empty-network travel times)
2. calling sumo to simulate "real" travel times result from the calculated routes. The result edge costs are used in the net routing step.

The number of iterations may be set to a fixed number of determined dynamically depending on the used options. Between successive calls of duarouter, the .rou.alt.xml format is used to record not only the current best route but also previously computed alternative routes. These routes are collected within a route distribution and used when deciding the actual route to drive in the next simulation step.

2.13.4 Dynamic User Assignment (Automatic Routing)

Model description(Simple One-shot routing)

One-shot algorithms have been proposed as an appropriate method for computing routes for each of the simulated vehicles[27]. The one-shot method implemented in SUMO computes a new route for each vehicle as soon as the vehicle is inserted into the network. The route is computed using the Dijkstra algorithm, where each edge's weight is continuously adapted to the travel time of this edge within the simulation. The used weight is:

$$w(t,e) = \begin{cases} l(e)/v_{\max}(e) & t = 0 \\ w(t-1,e)^{*}r + l(e)/v_{curr}(t,e)^{*}(1-r) & \text{otherwise} \end{cases} \tag{2.1}$$

where

- $w(t,e)$ weight of edge e at the current simulation step t
- $l(e)$ length of edge e
- $v_{\max}(e)$ maximum velocity allowed on edge e
- $v_{curr}(t,e)$ mean velocity of vehicles on edge e in time step t
- r remembering factor

the algorithm only needs to know the vehicle's start and end edges and the time the vehicle starts in order to compute a route. The default value of r is 0.5 [27].

Model description(One-shot with rerouting)

This method is an extension of the described one-shot routing approach[27], see the previous subsection 2.13.4 for the Simple One-shot routing. When the vehicle is inserted into the network, a new route is computed for each vehicle. Then, for every vehicle, a new, fastest route is computed every n simulation steps using the current edge weights $w(t,e)$ as long as the vehicle has not reached its destination.
When computing a new route for the vehicle, its destination is kept, whereas the edge the vehicle is currently at is used as the edge the new route shall start at. The part of the route after the vehicle's current edge is then directly replaced by the currently fastest continuation.

one-shot.py tool

The tool ONE-SHOT.PY provides a variant of the dynamic user assignment. Given trips will be assigned to respective fastest routes according to their departure times and a given travel-time updating interval. Different travel-time updating intervals can be defined by users, such as 900, i.e. link travel times will be updated every 900 sec. If the travel-time updating interval is set to -1, link travel times will not updated and link trave times at free-flow speeds are used for all trips.

The routes for oneShot assignment are computed using the automatic routing mechanism. A Dynamic user-equilibrium traffic state will not achieved with the use of this script.

Routing dynamically in the running simulation may be adequate in the following situations:

- there is not enough time and/or computing power to wait for the dynamic user equilibrium
- changes to the net occur while the simulation is running
- vehicles need to adapt their route while running

This routing approach works by giving some or all vehicles the capability to re-compute their route periodically. This routing takes into account the current and recent state of traffic in the network and thus adapts to jams and other changes in the network.

Listing 2.15 describes the command line, to perform the automatic routing using one-shot.py.

```
one-shot.py -f <travel-time updating interval> --net-file Network.net.xml
   --route-files Trips.trips.xml

where -f travel-time updating interval (sec); -1 means no travel-time
         updating (defaultt: -1,1800,300,15)
      --net-file network file name
      --route-files trip file name
```

Listing 2.15. Command line to perform automatic routing using one-shot.py

Conclusion

During this chapter, We described the most important cases of use for SUMO traffic simulator package using flowcharts, a use case diagrams and an illustrative scenario example to describe the most important modules of SUMO and especially the routing one.
SUMO is split into several features. Each of them has a certain purpose and must be run individually. This is something that makes SUMO different to other simulation packages.
In the next chapter, we will describe the eco-routing model to be implemented.

3. Description of the Eco-routing model

Contents

3.1 Fuel and CO_2 emission model **49**

3.2 Impacts of acceleration **53**

3.3 Eco-routing model description **54**

3.3.1 Network modeling and formulation 54

3.3.2 Eco-routing problem 55

3.3.3 Eco-routing model(solution) formulation 55

3.4 Numerical experiments for Eco-routing model **57**

3.4.1 First experiment 57

3.4.2 Second experiment 58

Introduction

In this chapter, We describe our implemented eco-routing model. we have picked the Nie and Li, 2013 eco-routing model [28]. This publication presents an eco-routing method that aims to find a path that minimizes the cost value of the fuel combined with travel time while meeting a given CO_2 constraint.
First, we discuss the consumption and pollutant emission model that used in our implementation. The authors, Nie and Li, propose a consumption and pollutant emission model that depends on vehicle speed, propose a consumption and pollutant emission model that depends on vehicle speed (similarly like Barth et al., 2007[15] and Yao and Song, 2013[17]), see the state of the art eco-routing subsection 1.5 for more information about these articles, but also consider the cost associated with switching from one road to another.
Second, we talk about the impacts of acceleration. The authors, Nie and Li, have modeled The speed changes with a constant acceleration model, like De Nunzio et al., 2016 [16]. The proposed routing model minimizes the cost value of the trip (a linear combination of the fuel cost and of the travel time cost) under the CO_2 constraint.
Then, we detail the eco-routing model [28] that aims to find a path that minimizes the cost value of the fuel combined with travel time while meeting a given CO_2 constraint.
Finally, we deliberate the Numerical experiments that they have used to validate the Nie and Li, 2013 eco-routing model [28]. In validation, The authors do not propose to use any specific routing algorithm. They define their eco-routing model as an optimization problem instead. They conducted a numerical experiments, which were solved using CPLEX optimization software. They were designed to show that ignoring travel speed variations and vehicle type causes CO_2 estimation errors.

3.1 Fuel and CO_2 emission model

- Their approach to estimating fuel and CO_2 emissions is built on CMEM (Barth et al., 2000 [29]).
- CMEM provides an analytical link between an individual vehicle's characteristics (mass, speed, acceleration etc.) to second-by-second fuel consumption rates.
- Once the fuel consumption rates are known, they are used for estimating CO_2 emissions based on carbon balance and the empirical relationships between CO/HC emissions and fuel consumption (Nam, 2003 [30]).
- CMEM converts total engine power, denoted as P, to fuel rates, denoted as f, using the following simple relationship (Barth et al., 2000 [29])

$$f = \phi P/\lambda, \tag{3.1}$$

where

- λ is the fuel's lower heating value.
- ϕ is the "air–fuel equivalence ratio" Most modern engines are configured to achieve ϕ = 1 under light engine load conditions, and ϕ > 1 for high load conditions such as acceleration. In what follows, they use ϕ_0 and ϕ_a to denote the air–fuel equivalence ratio under regular and high load conditions, respectively. The default value is 1.0 for ϕ_0 and 1.13 for ϕ_a.
- The total engine output P consists of three parts:
  * the tractive power used to move vehicle P_t,
  * the power draw of accessories such as air conditioning (P_a),
  * and the power wasted in overcoming engine friction (P_w).
- P may be represented as a function of velocity v and acceleration a as follows:

$$P = \sum_{i=0}^{3} \alpha_i v^i + \beta a v, \tag{3.2}$$

  where the coefficients are specified as:

  *

$$\begin{aligned} &\alpha_0 = \tfrac{P_a}{\eta}; \quad \alpha_1 = Zg\tfrac{G+c_1}{\eta\epsilon} + c_4K_0V\theta\left(\underline{r} + c_3v_h^2\right); \quad \alpha_2 = Zg\tfrac{c_1}{c_2\eta\epsilon} - 2c_3c_4K_0Vv_h\theta \\ &\alpha_3 = \tfrac{\rho C_d A}{2\eta\epsilon} + c_3c_4K_0V\theta; \quad \beta = \tfrac{Z(1+e_0)}{\eta\epsilon} \end{aligned} \tag{3.3}$$

- Table 4 offers a detailed description of all the variables and parameters.
- When the vehicle is in the idling state (v = 0, a = 0), P_t = 0, P_w can be estimated using (Barth et al., 2000 [29])

$$P_w = K_I V N_I, \tag{3.4}$$

  where $K_I \sim 1.5K_0$ and N_I is the idling engine speed. Hence, the power required in the idling state is

$$P_I = K_I V N_I + \frac{P_a}{\eta} = K_I V N_I + \alpha_0. \tag{3.5}$$

- Once the fuel rate is determined, the CO_2 emission rate, denoted as e_{CO_2}, can be estimated based on the carbon balance(Nam, 2003 [30]; Barth et al., 2000 [29]):

$$e_{CO_2} = \gamma_1 f + \gamma_0, \tag{3.6}$$

where

–

$$\gamma_0 = -\frac{A_r\,(\mathrm{CO_2})\,c_8}{A_r C + \mu}; \quad \gamma_1 = A_r\,(\mathrm{CO_2}) \left(\frac{1 - c_7}{A_r(C) + \mu} - \frac{c_5\,(1 - \phi^{-1}) + c_6}{A_r(\mathrm{CO})} \right) \tag{3.7}$$

– Details of the parameters used in Equation (3.6) can be found in Table 5.

- Equations (3.6) and (3.1) estimate the fuel and $\mathrm{CO_2}$ emissions in the unit of gram per second. To measure these emissions on the basis of distance. Let F and $\mathrm{E_{CO_2}}$ denote the fuel and $\mathrm{CO_2}$ emissions in the unit of gram per meter. F and $\mathrm{E_{CO_2}}$ can be calculated as

$$F(v, a) = f/v = \frac{\phi}{\lambda} \left(\sum_{i=0}^{3} \alpha_i v^{i-1} + \beta a \right), \tag{3.8}$$

$$E_{CO_2}(v, a) = e_{CO_2}/v = \gamma_1 F(a, v) + \frac{\gamma_0}{v}. \tag{3.9}$$

For the default valued reported in Table 2, $\gamma_0 \sim$ -0.016 and $\gamma_1 \sim 3$. Hence, in practice, ignoring γ_0 from $\mathrm{CO_2}$ estimation is an acceptable approximation.

- The authors validate the proposed model and indicate that the trend of the fuel and $\mathrm{CO_2}$ emissions captured seems to reasonably agree with the known empirical evidence.

Name	Description	Unit	Default value
Constant parameters			
g	Acceleration of gravity	Nm/s^2	9.81
c_1	Constant	-	0.01
c_2	Constant	m/s	44.73
η	Engine efficiency	-	0.4
Location specific parameters			
ρ	Air density	kg/m^3	1.247
Vehicle specific parameters			
A	Frontal area	m^2	2
C_d	Drag coefficient	-	0.3
K_0	Constant	J/rev/l	200
K_I	Constant	J/rev/l	300
N_I	Engine speed	rev/s	16.67
V	Engine displacement	l	2.0
j	Drive axle slippage	-	0.04
d	Wheel diameter	m	0.4064
$\bar{r}$	Constant	-	10
$\underline{r}$	Constant	-	2
v_h	Constant	m/s	35
Fuel specific parameters			
λ	Lower heating value	J/g	44.000
Operation specific parameters			
G	Grade	-	0
ϵ	Drivetrain efficiency	-	0.85
ϕ_0	Fuel–air ratio (regular conditions)	-	1.0
ϕ_a	Fuel–air ratio (acceleration)	-	1.13
P_a	Axillary power	W	1000
Z	Mass	kg	1500
Variables			
v	m/s	-	
a	m/s^2	-	

Table 4. Description of parameters used in calculating fuel emission rates [28].

Name	Unit	Value
$A_r(CO_2)$	-	44
$A_r(C)$	-	12
$A_r(CO)$	-	28
μ	-	1.85
c_5	-	0.4074
c_6	-	0.1174
c_7	-	0.01
c_8	g/s	0.0049

Table 5. Description of parameters used in calculating CO_2 emissions [28].

3.2 Impacts of acceleration

- the authors propose a new method that aims to approximate the impacts of acceleration while excluding acceleration as a decision variable in the model.
- The proposed method divides the movement of a vehicle on each link into two stages, an acceleration/deceleration stage and a cruise stage.
- Once a vehicle enters a link, it will change its cruise speed to the current link's prevailing cruise speed, denoted as v_1.
- Denote the initial speed at the entrance as v0, which may be either zero (if the vehicle starts the trip at the link) or the cruise speed of the predecessor link.
- To simplify the analysis, we assume that
  1. (1) the vehicle enters the link at time $t = 0$;
  2. (2) the acceleration/deceleration rate a is a constant;
  3. and (3) the length of the link, denoted as l, is long enough to allow the vehicle's speed to reach v_1.
- As per Assumptions (1) and (2), the time and distance required to reach v_1 are respectively $t_a = (v_1 - v_0)/a$, and $l_a = (v_1^2 - v_0^2)/2a$ (the distance it takes to accelerate from v_0 to v_1 at a constant acceleration a).
- As per Assumptions (3) implies that $l_a < l$.
- Ignoring the speed fluctuations and the within-link acceleration events may not adversely impact eco-routing decisions, because in the congested traffic conditions, travel time and emissions are not conflicting objectives (i.e. shorter travel time will always lead to lower emissions).
- Let $f(t)$ be the fuel emission rate at time t. Recalling Equation (3.8), the emission rate at

any given time, with a given constant a and an initial speed v_0, is

$$f(t) = \frac{\phi}{\lambda}\left(\sum_{i=0}^{3} \alpha_i \left(v_0 + at\right)^i + \beta a \left(v_0 + at\right)\right). \tag{3.10}$$

- Ignoring the impacts of deceleration events is conservative from the system perspective since it is most likely to overestimate CO_2 emissions.
- The total fuel emissions of a link with cruise speed v_1 and length l are estimated as follows:

$$T_F = \begin{cases} lF\left(v_1, 0\right) + \phi_a \sigma_1 & v_0 \leqslant v_1 \\ lF\left(v_1, 0\right) & v_0 > v_1 \end{cases} \tag{3.11}$$

where

$$\sigma_1 = \frac{\beta}{2\lambda}(v_1^2 - v_0^2); \tag{3.12}$$

- Such a simplification is useful because it allows one to estimate with reasonable accuracy the impacts of acceleration on total emissions, without requiring the knowledge of actual acceleration.

3.3 Eco-routing model description

3.3.1 Network modeling and formulation

- Consider a network G(*N*,*A*) that consists of a set of nodes *N* and a set of links A. Each link b is associated with a length l_b and a fixed prevailing travel speed v_b.
- We assume that all drivers must drive at v_b on link b. In practice, v_b may be obtained based on historical data collected from traffic sensors or estimated using urban travel forecasting tools. If necessary, such data can also be aggregated according to the time of day to capture within-day variation.
- Let $l(i)$ and $O(i)$ be the set of incoming and outgoing links at each node $i \in N$.
- Let $a \in I(i)$ and $b \in O(i)$ be a pair of links that identify a possible turning movement, denoted as m, across node i.
- Each m is defined by the origin link a and the destination link b. For convenience, this is written as m^- = a, m^+ = b.
- A set of all valid movements at node i is denoted as M_i.
- For each $m \in M_i$, $\tilde{w}_m$ denotes the waiting time of the movement, modeled as a discrete and independent random variable.
- $\tilde{w}_m$ can take a discrete set of values $\{w_m^0, w_m^1, ...\}$ with a corresponding probability mass $\{p_m^0, p_m^1, ...\}$. w_m^0 is assumed equal to 0 for any m.

3.3.2 Eco-routing problem

- The proposed Eco-routing problem aims to find a route between an origin (o) and a destination (d) that minimizes the total travel cost, while meeting a given CO_2 emission standard.
- The route travel cost is the total monetary value of both fuel and time consumed over the route.
- Note that the route cost here involves not only the costs on its member links, but also the costs associated with switching from one link to the next, i.e. the movement cost.

3.3.3 Eco-routing model(solution) formulation

- Let t_b and F_b denote the travel time and fuel consumption on link b. We have

$$t_b = \frac{l_b}{v_b}; \quad F_b = F(v_b, 0)l_b; \tag{3.13}$$

where $F(.)$ are defined in Equation (3.8).

- To compute the movement cost, let $\hat{t}_m$ and $\hat{F}_m$ denote the waiting time and fuel emissions associated with the movement m. We have

$$\hat{t}_m = \sum_{k=0} w_m^k p_m^k; \tag{3.14}$$

$$\hat{F}_m = max\left(0, \frac{\phi_a \beta (v_{m^+}^2 - v_{m^-}^2))}{2\lambda} p_m^0\right) + \sum_{k=1} p_m^k \left(\frac{\phi_a \beta v_{m^+}^2}{2\lambda} + p_I \frac{\phi}{\lambda} w_m^k\right); \tag{3.15}$$

where $\hat{t}_m$ is the expected waiting time on m, and $\hat{F}_m$ consists of both the emission associated with engine idling (because of waiting), and the extra energy consumed to accelerate the vehicle from stationary to the cruise speed on the next link m^+ (as defined in Function (3.13)). When the waiting time is zero, the extra energy consumption is associated with the speed change from m^- to m^+, and equals zero if $v_m^- > v_m^+$.

- Let E_b and $\hat{E}_m$ denote the CO_2 emissions on link b and movement m respectively, we use the following approximation

$$E_b = F_b \gamma_1; \quad \hat{E}_m = \hat{F}_m \gamma_1; \tag{3.16}$$

where γ_1 is defined in Function (3.13)

The eco-routing problem described above may be formulated as follows:

$$\textbf{min} \sum_{b \in A} (W_t t_b + W_f F_b)\, x_b + \sum_{i \in N} \sum_{m \in M_i} \left(W_t \hat{t}_m + W_f \hat{F}_m \right) y_m, \tag{3.17a}$$

$$\textbf{subject to :} \sum_{m \in M_i, m^- = b} y_m = x_b, \quad \forall i \in N, \;\; b \in I(i), \tag{3.17b}$$

$$\sum_{m \in M_i, m^+ = b} y_m = x_b + q_b, \quad \forall i \in N, \;\; b \in O(i), \tag{3.17c}$$

$$q_b = \begin{cases} 1 & b = s \\ -1 & b = r \\ 0 & otherwise \end{cases} \quad \forall b \in A\,, \tag{3.17d}$$

$$\sum_{b \in A} E_b x_b + \sum_{i \in N} \sum_{m \in M_i} (\hat{E}_m) y_m \leqslant \sum_{b \in A} l_b x_b \bar{E}, \tag{3.17e}$$

$$x_b \in \{0, 1\}\,; \quad \forall b \in A, \tag{3.17f}$$

$$y_m \in \{0, 1\}\,; \quad \forall m \in M_i, \;\; \forall i \in N, \tag{3.17g}$$

In this formulation :

- x_b and y_m are decision variables that depict routing decisions.
- The objective function ((3.17a)) represents the total expected cost of a given route, where W_t and W_f are the price of travel time and of fuel, respectively. t_b, F_b, $\hat{t}_m$ and $\hat{F}_m$ are defined in Equations (3.13)-(3.15).
- Constraints (3.17b), (3.17c) and (3.17d) state the link-based flow conservation conditions.
- The origin link r in (3.17d) is a dummy link ending at o and the destination link s is a dummy link starting from d. The travel costs on both dummy links are zero.
- Constraint (3.17e) is the emission constraint, where $\bar{E}$ equals to the proposed emission standards.
- Finally, Constraints (3.17f) and (3.17g) require x_b and y_m be binary variables.

With the emission constraint, the eco-routing problem Equation (3.17) falls into a class of constrained shortest path problem which is known to be NP-complete ([31]). Thus, solving Equation (3.17) on large networks may require specialized algorithms.

The eco-routing problem without the emission constraint may be solved by a standard shortest path algorithm on on large networks [28].

3.4 Numerical experiments for Eco-routing model

To validate the eco-routing model proposed in the previous section. the authors in [28] conduct a numerical experiments.

1. The first experiment is designed to examine the sensitivity of the model to selected vehicle characteristics (such as engine displacement, weight, auxiliary power).
2. The second experiment aims to demonstrate the impacts of acceleration and idling on the route choice decision.

and, they set W_t = \$15/h and W_f = \$0.0015/g. Note that they calculate W_f by assuming the price of gasoline is \$4/gallon and the gasoline weight is 6.073 lb/gallon. They code The models in AMPL and they solve them by CPLEX on a desktop computer with an Intel i5–2400 @3.1 GHz CPU and 16 Gb RAM.

3.4.1 First experiment

In first experiment, they use a simple 4-node network to examine the impacts of vehicle characteristics. They conduct the test on four cases involving three different types of vehicles, as detailed below, Parameters not mentioned below take the default values reported in Table 1.

1. passenger car: Z = 1500 kg, C_d = 0.3, V = 2 L, P_a = 1000 W
2. passenger car with air condition turned on: Z = 1500 kg, C_d = 0.3, V = 2L, P_a = 3000 W
3. minivan: Z = 2730 kg, C_d = 0.35, V = 3.6 L, P_a = 1000 W
4. light duty truck: Z = 1900 kg, C_d = 0.53, V = 5 L, P_a = 1000 W

The network contains only three paths. Thus, they don't use the emission constraint to filter out paths when we solve the eco-routing model.

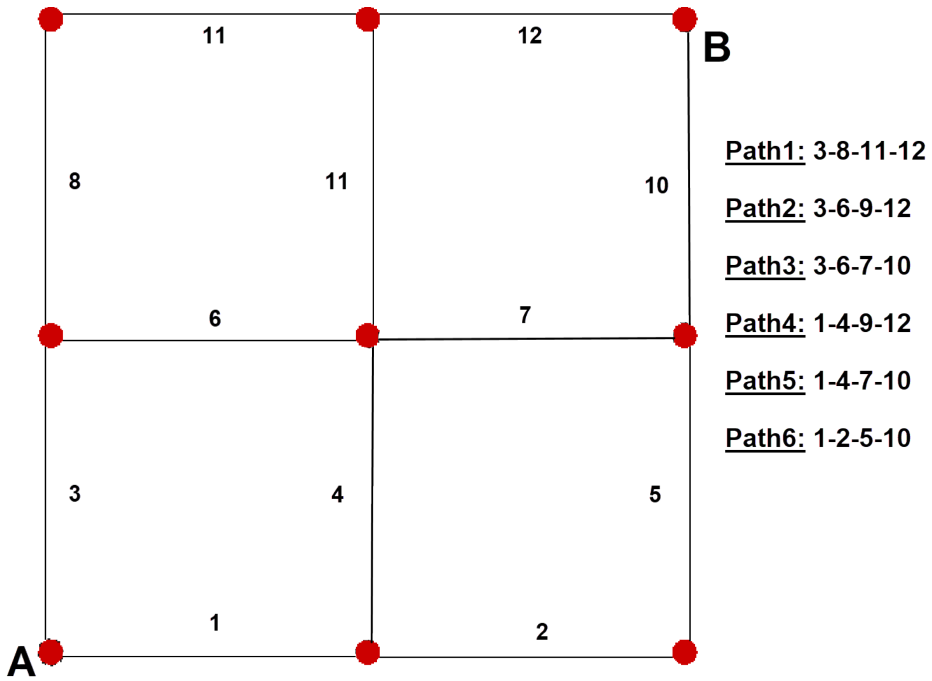
The experiment indicates that :

1. The total costs on all three paths increase when the vehicle type is changed from passenger car to minivan and then to light truck. However, the magnitude of the increase depends on the type. Because of that, a light truck driver would rank the three paths differently than those who drive other vehicles.
2. The results indicate that the impact of using air conditioning during operation is insignificant. Turning on air conditioning results in 25 g (less than 3%) more CO2 emissions and

about 10 cents more fuel cost in a 10-mile journey. Thus, avoid air conditioning while driving does not seem to be a worthwhile sacrifice for the sake of sustainability

3.4.2 Second experiment

In second experiment, They use a simple grid network that mimics a portion of a typical city center as designed in Figure 21. The network consists of twelve links and nodes A and B are set as the origin and destination for eco-routing. There are six possible paths and 18 possible movements for the given O–D pair. The movements are grouped into three categories: left turns, right turns and through movements. They assume the Movements in the same category to have the same waiting time distribution. The left turns cause longest expected delay while right turns have the lowest expected delay. The emission standard $\bar{E}$ is set to 400 gram per mile to ensure the constraint is not binding. Finally, they enumerate in Table all six paths and computes their total costs in each scenario.



Movement	Delay (min)	Probability	Expected Delay (min)
Left	0, 1.2, 1.8	0.2, 0.3, 0.5	1.26
Through	0, 0.6, 1.2	0.5, 0.3, 0.2	0.42
Right	0, 0.6	0.5, 0.5	0.3

Figure 21. Network topology and movement properties of the grid network[28].

They conduct the test on four scenario involving three different types of vehicles, each corresponding to a specific objective function, as detailed below.

1. Case 1: The objective function is set as $\sum_{b\in A}(W_t t_b + W_f F_b)x_b + \sum_{i\in N}\sum_{m\in M_i}(W_t \hat{t}_m + W_f \hat{F}_m)y_m$, which consists of both time cost and emission cost. Time cost (fuel cost) refers to the sum of time (fuel) cost on the path and waiting time (fuel consumption) at intersections.
2. Case 2: The objective function is set to $\sum_{b\in A}(W_t t_b + W_f F_b)x_b + \sum_{i\in N}\sum_{m\in M_i} W_t \hat{t}_m y_m$, which suppresses the effect of acceleration.
3. Case 3: The objective function is set to $\sum_{b\in A}(W_t t_b + W_f F_b)x_b$. That is, the effect of turning movements is not taken into account.
4. Case 4: The objective function is set to $\sum_{b\in A} W_t t_b x_b$, which represents the conventional shortest path problem.

The results indicate that :

1. the optimal path in Case 1 which both time cost and emission cost are considered is Path 1.
2. the optimal path in Case 2 when the effect of acceleration is ignored is Path 1.
3. the optimal path in Case 3 when turning movements are suppressed changes to Path 5.
4. the optimal path in Case 4 which represents the conventional shortest path changes also to Path 5.
5. Path 1 outperforms all the other paths when turning movements and fuel consumption is taken into account, although that it has the highest cost in Case 4.
6. Path 5 degrades from the best to the second least favorable path as more measures are considered
7. In this experiment the conventional shortest path with 16.7% less travel time compared to Path 5 ends up with 6.7% higher total cost. Thus, the consideration of microscopic vehicle operating conditions does affect Eco-routing.

They illustrate the effect of turning movement by comparing paths 1 and 2 :

1. If turning movements and acceleration are taken into consideration (Case 1), eco-drivers would prefer path 1 ($3.21) to Path 2 ($3.52). This is because Path 1 only has right turns and through movements while Path 2 includes one left turn and two right turns, which leads to increased time cost and fuel consumption.
2. If the effect of turning movements is ignored,both paths (1 and 2) admit the same total travel cost ($2.78), because they have exactly the same length and free flow travel time.

The extra cost on Path 2 in Case 1 comes from waiting at intersections (both in terms of idling and implicit acceleration events). Such a cost could potentially change the path ranking.

There experiment only shows the saving from avoiding one left turn. Savings could be much larger in a real network where multiple left turns may be included.

Path	Case 1	Time cost[1]	Fuel cost[2]	Emission (gpm)	Case 2	Case 3	Case 4
1	3.21	2.55	0.67	368.0	3.12	2.78	2.16
2	3.52	2.84	0.69	377.4	3.45	2.78	2.16
3	3.34	2.67	0.67	370.5	3.28	2.68	2.07
4	3.32	2.64	0.68	370.1	3.25	2.68	2.07
5	3.44	2.76	0.68	373.5	3.36	2.58	1.98
6	3.34	2.67	0.67	370.5	3.28	2.68	2.07

Table 6. Total costs of all paths (all in monetary value except for 'Emission')[28]. [1] Time cost in Case 1, [2] Fuel cost in Case 1

Conclusion

Throughout this chapter, we Clearly describe the eco-routing model to be implemented, in particular the parameters relating to this model and the algorithms and state changes, which govern their operation.
We recapitulate that the picked,Nie and Li, 2013 eco-routing model publication [28], presents an eco-routing method that aims to find a path that minimizes the cost value of the fuel combined with travel time while meeting a given CO_2 constraint.
In the next chapter, we will tackle the process of integration and implementation of our picked and described Nie and Li, 2013 eco-routing model [28].

4. Implementation of the Eco-routing model in SUMO

Contents

4.1 Requirement analysis and specification **62**

4.1.1 Actors Identifications 62

4.1.2 Functional requirements 62

4.1.3 Non-Functional requirements 63

4.1.4 duarouter's use case diagram with our extended eco-routing model . 64

4.1.5 Use case : Perform routing with eco-routing, distance and travel time models 65

4.2 Conceptual design **72**

4.2.1 Static modeling: class diagrams 72

4.2.2 dynamic modeling: sequence diagrams 74

4.2.3 Sequence diagram for " Load the road network" 74

4.2.4 Sequence diagram for "Load trips" 75

4.2.5 Sequence diagram for "Activation of routing algorithm" 77

4.2.6 Sequence diagram for "Routing with Ecological model" 78

4.3 Realisation **80**

4.3.1 The Integrated objectives functions and parameters 80

4.3.2 Hardware environment 83

4.3.3 Software environment 83

4.3.4 Technologies used 85

Introduction

In this Chapter, First, we extract and describe the different requirement analysis and specification for our duarouter's extend implementation. Second, we design our static and dynamic view of our proposed system using UML diagrams. Then, we rehighlighte and describe the objectives functions and their parameters that used in our implemented Eco-Routing model to calculating each edge's Eco-Routing weight. Finally, we present the technology stack including hardware and software.

4.1 Requirement analysis and specification

Each and every software or application has a set of functional and non-functional requirements describing the tasks or services that it is built to deliver or execute. In next subsections we will specify the functional and non-functional requirement.

4.1.1 Actors Identifications

The travel impedance for the transportation network are travel time, travel cost, travel distance. However, due to congestion and transportation environmental pollution, route choice not only depends on the shortest path but also it depends on ecological parameters

Therefore, we want by this present project,

- to add a modest material for **the eco-routing models field** and for the eco-routing models **Junior Researchers** for a quick start.
- Also, to insist that **the road users** should focus on the optimal path rather than the shortest path in their daily routes planning.

Optimal paths may be defined as the paths having a reasonable travel time or cost along with higher safety and ecological factors respect

4.1.2 Functional requirements

The functional requirement used to better understand the role of the application as well as its main functionalities and the expectations of the actor of the application to be developed.
In our contributed application, we have only one action Our contributed implementation must address the following main modules:

- **Routing with ecological model:**
  The user can activate the user assignment(Routing) with Eco-Routing model using shortest path computation and he gets his routes file eco-routing based
- **Routing with distance model:**
  The user can activate the user assignment(Routing) with Distance routing model using shortest path computation and he gets his routes file Distance based
- **Routing with time model:**
  The user can still activate the user assignment(Routing) with time routing model using shortest path computation and he gets his routes file time based
- **Dynamic Routing:**
  The user can perform the dynamic user assignment for his traffic scenario
- **Begin time:**
  The user can add the begin time of routes computation.
- **End time:**
  The user can add the end time of routes computation.
- **Routing algorithm:**
  The user can choose the routing algorithm to perform shortest path computation among ['dijkstra', 'astar', 'CH', 'CHWrapper'].
- **Get help:**
  The user can type for help.
- **Add SUMO-network**
  The user can add file as SUMO-network format to route on.
- **Add trips:**
  The user can add file(s) as sumo routes, alternatives, flows, and trips to get a full routes based on predefined routing criteria.
- **Get verbose's output:**
  The user can get a verbose's additional information output.
- **Get full log output file:**
  The user can can get all log messages into file.

4.1.3 Non-Functional requirements

Besides functional requirement, there are non-functional requirements are important and their lack can lead this application to perish. They boost performance and quality.
The non-functional requirements for our contribution are as follows:

- **simplicity:**

Application must have simple treatment, so that users easily handle it. The user must activate a desired model and functionality with a simple and efficient way.

- **Usability:**
  it defines the extent to which users can learn, operate, prepare, interpret application's outputs through using the application. ISO defines usability as "The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use".
- **Maintainability:**
  this requirement is related to the source code quality, it must be readable, understandable and easy to upgrade and update.
- **Scalability:**
  this point also is related to code as it must allow reuse and evolution by making this application extensible to offer other features to it.
- **Compatibility :**
  YOU must cover the possible range of operating system. The extended duarouter is a command line, portable application (Linux/Windows is tested).

4.1.4 duarouter's use case diagram with our extended eco-routing model

Based on the functionalities above, we will detail the general diagram of the use cases for our application. In facts this diagram is general and does not detail all the functionality one by one. It brings together a set of functionalities in a single case which can in turn be detailed in order to better understand it. Figure 22 represents the general use case diagram.

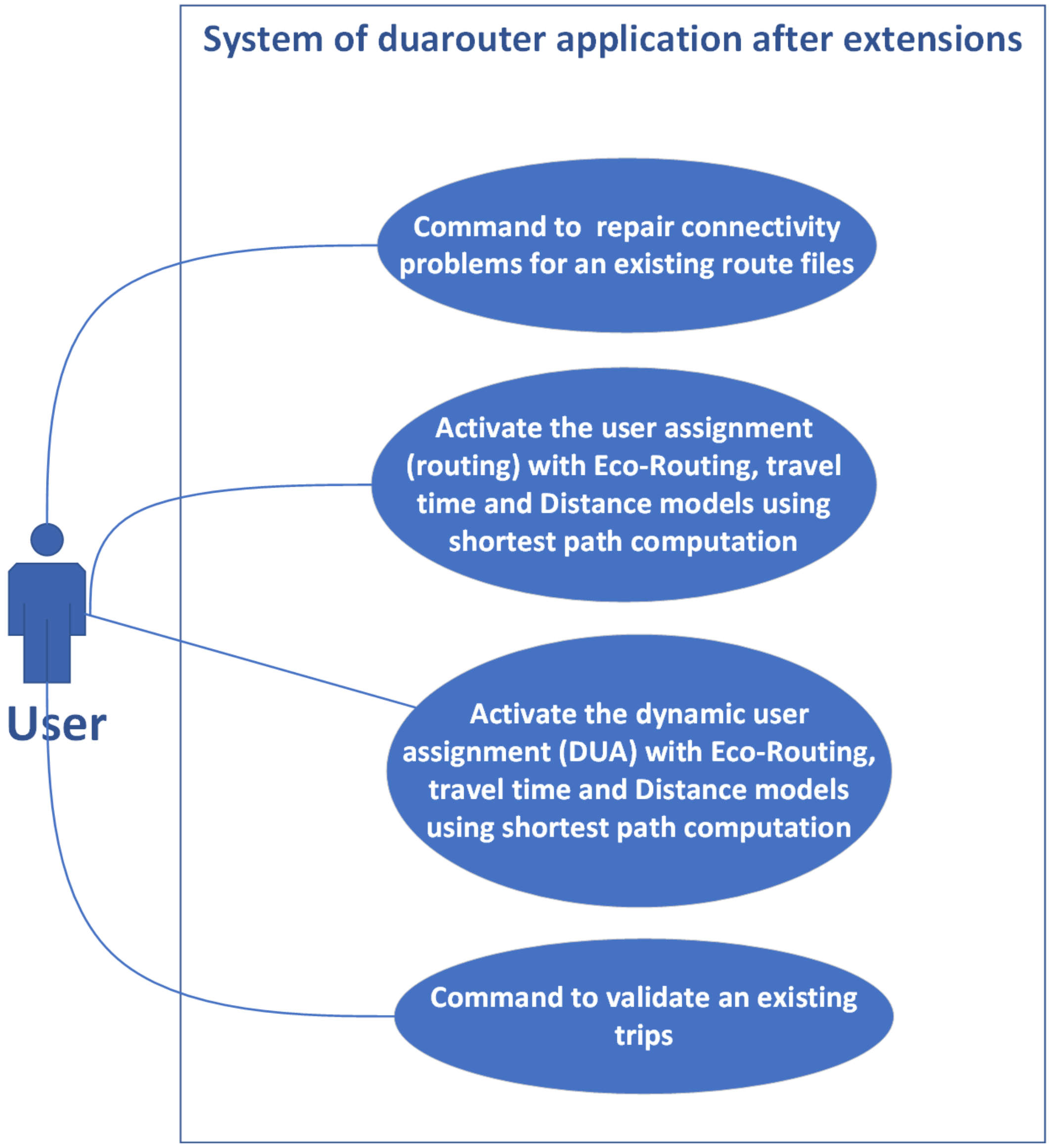


Figure 22. Extented duarouter's use case diagram.

4.1.5 Use case : Perform routing with eco-routing, distance and travel time models

The refinement of the use case "Perform routing with eco-routing, distance and travel time models use case diagram" is illustrated in Figure 23.

Figure 23. Perform routing with eco-routing, distance and travel time models use case diagram.

Use case description of "Activate a routing model"

Table 7 details how the use case "Select a routing model" works

Functionality	Activate a routing model
Actor	User
Precondition	open duarouter's command line application
Nominal scenario	1. Add the configuration option $--weight-attribute$ 2. type a routing model name : Eco-routing or travel time 3. duarouter will compute and build the traffic assignment of all validating trips with the previous selected model using shortest path computation. 4. duarouter will return a full description route file usable by sumo/sumo-gui based on the previous selected routing model.
Postcondition	A route file based on the previous selected routing model usable by sumo/sumo-gui

Table 7. Use case Description of "Activate a routing model".

Use case description of "Activate a routing algorithm"

Table 8 details how the use case "Activate a routing algorithm" works

Functionality	Activate a routing algorithm
Actor	User
Precondition	open duarouter's command line application
Nominal scenario	1. Add the configuration option $--routing-algorithm$ 2. type the name among those routing algorithms ['dijkstra', 'astar', 'CH', 'CHWrapper'] default:dijkstra. 3. duarouter will compute and build the traffic assignment using shortest path computation with the previous selected routing algorithm. 4. duarouter will return a full description route file usable by sumo/sumo-gui based on the previous selected routing algorithm.
Postcondition	A route file based on the previous selected routing algorithm usable by sumo/sumo-gui

Table 8. Use case Description of "Activate a routing algorithm".

Use case description of "Command to load all routes in one step or go through them stepwise"

Table 9 details how the use case "Select a routing model" works

Functionality	Command to load all routes in one step or go through them stepwise
Actor	User
Precondition	open duarouter's command line application
Nominal scenario	1. Add the configuration option $" --route-steps"$ 2. type the next number of seconds that referred to the set of loaded routes in each step. The purpose is to not keep all routes in memory for a better performance. If $" --route-steps$ 0" duarouter will load all routes in one step; default: 200. 3. duarouter will compute and build the traffic assignment. it will Start at the begin time step, new routes are loaded every n time steps for the next n time steps. 4. duarouter will return a full description route file usable by sumo/sumo-gui that may be up to millions of routes.
Postcondition	A route file that may be up to millions of routes usable by sumo/sumo-gui

Table 9. Use case Description of "Command to load all routes in one step or go through them stepwise".

Use case description of " Add file as SUMO-network format to route on"

Table 10 details how the use case "Add file as SUMO-network format to route on" works

Functionality	Add file as SUMO-network format to route on
Actor	User
Precondition	open duarouter's command line application
Nominal scenario	1. Add the configuration option " $--net-file$" 2. type the name and the path of his desired SUMO-network format to route on. 3. duarouter will compute and build the traffic assignment using shortest path computation based on inserted network definition 4. duarouter will return a full description route file usable by sumo/sumo-gui based on inserted network definition.
Postcondition	A route file for a specific SUMO-Network format usable by sumo/sumo-gui

Table 10. Use case Description of "Add file as SUMO-network format to route on".

Use case description of "Add file(s) as sumo routes, alternatives, flows and trips"

Table 11 details how the use case "Add file(s) as sumo routes, alternatives, flows and trips" works

Functionality	Add file(s) as sumo routes, alternatives, flows and trips
Actor	User
Precondition	open duarouter's command line application
Nominal scenario	1. Add the configuration option " $--route-files$" 2. type the name and the path of his trips file. 3. duarouter will compute and build the traffic assignment using shortest path computation for those set of defined trips. 4. duarouter will return a full description route file usable by sumo/sumo-gui for those set of defined trips.
Postcondition	A route file for those set of defined trips usable by sumo/sumo-gui

Table 11. Use case Description of "Add file as SUMO-network format to route on".

Use case description of " Activate verbose's additional information output"

Table 12 details how the use case "Activate verbose's additional information output" works

Functionality	Activate verbose's additional information output
Actor	User
Precondition	open duarouter's command line application
Nominal scenario	1. Add the configuration option " $--verbose$" 2. Turn the configuration option to $True$; default: false. 3. duarouter will compute and build the traffic assignment. 4. Many additional informations and data about duarouter's process and computation will be printed in commandline output.
Postcondition	Additional information will be printed in commandline output

Table 12. Use case Description of "Activate verbose's additional information output".

Use case description of "Command to write all log messages into file"

Table 13 details how the use case "get all log messages written in file" works

Functionality	Command to write all log messages into file
Actor	User
Precondition	open duarouter's command line application
Nominal scenario	1. Add the configuration option " $--log$" 2. Turn the configuration option to $True$; default: false. 3. duarouter will compute and build the traffic assignment. 4. user will get all log messages written in file
Postcondition	get all log messages written in file

Table 13. Use case Description of "Command to write all log messages into file".

Use case description of "Add the begin time"

Table 14 details how the use case "Add the begin time" works

Functionality	define the begin time
Actor	User
Precondition	open duarouter's command line application
Nominal scenario	1. Add the configuration option " $--begin$" 2. Enter the begin time of route computation; default: 0. 3. duarouter will compute and build the traffic assignment. the previous trips of this typed begin time will be discarded. 4. A route file with a discarded previous begin-time's trips usable by sumo/sumo-gui
Postcondition	A route file with a discarded previous begin-time's trips usable by sumo/sumo-gui

Table 14. Use case Description of "Add the begin time".

Use case description of "Add the end time"

Table 15 details how the use case "Add the end time" works

Functionality	define the end time
Actor	User
Precondition	open duarouter's command line application
Nominal scenario	1. Add the configuration option " $--end$" 2. Enter the end time of route computation; Defaults to the maximum time that SUMO can represent; default: 9223372036854774. 3. duarouter will compute and build the traffic assignment. the later trips of this typed end time will be discarded. 4. A route file with a discarded later end-time's trips usable by sumo/sumo-gui
Postcondition	A route file with a discarded later end-time's trips usable by sumo/sumo-gui

Table 15. Use case Description of "Add the end time".

Use case description of "Ask for help"

Table 16 details how the use case "Ask for help" works

Functionality	Ask for help
Actor	User
Precondition	open duarouter's command line application
Nominal scenario	1. Add the configuration option $" - -help"$; default: false. 2. Add a specific topics (optional) 3. duarouter will print the description of configuration options that related for all or for a selected topics in the commandline output.
Postcondition	A description for a selected configuration options

Table 16. Use case Description of "Ask for help".

4.2 Conceptual design

Before moving on to development, we must go through an essential step which describes how to build the proposed system. Now we will detail this step using UML diagrams.

4.2.1 Static modeling: class diagrams

The class diagram, as described in the figure 24 below, is a static modeling of the system in terms of classes and relationships between these classes. Its interest lies in the modeling of information system entities and it is considered the most important of object-oriented modeling.
Figure 24 is the class diagram of our extented duarouter application. In Figure 17, we have given a brief description for each class.

Class	Description
Global: duarouter_main	main duarouter's entry point to import a different demand definitions, computes vehicle routes that may be used by sumo using shortest path computation
OptionsCont	A storage for options typed value containers. In the case of command line tools, this container is the main interface between a user's definitions about what to do (command line options, configuration files) and the application.
RONet	The router's network representation.
ROEdge	A basic edge's class description for routing applications.
DijkstraRouter	Computes the shortest path through a network using the Dijkstra algorithm..
EdgeInfoByEffortComparator	Class to compare (and so sort) nodes by their effort.
SUMOAbstractRouter	An abstract router base class. The interface for routing the vehicles over the network.
ROVehicule	A vehicle as used by router.
SUMOVTypeParameter	Structure representing possible vehicle parameter.
SUMOVehicleParameter	Structure representing possible vehicle parameter.
RONode	Base class for nodes used by the router.
RORoute	This class represents a single and complete vehicle route after being computed/imported.
RORouteDef	Base class for a vehicle's route definition. This class resembles what a vehicle knows about its route when being loaded into a router (e.g. DijkstraRouter).
ROLoader	The data loader class. It Loads the network and route descriptions using further classes. It Is capable to either load all routes in one step or go through them step wise.
RORoutable	A routable thing such as a vehicle or person.
RODUAFrame	Sets and checks options for dua-routing.
SystemFrame	A set of actions common to all applications.

Table 17. Extented duarouter's class diagram description

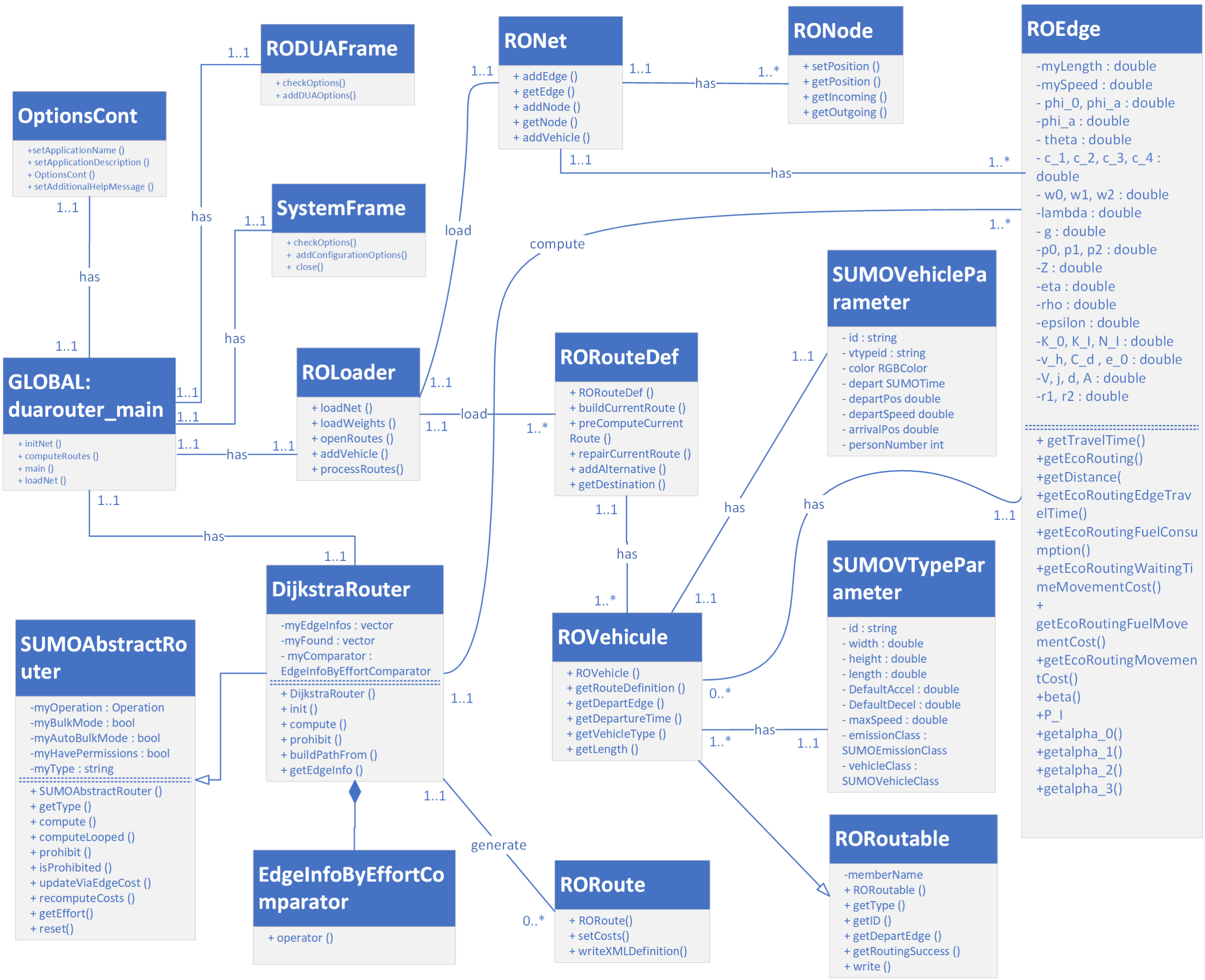


Figure 24. Extented duarouter's class diagram.

4.2.2 dynamic modeling: sequence diagrams

The dynamic view allows us to model the dynamic behavior of our system by indicating how its objects interact at runtime. To do this, and referring to the textual descriptions in the previous section 4.2.1, we present the adequate sequence diagrams of the system.

4.2.3 Sequence diagram for " Load the road network"

Figure 25 is the sequence diagram of "Load the road network".

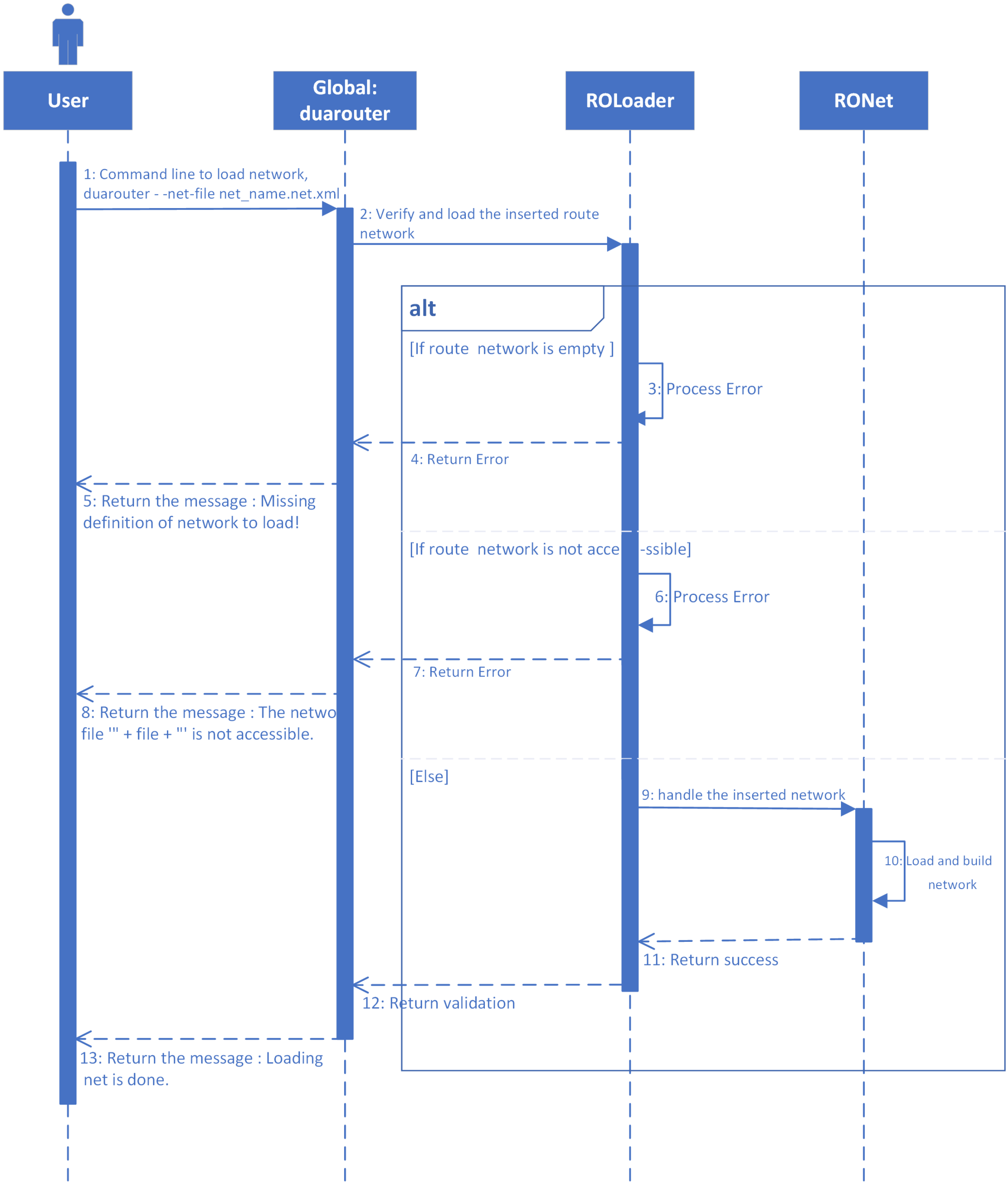


Figure 25. Sequence diagram of "Load the road network".

4.2.4 Sequence diagram for "Load trips"

Figure 26 is the sequence diagram of "Load trips".

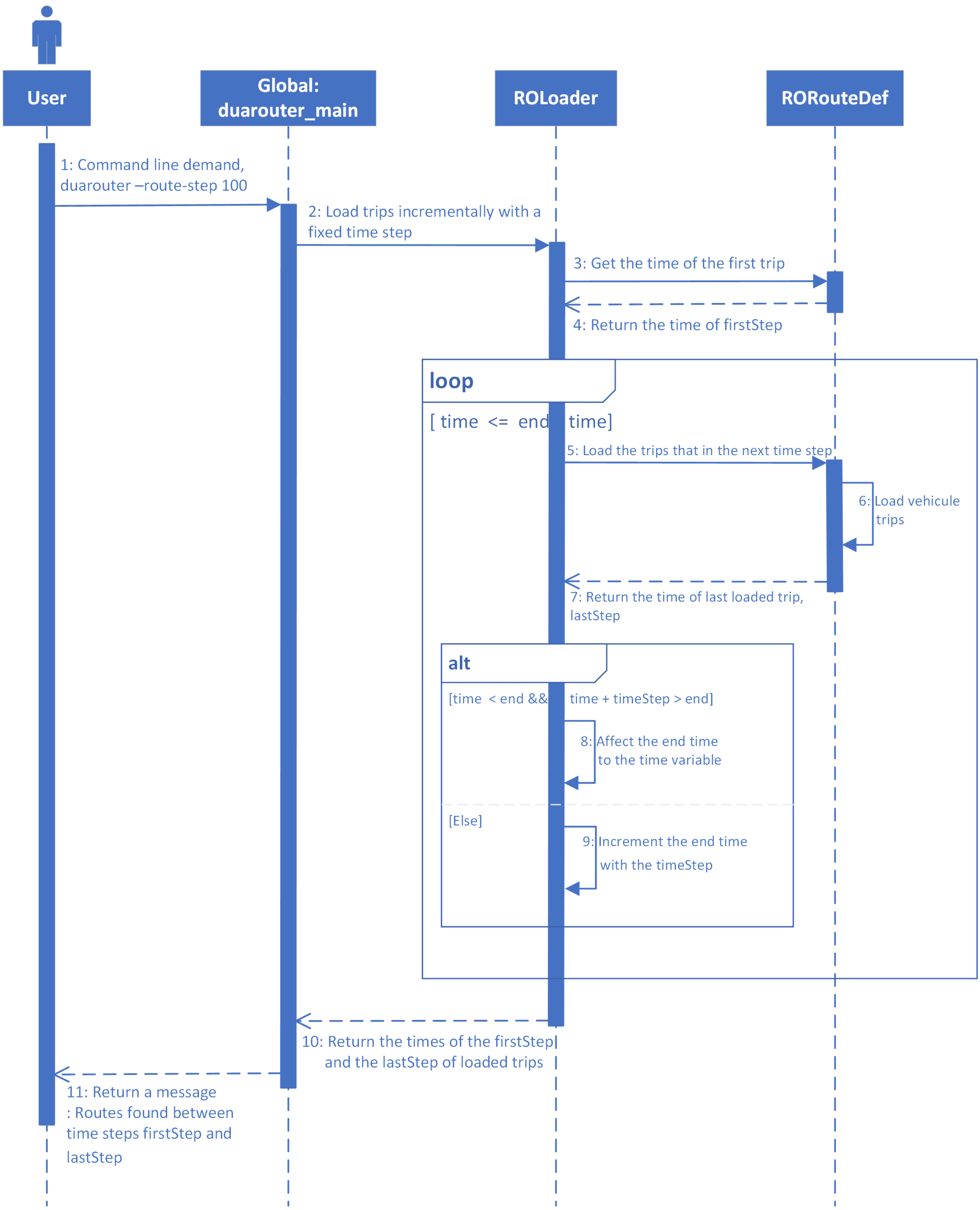


Figure 26. Sequence diagram of "Load trips".

4.2.5 Sequence diagram for "Activation of routing algorithm"

Figure 27 is the sequence diagram of "Activation of routing algorithm".

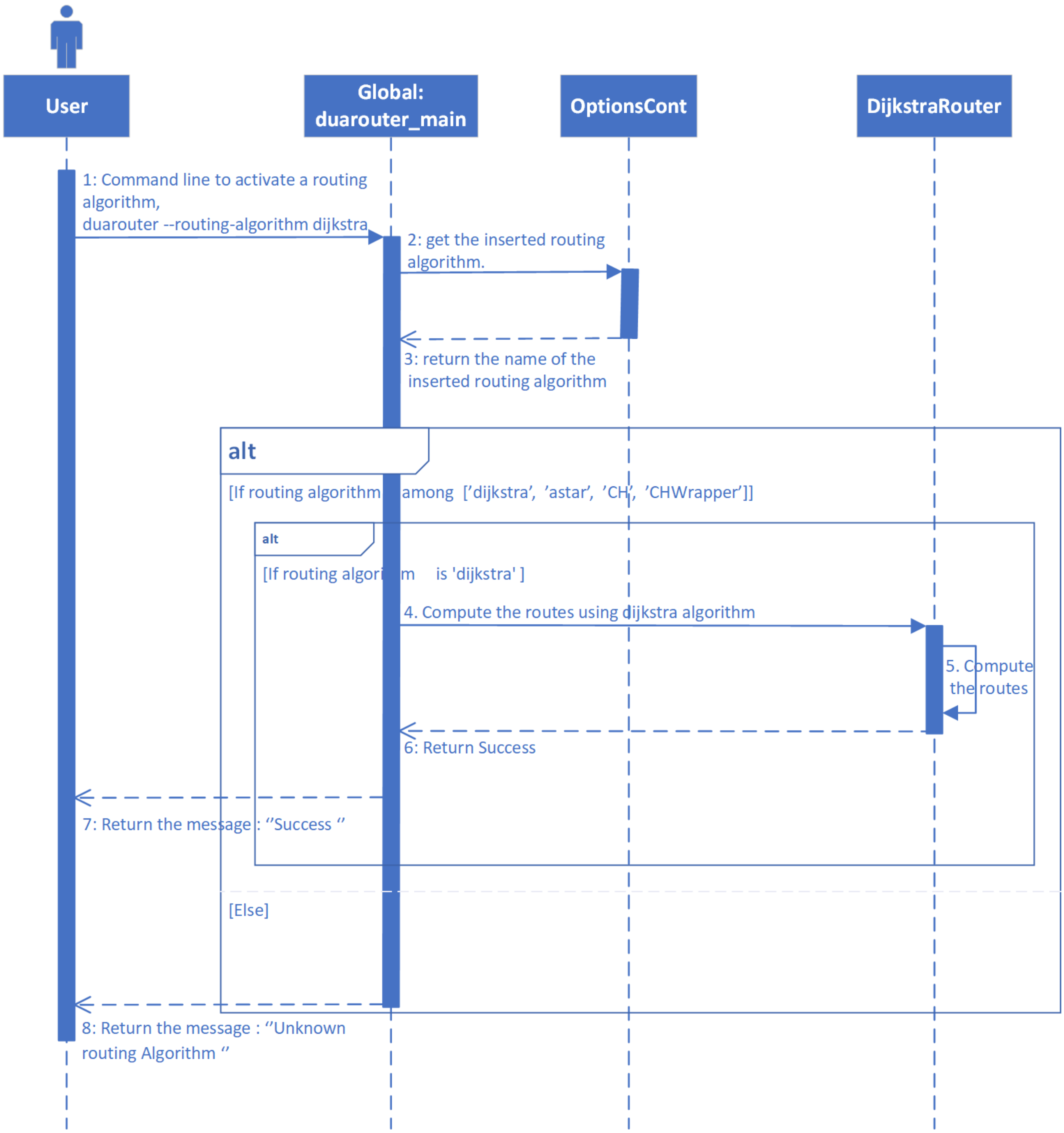


Figure 27. Sequence diagram of "Activation of routing algorithm".

4.2.6 Sequence diagram for "Routing with Ecological model"

Figure 28 is the sequence diagram of "Routing with Ecological model".

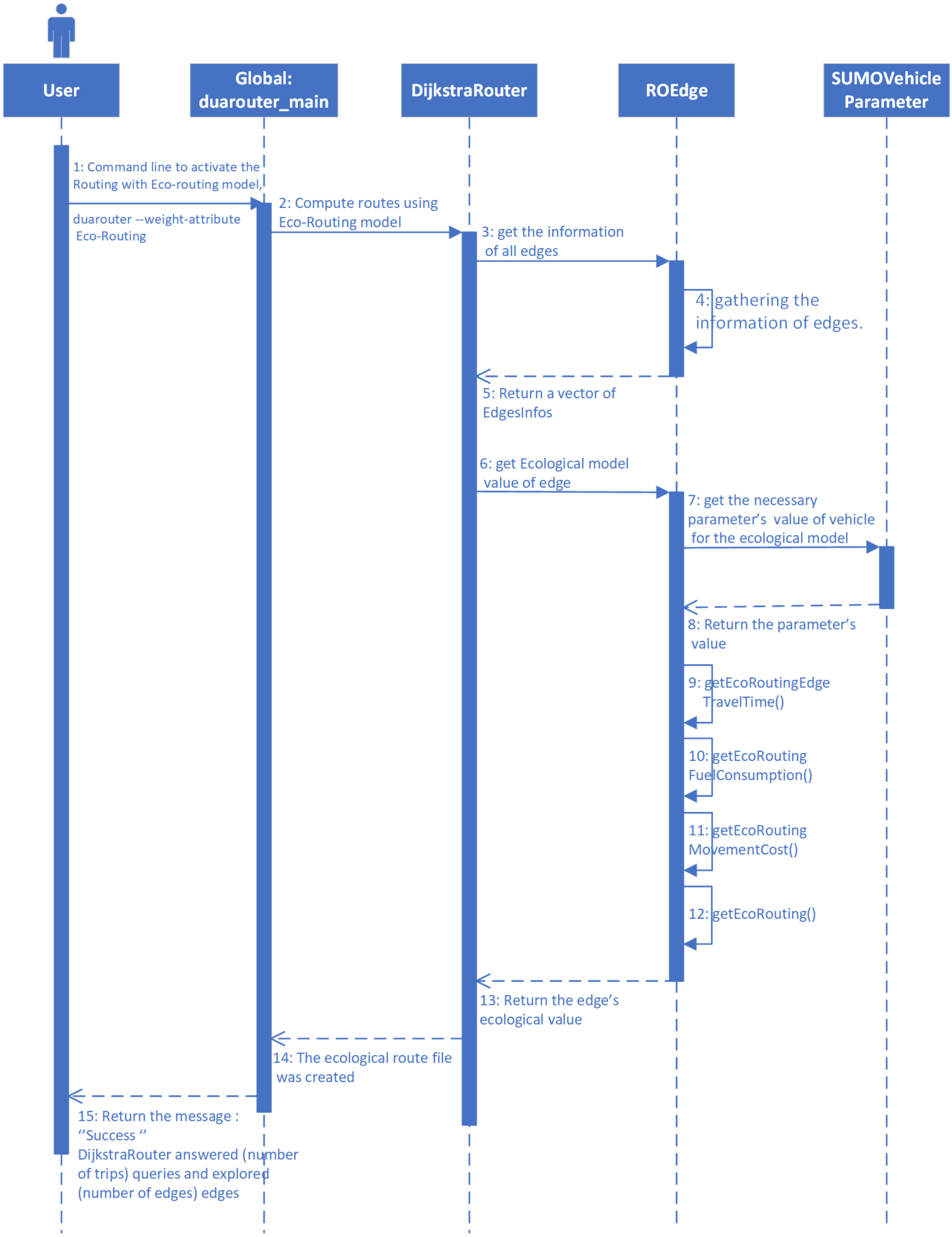


Figure 28. Sequence diagram of "Routing with Ecological model".

4.3 Realisation

After presenting the functionalities and the design of our extended duarouter's application, this chapter is devoted to the implementation and realisation phase of the project.

4.3.1 The Integrated objectives functions and parameters

In this subsection, We will describe the objectives functions and their parameters that used in our integrated Eco-Routing model to calculating each edge's Eco-Routing weight:

The objective function of edge's travel time:

Let t_b denotes the travel time on edge b. We have :

$t_b = \frac{l_b}{v_b};$

where

- l_b = the length of the edge
- v_b = the fixed maximum speed of the edge.

The objective function of fuel consumption:

Let F_b denotes the fuel consumption on edge b. We have :

$F_b = F(v_b, 0)l_b$

where

- $F(v,a) = f/v = \frac{\phi}{\lambda}\left(\sum_{i=0}^{3} \alpha_i v^{i-1} + \beta a\right),$
- $\alpha_0 = \frac{P_a}{\eta},$
- $\alpha_1 = Zg\frac{G+c_1}{\eta\epsilon} + c_4 K_0 V\theta\left(\underline{r} + c_3 v_h^2\right),$
- $\alpha_2 = Zg\frac{c_1}{c_2\eta\epsilon} - 2c_3c_4K_0Vv_h\theta$
- $\alpha_3 = \frac{\rho C_d A}{2\eta\epsilon} + c_3c_4K_0V\theta,$
- $\beta = \frac{Z(1+e_0)}{\eta\epsilon}$
- $c_3 = \frac{\bar{r}-\underline{r}}{v_h^2}$
- $c_4 = 1 + 0.0001(N-33)^2$
- $\theta = \frac{1}{\pi d(1-j)}$

- $P_I = K_I V N_I + \alpha_0$

The objective function of movement cost

The objective function of movement cost is the sum of the waiting time objective function and fuel consumption objective function associated with the movement between two edges.
Let $\hat{t}_m$ denotes the waiting time associated with the movement m. We have:

$\hat{t}_m = \sum_{k=0} w_m^k p_m^k;$

Let $\hat{F}_m$ denotes the fuel consumption associated with the movement m. We have:

$\hat{F}_m = max\left(0, \frac{\phi_a \beta(v_{m+}^2 - v_{m-}^2))}{2\lambda} p_m^0\right) + \sum_{k=1} p_m^k \left(\frac{\phi_a \beta v_{m+}^2}{2\lambda} + p_I \frac{\phi}{\lambda} w_m^k\right);$

Let m denotes the movement between two edges. $\tilde{w}_m$ denotes the waiting time of the movement, modeled as a discrete and independent random variable. $\tilde{w}_m$ can take a discrete set of values $\{w_m^0, w_m^1, ...\}$ with a corresponding probability mass $\{p_m^0, p_m^1, ...\}$. w_m^0 is assumed equal to 0 for any m (Nie and Li, [28]).

The below table, table 18, describes the value parameters that used in calculating the both objectives functions of waiting time and fuel consumption associated with the mouvement m in our integrated model, as it is noticed and assumed in (Nie and Li, [28]).

K discrete variable	Delay (min)	Probabity	Expected Delay (min)
0	0	0.5	0
1	0.6	0.3	0.18
2	1.2	0.2	0.24

Table 18. Description of parameters used in movement cost [28].

Also Table 19 describes the Default value parameters that we have used to calculate the objective function of fuel consumption as it is indicated in (Nie and Li, [28]).

Name	Description	Unit	Default value
Constant parameters			
g	Acceleration of gravity	Nm/s^2	9.81
c_1	Constant	-	0.01
c_2	Constant	m/s	44.73

η	Engine efficiency	-	0.4
Location specific parameters			
ρ	Air density	kg/m^3	1.247
Vehicle specific parameters			
A	Frontal area	m^2	2
C_d	Drag coefficient	-	0.3
e_0	mass factor accounting for the inertia of rotating part	-	0.1
K_0	Constant	J/rev/l	200
K_I	Constant	J/rev/l	300
N_I	Engine speed	rev/s	16.67
V	Engine displacement	l	2.0
j	Drive axle slippage	-	0.04
d	Wheel diameter	m	0.4064
$\bar{r}$	Constant	-	10
$\underline{r}$	Constant	-	2
v_h	Constant	m/s	35
Fuel specific parameters			
λ	Lower heating value	J/g	44.000
Operation specific parameters			
G	Grade	-	0
ϵ	Drivetrain efficiency	-	0.85
ϕ_0	Fuel–air ratio (regular conditions)	-	1.0
ϕ_a	Fuel–air ratio (acceleration)	-	1.13
P_a	Axillary power	W	1000
Z	Mass	kg	1500
Variables			
v	m/s	-	
a	m/s^2	-	

Table 19. Description of parameters used in fuel consumption [28].

4.3.2 Hardware environment

For the realization of our project, we used a Lenovo ThinkPad X1 carbon computer with the following technical characteristics as a hardware resource:

- Processor : Intel®Core i7-5600U CPU @ 2.6 GHZ
- 8 GO RAM
- Graphic card Intel®HD Graphics 5500
- 64-bit system
- Hard driver 256 GB
- Operating system : Windows 10

4.3.3 Software environment

In table 20, we have represented the software that we have used for completing our project.

Software	Description	Logo
Visual Studio Community 2019	Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio supports 36 different programming languages and allows the code editor and debugger to support nearly any programming language.	
CMake	CMake is a cross-platform free and open-source software tool for managing the build process of software using a compiler-independent method. CMake is not a build system but rather it's a build-system generator.	

Visio 2019	Microsoft Visio is a diagramming solution with a robust library of templates and shapes, data connectivity capabilities allowing users to create flowcharts, organizational charts, floor plans, network diagrams, mind maps, infographics and more. Visio is part of the Microsoft Office software suite, although it is sold as a stand-alone program.	
Jira software	Jira Software edition is a project management system to plan, track, and release great software, developed by Atlassian.	
Notion	Notion Labs Inc, a startup based in San Francisco, was founded in 2016. Notion company claims to be an all-in-one workspace for note-taking, knowledge and data management, project and task management. The company's tool helps users to schedule tasks, manage files, save documents and set a reminder for the weekly agenda, enabling users to organize their work. An academic email address allows for a free personal plan.	
LaTeX	LaTeX is a high-quality document preparation system; it includes features designed for the production of technical and scientific documentation. LaTeX is the de facto standard for the communication and publication of scientific documents. LaTeX is available as free software.	

Skype	Skype is a telecommunications application. It allows users to communicate over the Internet by voice, using a microphone, by video using a webcam, and by instant messaging. In March 2020, Skype was used by 40 million people on a daily basis. That was a 70% increase in the number of daily users from the previous month, due to the COVID-19 pandemic.	

Table 20. Description of software

4.3.4 Technologies used

C++

C++ is a middle-level programming language developed by Bjarne Stroustrup starting in 1979 at Bell Labs. C++ runs on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX. C++ is a powerful general-purpose programming language. It can be used to develop operating systems, browsers, games, and so on. C++ supports different ways of programming like procedural, object-oriented, functional, and so on. This makes C++ powerful as well as flexible.

Python

Python is an interpreted, high-level and general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Conclusion

Throughout this chapter, we have described the different requirement analysis and specification for our duarouter's extend implementation. Then, we designed a static and a dynamic view of our proposed system using UML diagrams. And we ended up by rehighlighting the implemented functions and parameters in our integrated eco-routing model, and giving the technology stack including hardware and software.
In the next chapter, we will validate the implementation through simulation of scenarios using real maps and the generation of evaluation metrics including average time, average distance and estimated average consumption for the simulated trips.

5. Simulations and results

Contents

5.1 Simulation scenario **88**
5.2 Evaluation **98**

Introduction

In this chapter, we validate the implementation through simulation of scenarios. First, we create a real grand tunis SUMO-network map. Then, we generate a 600 random trips. Next, we create three variant of optimal routes : optimal ecological-based routes, optimal time-based routes and optimal distance-based routes. Finally, We perform a simulation for each routing model, and we get a set of average values for all vehicle trips (i.e. average travel distance, average travel time, average travel fuel consumption).

5.1 Simulation scenario

To create our simulation scenario with a real word map:

- First step, we need to prepare our grand tunis road network and to generate a set of randomly trips, see the flowchart of our road network and trips generation process Figure 32.
  We need to download a map file from the open street map site[32] with a file name extension (.osm). We boundaries a grand tunis surface and then we export the map. As a notice, OpenStreetMap is a free, editable map of the whole world that is being built by volunteers largely from scratch and released with an open-content license. The following image Figure 29, is our OSM grand tunis street map views in 5km and 100m distance granularity. The resulting download is a map file with a different roads information and a name extension (.osm.pbf). To convert our map file to .osm extension, we just have to use a Osmconvert application [33].
  Now, after we have our osm map file, we must convert it to SUMO road network format with file name extension (.net.xml). We have to use the SUMO netconvert application, see the subsection 2.6 for more information about netconvert application. As input, we have the osm file and the additional information SUMO file osmNetconvert.typ.xml [34], this file is a default SUMO's edge type file for importing OpenStreetMap files. It contains the edge type configuration for roads to give more accuracy in our road network definition, See the flowchart Figure 32 for the command line to generate our grand-tunis-map.net.xml file.
  To import a wide range of additional information for our road network such as building, rivers, green spaces, etc... We need to perform one more operation called polyconvert. To apply polyconvert operation, we need a SUMO's (osmPolyconvert.typ.xml) file [35] as input to polyconvert SUMO application, also osm file(map-grand-tunis.osm) and sumo network file(grand-tunis-map.net.xml). The resulting is a Shapes of polygons definition

file (map-grand-tunis.poly.xml). Figure 30 is how our road network it looks in netedit SUMO application, see the subsection 2.7 for more information about netedit application. Now, after we have prepared our traffic network, we must generate our traffic demand. We are going to generate a set of traffic demand with random starting points and destinations for this purpose we are going to use randomtrip.py tool, see subsection 2.11 for more information about randomtrip.py tool.
In our simulation we have used a passenger class car with a set of SUMO recommended characteristic, to have more accuracy and precision for our simulation outputs.
the following Table 21 is the different characteristics of our used passenger type car. This characteristics are inserted to randomtrip.py tool as an additional input file(type.add.xml) with our SUMO road network file (map-grand-tunis.net.xml), See the flowchart Figure 33 for the command line to generate our 600-trips.trips.xml file which consist of 600 randmly generated trips.

- Next step, we need to create a routes between those different trips, so we are going to use duarouter application of SUMO package, see the subsection 2.13.2 for more information about duarouter, to find the optimal route using the computation of the shortest-path, see the flowchart of our routes generation process Figure 33.
  To validate our Eco-routing model we will use three variant of optimal routing : optimal ecological-based routing, optimal time-based routing and optimal distance-based routing. Noting also that, we will not use the PRIORITYFACTOR option for routing in our scenario. By activating the travel time routing model, duarouter will compute travel time routes from the maximum velocity allowed on the edge and the edge's length, See the flowchart Figure 33 for the command line activation.
  When activated our implemented distance routing model, duarouter will find the routes that have the lowest length, See the flowchart Figure 33 for the command line activation. Same for the case of our implemented eco-routing model, duarouter will return the optimal ecological routes taking into consideration the energy consumption, travel time and emission rate factors, See the flowchart Figure 33 for the command line activation.

- Finally, We will perform a simulation for each routing model, so we will perform three simulation one with the generated travel time routes, second with the generated travel distance routes, and the last one with the generated travel ecological routes.
  To complete the simulation we can use sumo application or sumo-gui application that just extended by a graphical user interface, see the subsection 2.12 for more information about sumo and sumo-gui application.

  Sumo/sumo-gui application needs a configuration file to simulate and to generate outputs.

We have created for each kind of simulation a unique configuration file. All configuration files have the same road network file(map-grand-tunis.net.xml) and the same polygons additional definition file (map-grand-tunis.poly.xml). For simulation time we indicate that the simulation computation start in $0s$ and sumo will end its simulation when all vehicles in our scenario have been simulated 2.12. Figure 31 is a screenshot for our simulation scenario with sumo-gui application 2.12.
Listing 5.1 referred to Sumo/sumo-gui Configuration file: (simulation-time-routing-based.sumo.cfg.xml). This Configuration file adds as input a route file based travel time (route-time-based.rou.xml).
When we specify -DURATION-LOG.STATISTICS option while simulation, See the command line in the flowchart Figure 34, a set of different average values for all vehicle trips are generated, see the subsection 2.12 for more information about sumo and sumo-gui outputs. As values we have average travel time and average travel distance, and to have the average value of the travel fuel consumption we simply specify in sumo configuration file the parameter DEVICE.EMISSIONS.PROBABILITY "1", this parameter will add an emission's factors of outputs for each vehicle trip into our trip-info file's output, then we calculate the average value for all the trips that are simulated.
Listing 5.2 referred to Sumo/sumo-gui Configuration file: (simulation-distance-routing-based.sumo.cfg.xml). It adds a route file based travel distance (route-distance-based.rou.xml) as input. When we activate the command line, See the flowchart Figure 35, a set of distance based output simulation are generated.
Listing 5.3 referred to Sumo/sumo-gui Configuration file: (simulation-eco-routing-based.sumo.cfg.xml). This Configuration file adds the route file based travel ecological (route-ecological-based.rou.xml) as input. When we activate the command line, See the flowchart Figure 36, a set of ecological based output simulation files are generated:

- (my-network-dump-time-based.xml) file output contains detailed information for each edge, each vehicle and each simulation step.
- (my-trip-info-time-based.xml) file output that contains the information about each vehicle's trip departure time, the time the vehicle wanted to start at (which may be lower than the real departure time) and the time the vehicle has arrived, to add into this file the sum of all emissions/consumption generated/consumed by the vehicle during its journey, we must instantiate the option $< device.emissions.probability$ $value = "1" >$, see sumo/sumo-gui ecological Configuration file in Listing 5.3.
- (my-summary-time-based.xml) file output contains contains the simulation-wide number of vehicles that are loaded, inserted, running, waiting to be inserted, have reached their destination and how long they needed to finish the route.

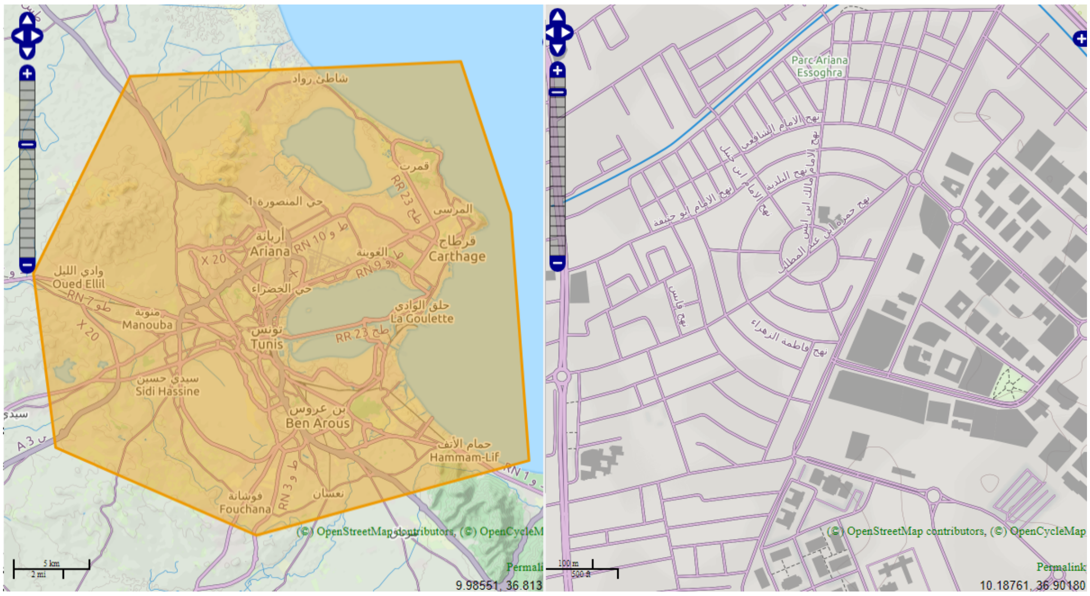


Figure 29. Our OSM grand tunis street map boundries and views.

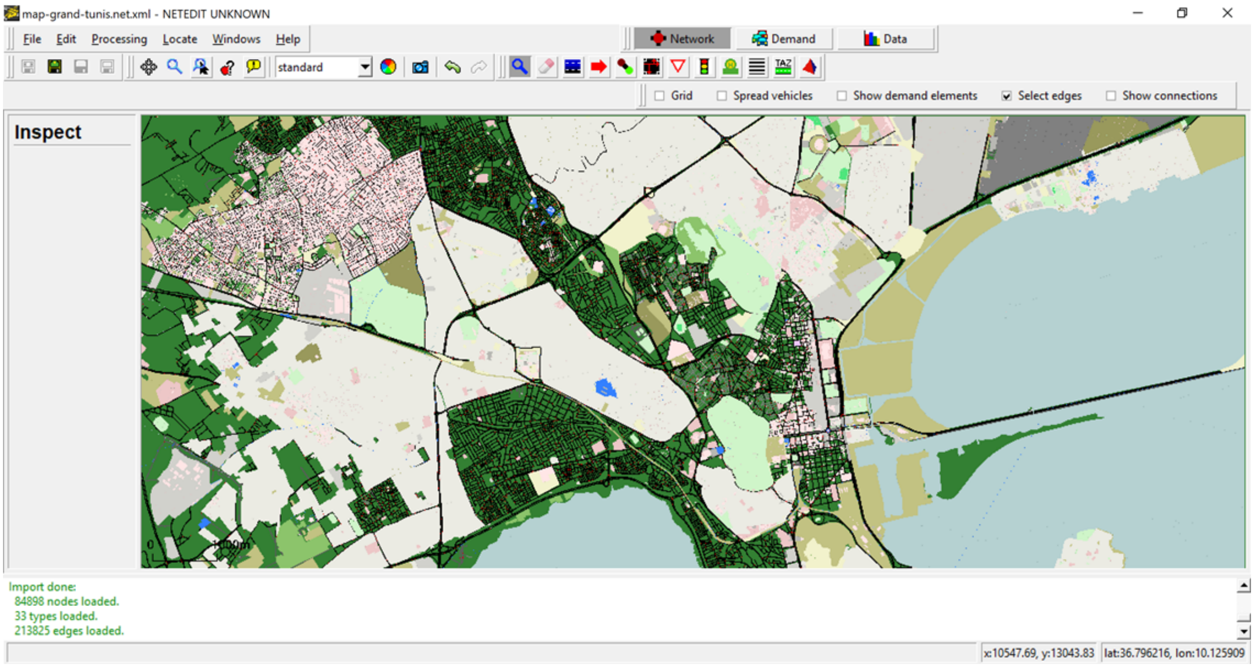


Figure 30. Our road network and additional information opened in netedit.

characteristic	**Type**	**Value**	**Remark**
Identification	String	Passenger car	The name of the vehicle type
Vehicle class	class ($enum$)	Passenger	normal passenger cars
accel	float(m/s^2)	2.6	-
decel	float(m/s^2)	4.5	-
sigma	float	0.5	-
length	float(m)	5	The physical length of the vehicle
minGap	float(m)	2.5	The minimum gap between this vehicle and the vehicle before it
maxSpeed	float(m/s)	70(70*3.6 = 252 km/h)	The vehicle's maximum velocity (in m/s)
emission Class	emission class ($enum$)	"PC_G_EU4"	a gasoline passenger car conforming to emission standard EURO 4 is used
guiShape	shape ($enum$)	-	a standard passenger car body is drawn
width	float(m)	1.8	The physical width of the vehicle
height	float(m)	1.5	The vehicle's height
lane Change Model	lane changing model name ($string$)	'LC2013'	The model used for changing lanes
car Follow Model	car following model name ($string$)	'Krauss'	The model used for car following
person Capacity	int	4	The number of persons that the vehicle can transport.
container Capacity	int	0	The number of containers that the vehicle can transport.
boarding Duration	$float$	0.5	The time required by a person to board the vehicle.

Table 21. Our used passenger type car's characteristics

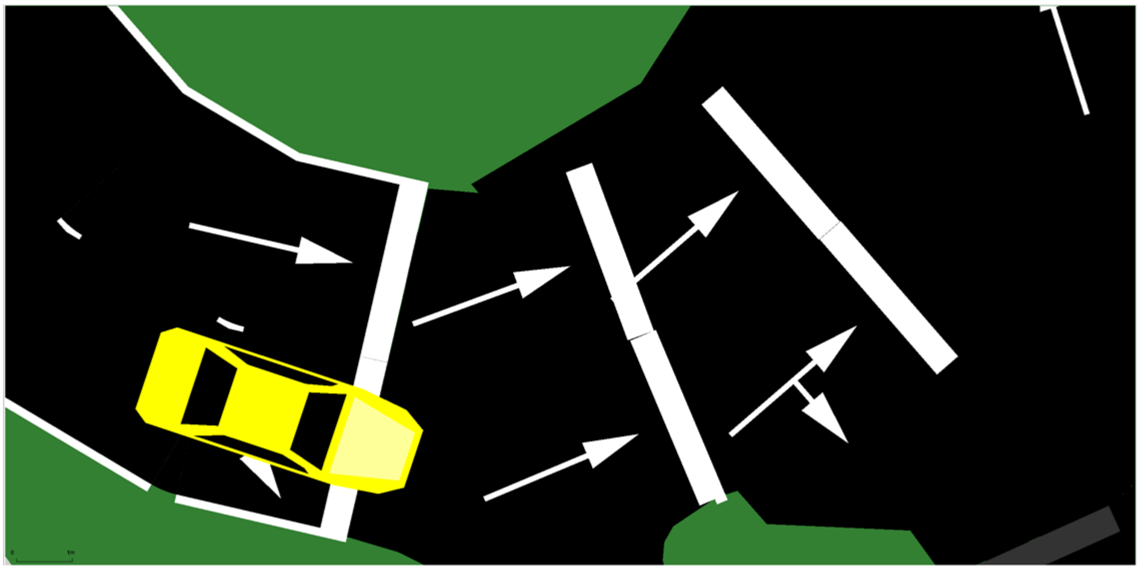


Figure 31. Sumo-gui with our scenario of simulation.

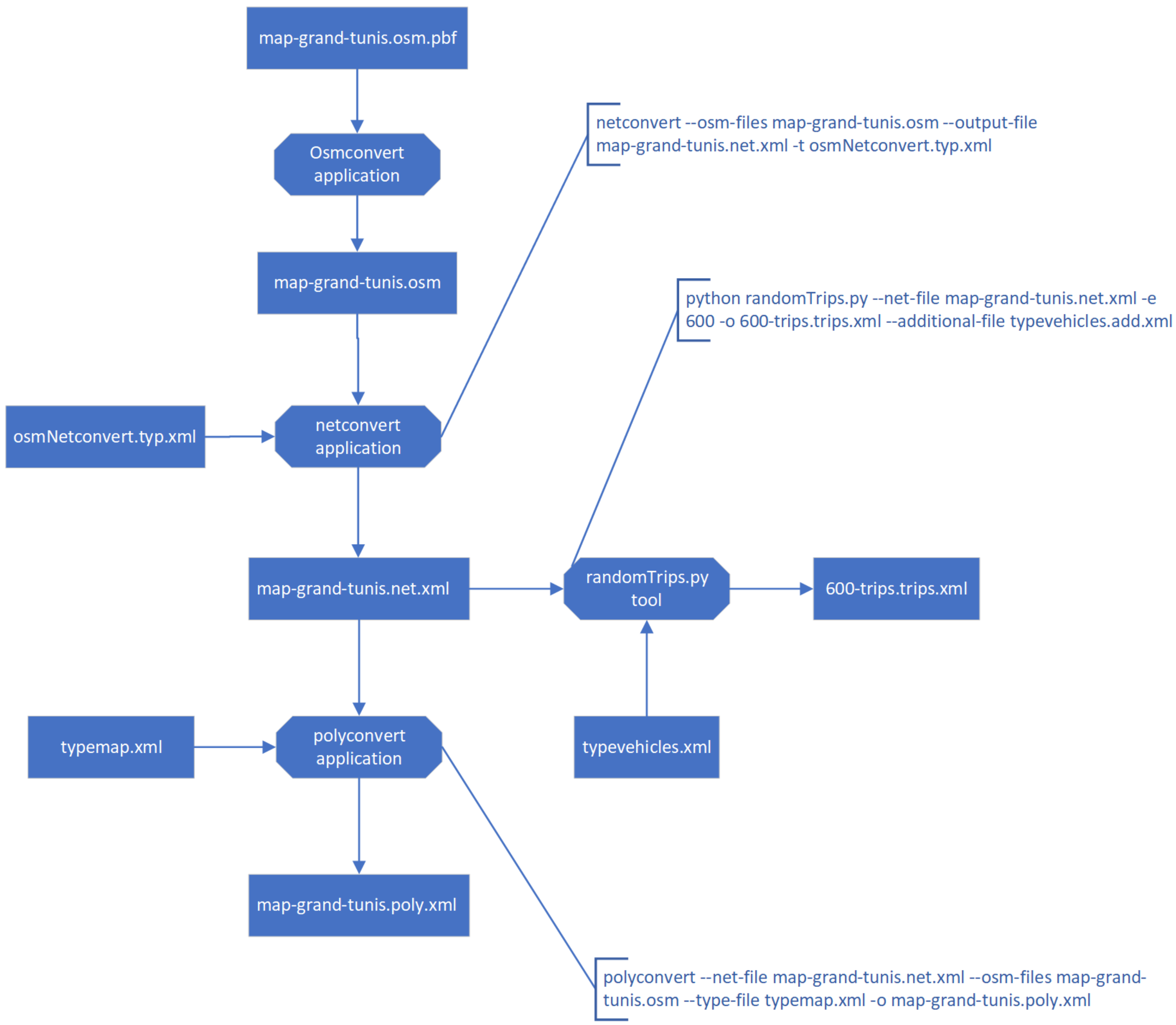


Figure 32. Flowchart of our road network and trips generation process.

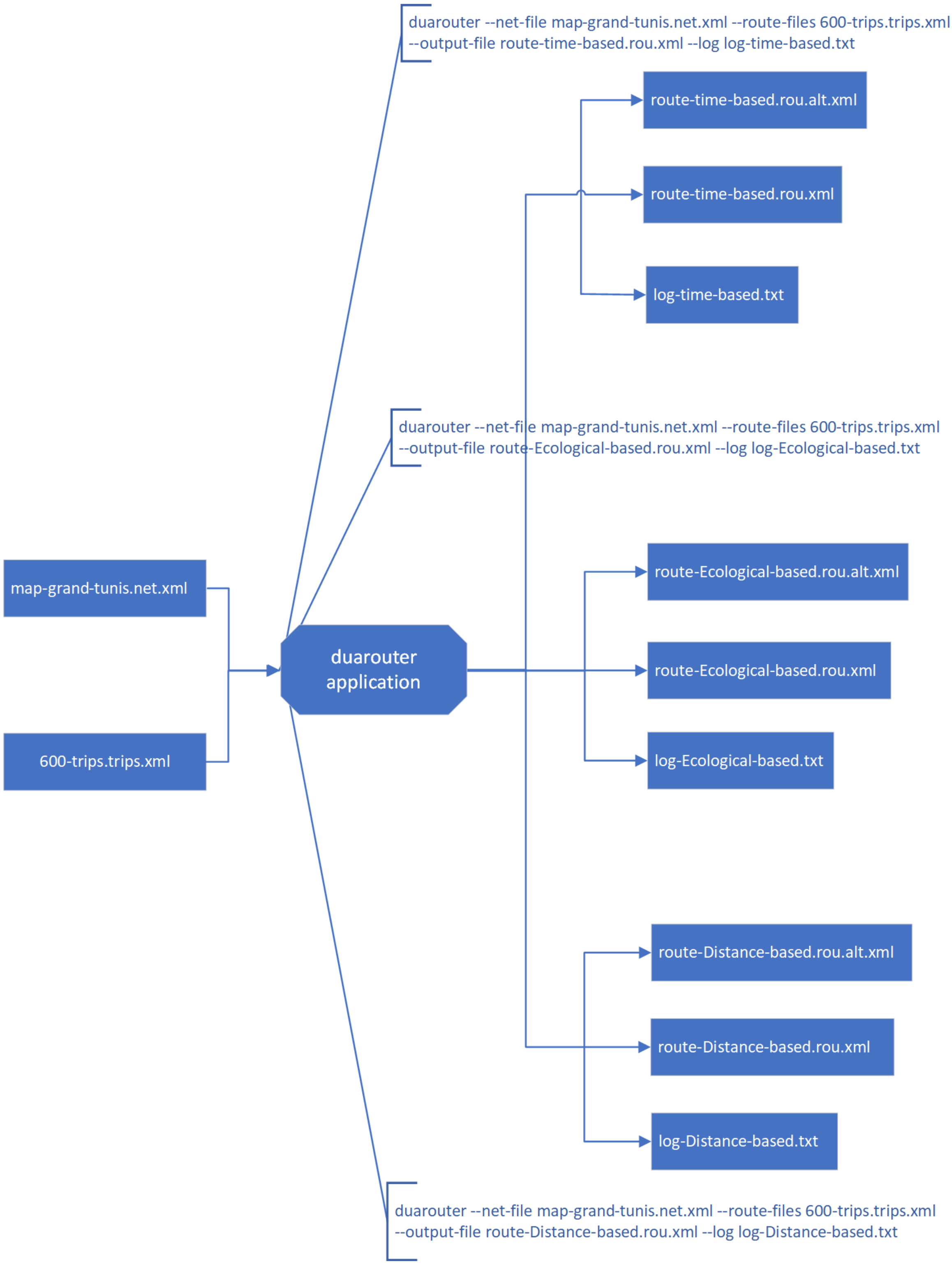


Figure 33. Flowchart of our routes generation process.

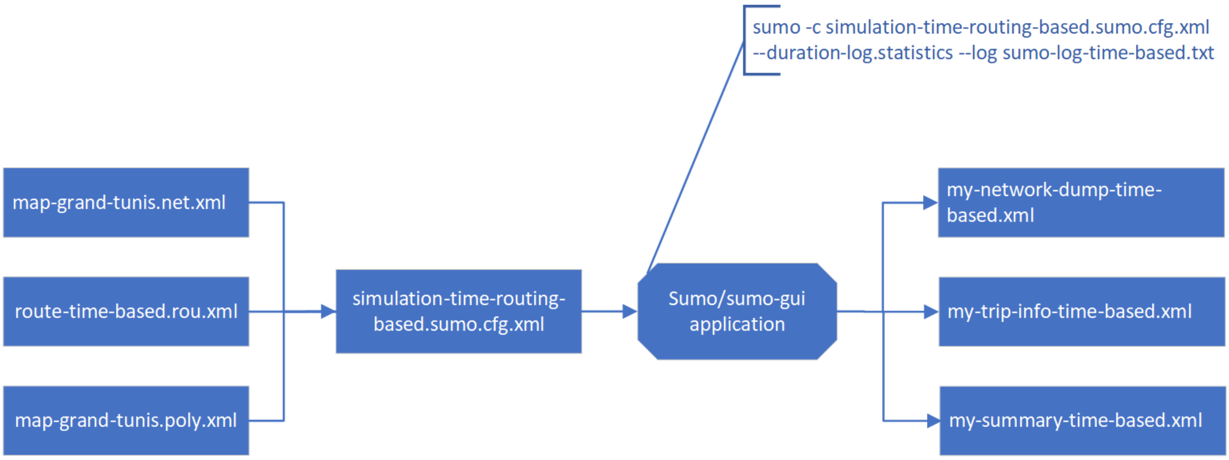


Figure 34. Flowchart of our simulation with travel time routing process.

```xml
<configuration>
    <input>
        <net-file value="map-grand-tunis.net.xml"/>
        <route-files value="route-time-based.rou.xml"/>
        <additional-files value="map-grand-tunis.poly.xml"/>
    </input>
    <output>
        <netstate-dump value="my-network-dump-time-based.xml"/>
        <tripinfo-output value="my-trip-info-time-based.xml"/>
        <summary value="my-summary-time-based.xml"/>
        <device.emissions.probability value="1"/>
    </output>
    <time>
        <begin value="0"/>
    </time>
</configuration>
```

Listing 5.1. Sumo/sumo-gui Configuration file: "simulation-time-routing-based.sumo.cfg.xml"

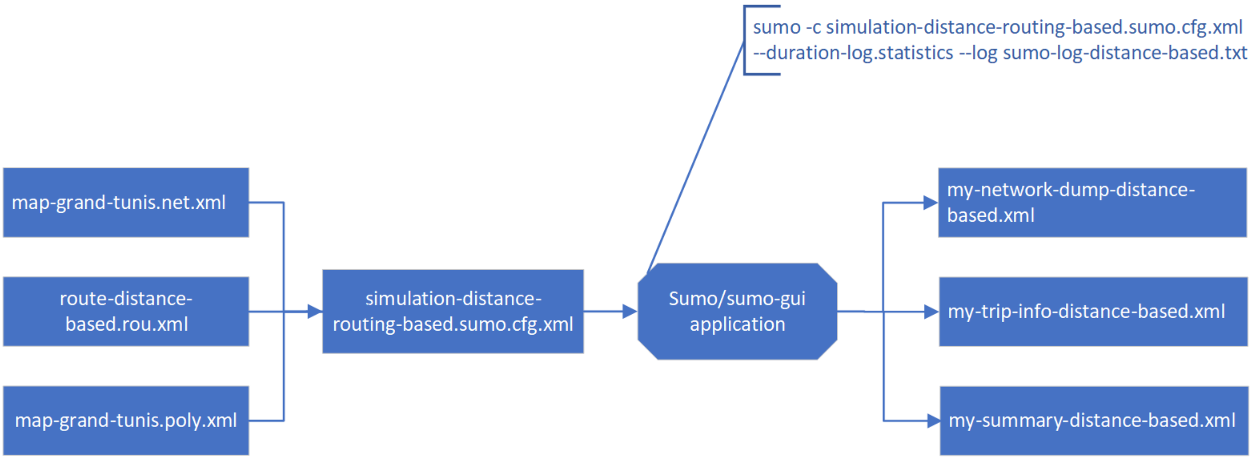


Figure 35. Flowchart of our simulation with distance routing process.

```xml
<configuration>
    <input>
        <net-file value="map-grand-tunis.net.xml"/>
        <route-files value="route-distance-based.rou.xml"/>
        <additional-files value="map-grand-tunis.poly.xml"/>
    </input>
    <output>
        <netstate-dump value="my-network-dump-distance-based.xml"/>
        <tripinfo-output value="my-trip-info-distance-based.xml"/>
        <summary value="my-summary-distance-based.xml"/>
        <device.emissions.probability value="1"/>
    </output>
    <time>
        <begin value="0"/>
    </time>
</configuration>
```

Listing 5.2. Sumo/sumo-gui Configuration file: "simulation-distance-routing-based.sumo.cfg.xml"

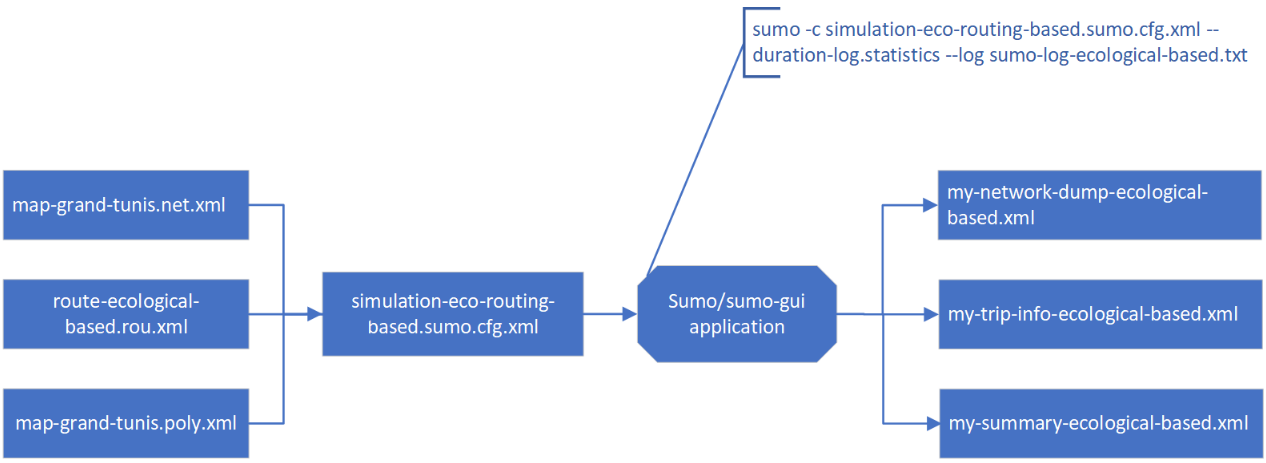


Figure 36. Flowchart of our simulation with eco-routing process.

```xml
<configuration>
    <input>
        <net-file value="map-grand-tunis.net.xml"/>
        <route-files value="route-ecological-based.rou.xml"/>
        <additional-files value="map-grand-tunis.poly.xml"/>
    </input>
    <output>
        <netstate-dump value="my-network-dump-ecological-based.xml"/>
        <tripinfo-output value="my-trip-info-ecological-based.xml"/>
        <summary value="my-summary-ecological-based.xml"/>
        <device.emissions.probability value="1"/>
    </output>
    <time>
        <begin value="0"/>
    </time>
</configuration>

```

Listing 5.3. Sumo/sumo-gui Configuration file: simulation-eco-routing-based.sumo.cfg.xml

5.2 Evaluation

The table 22 below presents for each routing model (i.e. Distance-based routing model, Time-based routing model and Ecological-based routing model), its three corresponding average cost metric values for all trips, we mean by that : the value of average travel distance in meter, the value of Average travel time in second and the value of Average travel fuel consumption in milliliter.

As we can see, and in light of the results we obtained. We can come up with these observations :

- The distance-based routing model and the time-based routing model, they still get the best average travel distance value and the best average travel time value, respectively.
- For a travel based on optimal distance route, and for a travel based on optimal time route, we highly recommend using of the distance-based routing model and the time-based routing model, respectively. Due to the huge difference between the cost of each model and the cost of the two other models.
- For a travel based on optimal fuel consumption route, the distance-based routing model is in the top but only with a slight edge.
- The ecological-based routing model returns awful results compared to the other two models.
- According to the results that we obtained, we can nominate the distance-based routing model as the best model, considering the balance between all our approved metrics, and also where it exceeds in all metrics values except for the time metric.
- In this present project, the ecological-based routing model returns awful results almost for all our approved metrics, and we shall improve these results, in particular the fuel and energy consumption, using other eco-routing techniques during our next project.

Model / average cost metric	**Average travel distance(m)**	**Average travel time(s)**	**Average travel fuel consumption(ml)**
Distance-based routing model	15092.21	875.49	1517,455
Time-based routing model	16279.58	740.23	1534,74
Ecological-based routing model	17098.82	1167.18	1795,35

Table 22. SUMO routing models and the average cost metric values for all simulated trips

Conclusion

During this chapter, we have validated the implementation through simulation of scenarios using real maps and the generation of evaluation metrics including average time, average distance and estimated average fuel consumption for the simulated trips.
According to the results that we obtained, the distance-based routing model is the best model, considering the balance between all our approved metrics, and also where it exceeds in all metrics values except for the time metric.
In this present project, the ecological-based routing model returns awful results. We shall improve these results, in particular the fuel and energy consumption, using other eco-routing techniques during our next project.

General Conclusion

Transportation plays an important role in climate change. Transportation consumes over 27.8% of all energy, of which 92.8% comes from petroleum use; Transportation also contributes roughly 32.7% of total greenhouse gas (GHG) emissions [5].

In the last two decades , and because of increasingly severe environmental issues, the environmental and economical impacts of the transportation sector have attracted the attention of scholarly communities, who devoted much research effort towards sustainable mobility, to save fuel consumption and emissions.

Eco-routing solutions will address precisely the above issues. It is a vehicle navigation method that chooses those paths that minimize fuel consumption, energy consumption or pollutant emissions for a trip to given destination. Also, the cost estimation of road segment is key to route planning. This estimation gives wide range of impact, including monetary costs, state of the traffic in the road network, safety and pollution and climate change.

Simulation in transportation is important because it can study models too complicated for analytical or numerical treatment, can be used for experimental studies, can study detailed relations that might be lost in analytical or numerical treatment and can produce attractive visual demonstrations of present and future scenarios.

In this contexe, this project aims to implement an eco-routing techniques in the SUMO (Simulation for Urban MObility).

We have picked the Nie and Li, 2013 eco-routing model [28]. This publication presents an eco-routing method that aims to find a path that minimizes the cost value of the fuel combined with travel time while meeting a given CO_2 constraint. Then, we validated the implementation through simulation of scenarios using a grand-tunis real map and the generation of evaluation metrics including average time, average distance and estimated average fuel consumption for the 600 simulated trips.

In this present project, the ecological-based routing model returns awful results. We shall improve these results, in particular the fuel and energy consumption, using other eco-routing techniques during our next project, as part of my master's end-of-study project.

During this 6-month internship, I was able to put into practice my theoretical knowledge acquired during my training at TEK-UP private university, while being confronted with the real difficulties of the working world.

Bibliography

[1] Jan Fuglestvedt et al. “Climate forcing from the transport sectors”. In: *Proceedings of the National Academy of Sciences of the United States of America* 105 (Feb. 2008), pp. 454–8. DOI: 10.1073/pnas.0702958104.

[2] Ole Raaschou-Nielsen. “Air Pollution and Lung Cancer Incidence in 17 European Cohorts: Prospective Analyses from the European Study of Cohorts for Air Pollution Effects (Escape)”. In: *The lancet oncology* 14 (July 2013). DOI: 10.1016/S1470-2045(13)70279-1.

[3] Pablo Alvarez Lopez et al. “Microscopic Traffic Simulation using SUMO”. In: *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018. URL: https://elib.dlr.de/124092/.

[4] *Riadi Laboratory-Tunisia)*. URL: http://www.riadi.rnu.tn/.

[5] Stacy Cagle Davis, Susan W Diegel, and Robert Gary Boundy. “Transportation Energy Data Book: Edition 31”. In: (Aug. 2012). DOI: 10.2172/1049830.

[6] S.K. Zegeye et al. “Model-based traffic control for balanced reduction of fuel consumption, emissions, and travel time”. In: *IFAC Proceedings Volumes* 42 (July 2010).

[7] Yi-Chang Chiu et al. “Dynamic Traffic Assignment: A Primer”. In: *Transportation Research E-Circular* (June 2011).

[8] Tolga Bektas and Gilbert Laporte. “The Pollution-Routing Problem”. In: *Transportation Research Part B: Methodological* 45 (Sept. 2011), pp. 1232–1250. DOI: 10.1016/j.trb.2011.02.004.

[9] Liya Guo, Shan Huang, and Adel W. Sadek. “An Evaluation of Environmental Benefits of Time-Dependent Green Routing in the Greater Buffalo–Niagara Region”. In: *Journal of Intelligent Transportation Systems* 17.1 (2013), pp. 18–30. DOI: 10.1080/15472450.2012.704336. eprint: https://doi.org/10.1080/15472450.2012.704336. URL: https://doi.org/10.1080/15472450.2012.704336.

[10] C S Papacostas; P D Prevedouros. “Transportation engineering and planning”. In: *Englewood Cliffs (New Jersey) : Prentice Hall* Print book : English : 2nd ed (1993).

[11] Yi-Chang Chiu et al. “Dynamic Traffic Assignment: A Primer”. In: *Transportation Research E-Circular* (June 2011).

[12] David Watling and Martin Hazelton. "The Dynamics and Equilibria of Day-to-Day Assignment Models". In: *Networks and Spatial Economics* 3 (Sept. 2003), pp. 349–370. DOI: 10.1023/A:1025398302560.

[13] Lihua Luo et al. "Real-time route diversion control in a model predictive control framework with multiple objectives: Traffic efficiency, emission reduction and fuel economy". In: *Transportation Research Part D Transport and Environment* 48 (Aug. 2016), pp. 332–356. DOI: 10.1016/j.trd.2016.08.013.

[14] F.S. Zuurbier. "Intelligent Route Guidance". In: *TRAIL* (Nov. 2010).

[15] M. Barth, K. Boriboonsomsin, and A. Vu. "Environmentally-Friendly Navigation". In: *2007 IEEE Intelligent Transportation Systems Conference*. 2007, pp. 684–689.

[16] G. De Nunzio, L. Thibault, and A. Sciarretta. "A model-based eco-routing strategy for electric vehicles in large urban networks". In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. 2016, pp. 2301–2306.

[17] Enjian Yao and Yuanyuan Song. "Study on Eco-Route Planning Algorithm and Environmental Impact Assessment". In: *Journal of Intelligent Transportation Systems* 17.1 (2013), pp. 42–53. DOI: 10.1080/15472450.2013.747822. eprint: https://doi.org/10.1080/15472450.2013.747822. URL: https://doi.org/10.1080/15472450.2013.747822.

[18] José Luis Jiménez-Palacios. "Understanding and quantifying motor vehicle emissions with vehicle specific power and TILDAS remote sensing". In: 1999.

[19] Michael Behrisch et al. "SUMO – Simulation of Urban MObility: An Overview". In: vol. 2011. Oct. 2011. ISBN: 978-1-61208-169-4.

[20] "the structure of SUMO with the doxygen documentation". In: URL: https://sumo.dlr.de/daily/doxygen/.

[21] Teron Nguyen et al. "DFROUTER—Estimation of Vehicle Routes from Cross-Section Measurements". In: *Lecture Notes in Control and Information Sciences* 13 (Jan. 2015), pp. 3–23. DOI: 10.1007/978-3-319-15024-6_1.

[22] INFRAS. *HBEFA, Handbook emission factors for road transport 4.1*. https://www.hbefa.net/e/help/HBEFA41_help_en.pdf. 2019.

[23] Krishna Saw, B. K. Katti, and G. Joshi. "Literature Review of Traffic Assignment: Static and Dynamic". In: *International Journal of Transportation Engineering* 2.4 (2015), pp. 339–347. ISSN: 2322-259X. DOI: 10.22119/ijte.2015.10447.

[24] "Dynamic traffic assignment: A review of the methodological advances for environmentally sustainable road transportation applications". In: *Transportation Research Part B: Methodological* 111 (2018), pp. 370–394. ISSN: 0191-2615. DOI: https://doi.org/10.1016/j.trb.2018.03.011. URL: http://www.sciencedirect.com/science/article/pii/S0191261517308056.

[25] M. Fu, J. Li, and Z. Deng. "A practical route planning algorithm for vehicle navigation system". In: *Fifth World Congress on Intelligent Control and Automation (IEEE Cat. No.04EX788)* 6 (2004), 5326–5329 Vol.6.

[26] F. Zhan and C. E. Noon. "Shortest Path Algorithms: An Evaluation Using Real Road Networks". In: *Transp. Sci.* 32 (1998), pp. 65–73.

[27] Michael Behrisch, Daniel Krajzewicz, and Yun-Pang Flötteröd. "Comparing performance and quality of traffic assignment techniques for microscopic road traffic simulations". In: Jan. 2010.

[28] Yu Nie and Qianfei Li. "An eco-routing model considering microscopic vehicle operating conditions". In: *Transportation Research Part B: Methodological* 55 (Sept. 2013), pp. 154–170. DOI: 10.1016/j.trb.2013.06.004.

[29] M. Barth et al. "Development of a Comprehensive Modal Emissions Model". In: 2000.

[30] E. K. Nam. "Proof of Concept Investigation for the Physical Emission Rate Estimator (PERE) to be Used in MOVES". In: 2003.

[31] Matt Carlyle, Johannes Royset, and R. Wood. "Lagrangian Relaxation and Enumeration for Solving Constrained". In: *Networks* 52 (Dec. 2008), pp. 256–270. DOI: 10.1002/net.20247.

[32] OpenStreetMap contributors. *Planet dump retrieved from https://planet.osm.org*. https://www.openstreetmap.org. 2020.

[33] Osmconvert. *Convert and process OpenStreetMap files*. URL: https://wiki.openstreetmap.org/wiki/Osmconvert.

[34] SUMO-W. Bamberger. *OsmNetconvert.typ.xml file for edge type configuration for roads*. URL: https://sumo.dlr.de/docs/OsmNetconvert.typ.xml.html.

[35] SUMO. *osmPolyconvert.typ.xml file for additional Polygons (Buildings, Water, etc.)* URL: https://sumo.dlr.de/docs/Networks/Import/OpenStreetMap.html.