A Novel Design of the Transport Infrastructure for Traffic Simulation Models

Tamminga, Guus

DOI
10.4233/uuid:35d2e152-0cfe-439e-a276-da4a69b11acd

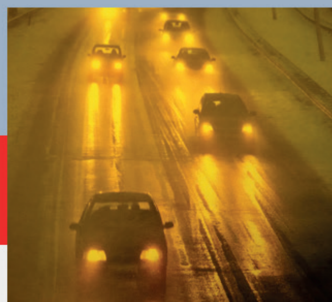
Publication date
2019

Document Version
Final published version

Citation (APA)
Tamminga, G. (2019). *A Novel Design of the Transport Infrastructure for Traffic Simulation Models*. [Dissertation (TU Delft), Delft University of Technology]. TRAIL Research School. https://doi.org/10.4233/uuid:35d2e152-0cfe-439e-a276-da4a69b11acd

TRAIL THESIS SERIES

Guus Tamminga



A Novel Design of the Transport Infrastructure for Traffic Simulation Models

A Novel Design of the Transport Infrastructure for Traffic Simulation Models

Guus Tamminga

Delft University of Technology, 2019

This thesis is part of the research program Traffic and Travel Behaviour in Case of Exceptional Events sponsored by the Dutch Foundation of Scientific Research MaGW-NWO

TU Delft
Delft University of Technology

NWO
Netherlands Organisation for Scientific Research

TRAIL

Cover illustration by Mariette Lavrijsen

A Novel Design of the Transport Infrastructure for Traffic Simulation Models

Proefschrift

ter verkrijging van de graad van doctor

aan de Technische Universiteit Delft,

op gezag van de Rector Magnificus Prof. dr. ir. T.H.J.J. van der Hagen,

voorzitter van het College voor Promoties,

in het openbaar te verdedigen op dinsdag 19 maart 2019 om 15:00 uur

door

Guus TAMMINGA

Doctorandus in Ruimtelijke Economie,

Rijksuniversiteit Groningen

Geboren in Assen, Nederland

This dissertation has been approved by:
promotor: Prof. dr. ir. S.P. Hoogendoorn
2nd promotor: Prof. dr. J.W.C. van Lint

Composition of the doctoral committee:

Rector Magnificus	Chairman
Prof. dr. ir. S.P. Hoogendoorn	Technical University of Delft
Prof. dr. ir. J.W.C. van Lint	Technical University of Delft

Independent members:

Prof. dr. E. Chung	Hong Kong Polytechnic University
Prof. dr. M. Fellendorf	Technical University Graz
Prof. dr. ir. J.E. Stoter	Technical University of Delft
Prof. dr. ir. A. Verbraeck	Technical University of Delft
Prof. dr. F. Viti	University of Luxembourg

TRAIL Thesis Series no. T2019/4, the Netherlands Research School TRAIL

TRAIL
P.O. Box 5017
2600 GA Delft
The Netherlands
E-mail: info@rsTRAIL.nl

ISBN: 978-90-5584-247-6

Printed in the Netherlands

Preface

For several decades I have applied models in the field of traffic and transportation and, at an earlier stage, also in the field of agriculture. In order to get a deeper understanding of the models, I have been very lucky to receive the opportunity to conduct this PhD study.

Serge Hoogendoorn, I am very grateful that you offered me the chance to start this project and for teaching me a lot about traffic under exceptional circumstances. The support from Sweco by providing me time made it possible to actually finish it! Thank you, Bert van Velzen and Frans de Haes. Truly, it was an exceptional adventure.

I have learned a lot from traffic flow models, not in the last place by the ever-inspiring lectures from Hans van Lint, my daily supervisor. Thank you for stimulating and helping me to increase the quality of my thesis.

I would like to thank all of the committee members for your valuable and useful feedback on my dissertation.

During my PhD, I have had the opportunity to work on the development of *OpenTrafficSim*, initiated by Hans van Lint and Alexander Verbraeck. I have really enjoyed this time and want to thank Alexander and Hans for their support, enthusiasm and energy, during my work on this challenging software project. Alexander, Wouter Schakel and Peter Knoppers taught me a lot about software design and coding.

Victor Knoop gave me the opportunity to collaborate in a project for the city of The Hague and actually code the Network Transmission Model, again with support from Alexander. Thanks for allowing me to use your article that describes this project (section 8.3).

From the domain of Geographical Information Systems, Jantien Stoter, Linda van der Brink and Anna Labetski helped me a lot, to gain better understanding of topology, topography and data standards.

During my PhD, I have had the opportunity to collaborate with Mark Miska at the Queensland University of Technology in Brisbane. Thanks for your inspiring thoughts about modelling and the open source approach.

Jan Hendrik van Petegem, Marleen Hovens and Stefan van Gerwen initiated the IMWV that stimulates the implementation of a standardized data design for transportation in the Netherlands: thanks for your initiatives and enthusiasm!

I also would like to thank Conchita from TRAIL for helping me at the final stage of the thesis.

I have enjoyed my time working at the university and had a good time with my room mates Ramon, Dorine, Mahtab, Mignon and Raymond, whom all worked on the VICI project. Thanks!

Finally, I want to thank my family. First of all, my dear parents who supported me throughout their entire life. My dear brothers Meile and Jan Karst for being paranymph during the promotion ceremony. My lovely daughters Noortje, Renske and Carlijn, you give me such joy. And last but not least, my dearest Mariette. You make my life so worthful. Thanks for supporting and cheering me up during the last phase of my PhD. Writing a thesis is no easy job.

Guus Tamminga, February 2019

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Transport is induced by people's activities at different locations. Sleep at home, work at the office, buy food in a shop and have dinner at home again, requires movements between those activity locations. The resulting movements can be performed with different modes (e.g. car, train, bicycle or walking), taking different routes and at various times of day, week and season. Traffic and transport problems, in terms of travel times and throughput, arise when the supply of infrastructure is insufficient to process all movements without delay. The spatial and timely variation in demand determines the quality of accessibility by time of the day and by location.

While accessibility is a stimulant for economic development, transport also leads to adverse external effects such as congestion, noise, liveability, emissions of pollutants and road unsafety. A well-balanced transport policy that considers all of these effects, requires a quantified insight into current and future transport and traffic situations. In many cases, this insight cannot be derived from traffic observation and measurements alone, due to a lack of data. Future situations are not observable at all, so predictions are indispensable. This accounts both for the short and the longer term.

For all of these cases, transport and traffic models can be helpful to gain knowledge about both the current and the future state of transport and thereby support decision making.

1.2 Models: from theory to application

A model aims to provide a simplified representation of reality and focusses on certain elements, considered important from a particular point of view. Transport and traffic models combine mathematical and statistical methods with data to represent the transport system, based on a certain theory about how it works (de Dios Ortuzar and Willumsen (2011)). The mathematical models use equations that determine how a system changes from one state to the next and/or how one variable depends on the value or state of other variables. Various approaches such as discrete, event-based and continuous models have been developed to describe the transport and traffic system (Hoogendoorn (2001), Arentze and Timmermans (2000)). In addition, statistical models include items such as error estimates of observations, data fusion and to statistically characterize input data or model output (de Dios Ortuzar and Willumsen (2011)).

A transport and traffic modelling system contains a structured set of tools and workflows that combines methods and data to estimate and predict various aspects of transport and traffic, including information for related issues such as land use and environmental impact. Figure 1.1 shows a conceptual overview of the modelling process. Transportation is induced by a person′s activities that are performed at different locations, as is shown in the first layer. The second layer, named *transport services*, shows the means for transport to make the complete trip between activity locations. These services contain both public transport and private transport. The market where demand and supply of transport services meet, determines the choices for specific trip-patterns and modalities. In transport and traffic models this phase is often called *demand modelling*.

In the next phase the demand for *transport* services meets the actual supply of *traffic* services, and operational choices about trips, such as the time of departure and the preferred route, are being made. The execution of these trip patterns results in a specific traffic situation. In modelling terminology, this process of route and departure time choice, and making the trip, is called the *assignment* of traffic.

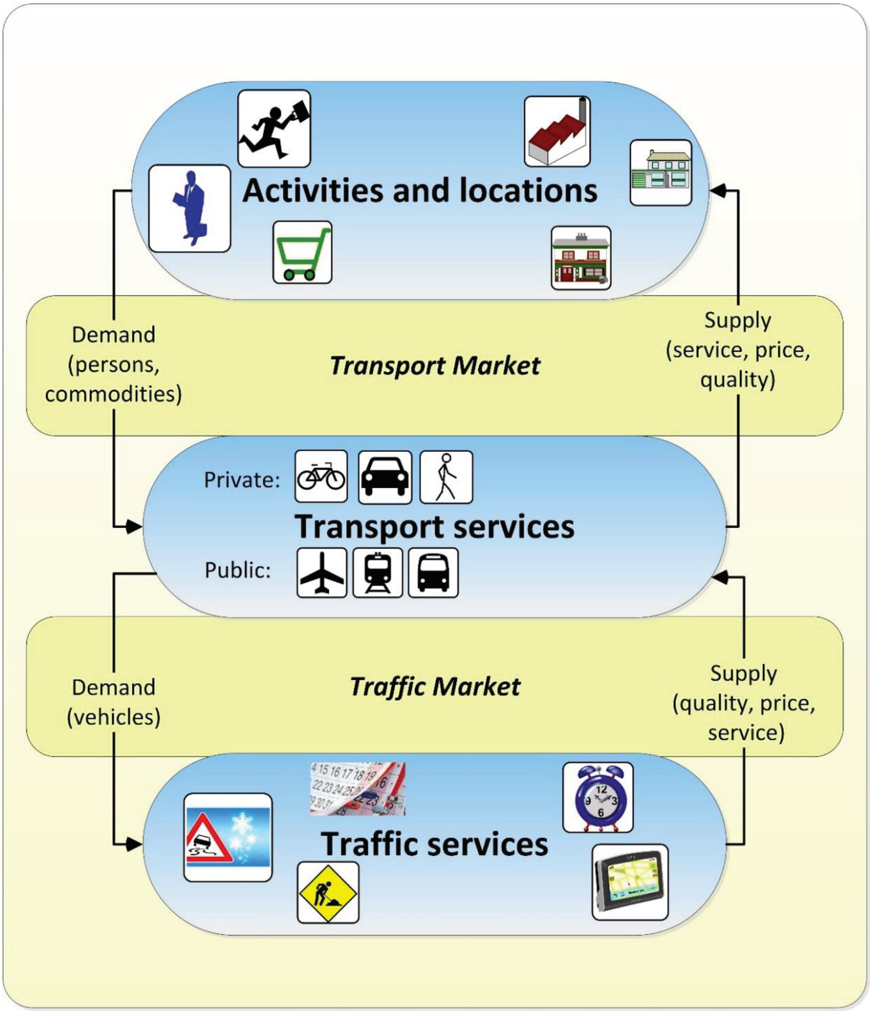


Figure 1.1: layer model of the transportation system (based on Van Nes, 2002)

Due to the computational and methodological complexity, most transport and traffic models are created with specific software applications. These can either be built with dedicated transport and traffic modelling software packages, or generic programming languages such as Java, C++, Python or Matlab. The transport and traffic modelling software packages provide utility tools (such as a graphical editor and facilities to import, store and edit data) and ready to use modelling methods that are primarily based on scientific theories. The implementation of these theories is not unambiguous. Figure 1.2 shows the steps from theory to software, where every step requires choices and implementation issues that are not clear-cut. As a result, the translation from theories and concepts into a software package is by definition not straightforward.

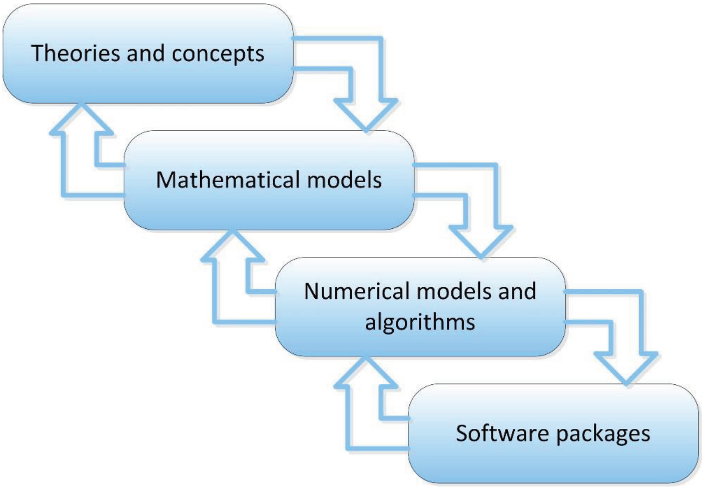


Figure 1.2: modelling from theory to software

One of the main challenges in this process, is to capture human behaviour in mathematical models. An example to show the inherent complexity, is the modelling of the route choice process of travellers.

First of all, there are various theories that aim to capture route choice behaviour. A conceptual choice about behaviour already shows how approaches may differ. The assumption that travellers maximize their utility implies that travellers choose the route(s) with the highest utility. Alternative assumptions base the choice for a route on the minimization of regret or on a satisfying level of utility.

Secondly, these behavioural theories need translation into mathematical formulae. This requires a mathematical formalization and quantification of concepts such as utility or regret with respect to route choice, which again requires a concretization into the factors that define utility or regret and equations that describe the quantification between those factors and the resulting utility (or regret). In many route choice model applications, the maximization of utility is assumed, and expressed as the reverse of the costs of a trip. The quantification requires a selection of the relevant variables and the mathematical format of the expression (linear, quadratic or otherwise). Traditionally, travel time, distance and monetary travel costs are implemented as dependant variables. More sophisticated approaches may add additional factors such as comfort of a trip and the reliability of travel times. More degrees of freedom in the design of the model arise from the fact that route choice behaviour may vary between travellers. The cost function may either be defined for an *average traveller* or specified for various homogeneous groups of travellers. To estimate these functions, we can use several research approaches, such as the method of stated preferences or that of revealed preferences.

In the *third* step the numerical models and algorithms are defined. Assuming that route choice is based on the fastest route, the quantification of travel time is needed to determine the optimal route (or path). The computation of travel time now depends on the type of assignment. In a macroscopic model, travel time can be derived from the functional diagram, whereas in simulation models the average speed of individual vehicles is a result of interaction with other vehicles and the (infrastructural) environment. Additionally, the travel times also depend on the assignment technique, such as the one shot-assignment or equilibrium methods such as the method of successive averages (MSA). In the *fourth* and final development step the numerical models are implemented in the actual simulation software. This requires additional choices such as object and data definition, simulation scheduling, error handling, etc... While being far from exhaustive, this example already shows the tremendous amount of choices that need to be made for building and applying a route choice model.

Data input is essential for transport and traffic models to describe the population, the activity locations, the traffic modes (cars, trains etc.) and the physical environment with its infrastructure and traffic systems. Moreover, data is required to define, derive, estimate and validate methods and algorithms that are part of the model. The amount and quality of available data determines both the modelling approach that is chosen and the quality of the model that can be achieved. For example, the assignment of traffic to a network requires information on all trips in terms of origin, destination and time of departure. If this information is available from actual data sources (could be from navigation systems where people enter a trip before they leave), we would not require methods for estimating trips and will have a higher quality of the description of traffic demand.

This wide range of choices in terms of methods and algorithms has been one of the sources for the development of transport and traffic model systems that significantly vary in their architecture, design and contents. The next section provides a short overview of the development of transport and traffic models from a historical perspective.

1.3 The use of transport and traffic models

In addition to the methodological choices, the use cases of transport and traffic models also vary widely, ranging from decision support for planning at a global and strategical level, towards detailed real time models that are embedded in operational systems (see Table 1.1).

Table 1.1: Fields of application for transport and traffic models

Name	Description
Decision support for planning	
- strategical planning	Longer term policy measures with an impact on land-use and the design of transport system
- tactical planning	Optimizing the traffic system given the existing facilities and infrastructure for instance by managed motorways
- operational planning	Estimating the impact of short-term measures such as an optimization of the signal plans of junctions
Ex-ante evaluation	A controlled environment for testing new technologies, systems and approaches
Real time traffic information and operations	On-line models within traffic control systems or in-car systems for traffic state estimation and prediction of future states

Based on Henk, Ballard et al. (2007)

For these different fields of application, various types of modelling approaches have been developed: real time models require a totally different approach than strategical planning models. But also, within a certain field of application, various approaches and methods are applied, even when these model systems have the same objectives.

A large part of the commercial transport and traffic model packages provides ready-to-use tools to develop model applications. The emphasis of these tools is the usability of the software, enabling users to create efficient work flows and providing utilities to import, create and edit traffic networks and controllers, generate and estimate traffic demand, and simulate and visualize vehicle movements, and analyse outputs. While these model packages offer similar functionalities, they use totally different designs in terms of data structures, modelling utilities and work flows. As these various model packages mostly apply their own data definitions and design, the exchange of data between models that use the same level of detail requires significant efforts. For models from different fields of applications, an attempt to achieve consistency and exchangeability is even more challenging. This is caused by differences in scale and level of detail, in combination with the absence of standards for data exchange (Tamminga, van den Brink et al. (2013)). We define our first issue:

(1) *The mutual exchange of data, both between models and with external data sources is hampered by a lack of data-standardization*

In terms of functionality and usability, most of the commercial packages provide sufficient capabilities for a large share of the user groups. Yet, for users that are focused on the development and application of new and innovative methods, the emphasis on ready to use building blocks has its drawbacks. The most important one is that the software is closed, meaning that users are not allowed to read and edit the source code. In most cases, the packages provide insight in parts of the formulae, and allow the user to change the parameters. In some cases, also part of the functions can be replaced by the user. Still, the opportunity for adjustments remains restricted to bounds that are imposed by the developing company. This inability to edit the source code freely, hampers users who wish to investigate the underlying methods, or are looking for possibilities to adjust or develop new algorithms and methods. This retards the development of new and innovative modelling approaches (Tamminga, Miska et al. (2012).

New trends and developments, for instance in the field of intelligent transport systems, and information and communication technology may significantly change the transport and travel behaviour and processes. The introduction of automated driving may lead to a strong reduction of headways between such cars. In the long term this may lead to higher road capacity and increased road safety. However, such a transition will take a long time. During this transition period there will be a mix of vehicles with different levels of automation. The challenge for transport and traffic models is to estimate the impact of these changes on capacity and safety.

Yet, current traffic flow models are not capable to reflect the behaviours of such a mix of vehicle types. As stated by van Lint (van Lint, Schakel et al. (2016)) "there is no unified theory of driving yet that enables us to quantitatively predict on beforehand the effects of increasing percentages of (heterogeneous) vehicle automation capabilities on either capacity or safety during this transitional period. The incorporation of psychological concepts in simulation can help in explaining such variations in driver behaviour. We require flexible transport and traffic models that make it possible to incrementally extend microscopic models with explanatory mental models, such that new behavioural theories can be tested and shared within the research community." The commercial transport and traffic model packages are not fully suited for this challenge. As the proprietary software of these packages is 'closed' there is generally no opportunity to use and extend the source code. The de facto option then is to code transport and traffic models (or parts thereof) from scratch. There are many merits in doing so, e.g. the learning experience of designing and coding itself, and in the fact that the software can be tailored specifically to a researcher's needs. There is however, a large cost involved. First of all, starting from scratch requires designing and coding much common and auxiliary functionality just for the basic simulation to run. Examples are network coding and storage, data preparation and streaming (IO), simulation bookkeeping, visualization, etc. Besides the obvious costs in terms of time and effort, much of this repetitive work is relatively mundane and does not contribute to scientific advances. Secondly, since code reuse is rarely a requirement in research projects, it is often cumbersome and time-consuming for others (including fellow academics) to utilize and extend transport and traffic models developed by their academic peers. As a result, a large portion of academic traffic simulation software exists for as long as its creator supports it, after which only a limited amount of supporting documentation (including scientific articles) is available for successive users. To prevent this loss of knowledge and experience, an approach is required where code reuse and proper documentation is stimulated. This leads us to the second issue:

(2) *We require a transport and traffic model system that enables model users and developers to reuse knowledge and/or extend it with new and innovative functionality*

These two issues are the main motivation for the research that is reported in this thesis and investigates:

- The requirements to create an open and free to code transport and traffic model environment
- The design of the data objects and structures of the infrastructure within a model environment
- The conditions and developments that are required to create an efficient data-exchange between models, and with external data-sources

1.4 Transport and traffic models and innovative research: the open source approach

The open source software approach is one of the means to create an open and free to code transport and traffic model environment. This approach at least guarantees that the generated knowledge, in the form of source code and documentation, stands open to anyone. Yet, not every Open Source Project becomes a success (Michlmayr (2005)). It at least requires sufficient preparations with respect to project governance, the transport and traffic modelling architecture and its implementation in software (design and quality) (Lerner and Tirole (2002), Prlić and Procter (2012), Raza, Luiz F. Capretz et al. (2010)). Once these requirements are fulfilled, the availability of reusable and mature software allows for continuous development of new and improved models (activity based, car following, merging, gap acceptance, response to ITS, etc.).

The design of such a modelling environment is a complex process in which many disciplines are involved. Figure 1.3 provides an overview of the main skills and disciplines that are required. The top of the figure shows that for the development of the model architecture, defined as the global functional design (or skeleton) of the transport and traffic model, knowledge is required from transportation, traffic flow behaviour and modelling on the one hand (domain knowledge) and from computer science and informatics on the other hand. In this initial phase of the model development process, the main building blocks, the key objects and the global relationships within the model are identified by the transportation and traffic modeler. The software designer regards the requirements from the informatics viewpoint, such as software quality, re-usability of components, efficiency, documentation and longer-term maintenance.

After this first phase a further elaboration of the model design with gradually more attention for operational issues, requires the input of more specialized knowledge. As an example, the modelling of route choice requires knowledge of path searching algorithms (operations research), and route choice behaviour from the domain of traffic and travel behaviour. Meanwhile, continuous attention from the domain of informatics is required to attain sufficient quality in terms of software design and programming skills.

As the domain of transport and traffic modelling covers a wide field of different topics it is sheer impossible to create a blueprint of the model that remains completely intact in course of time. An agile approach that enables adjustments of the initial architecture during the elaboration of components of the model system is required, allowing for changes of parts of the system. A modular approach with loosely coupled building blocks supports such a process.

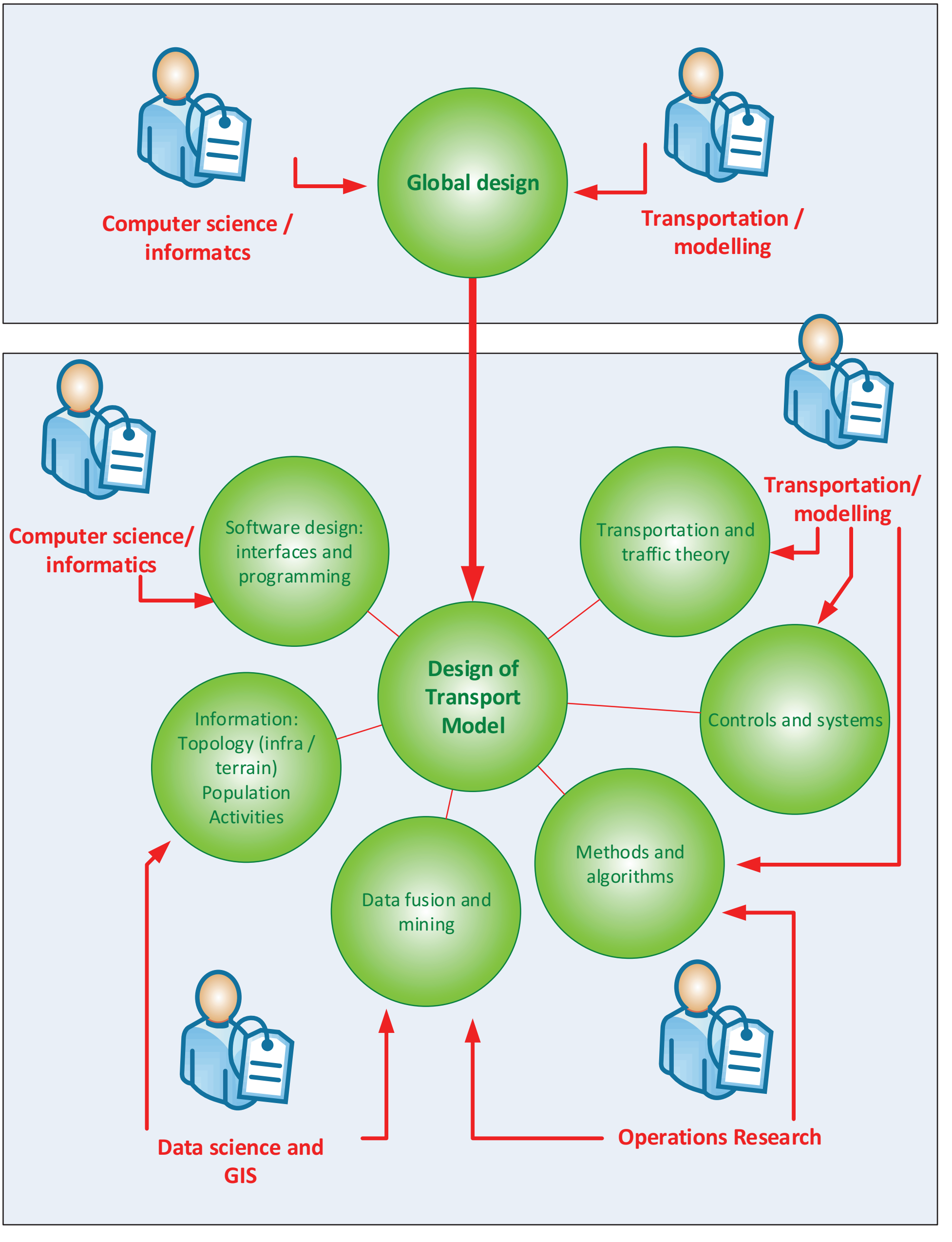


Figure 1.3: Indication of required skills in the design process of a transport and traffic model

The open source approach adds additional requirements to the architecture and model building process. Without the ability to parcel out work to programming teams in different areas, the effort is likely to be unmanageable (Lerner and Tirole (2002). This requires a design where non-coding developments are eliminated as much as possible, for instance by creating a clear, logical and intuitive modular design and by self-documenting programming approaches. One of the key issues of the design of an open source transport and traffic model is the ability to isolate specific model behaviours in separate modules, while simultaneously allowing the integration of modules into coherent and logical work flows. In such an ideal setting, researchers can focus on the modules of their specific interest. Only the pieces of code that are relevant for research can then be studied, evaluated, adjusted and hopefully result in innovations. One of the advantages of open source software is that after finalizing a new module, the costs of sharing and distributing it are nearly zero, which supports the spreading of knowledge.

The next section introduces the *OpenTrafficSim* project, an example of an open source software project for traffic microsimulation with free access to the code and thus to all algorithms.

1.5 Introduction of the *OpenTrafficSim* project

The open source simulation suite *OpenTrafficSim* has been initiated by Van Lint and Verbraeck from Delft University of Technology (van Lint, Schakel et al. (2016, Verbraeck (2017)). The main motivation for setting up *OpenTrafficSim* is the need for an open source software platform that provides free to use code for traffic simulation, which has also been stated by Miska, Santos et al. (2011) and Tamminga, Miska et al. (2012). The *OpenTrafficSim* traffic simulation framework builds on the design of microsimulation model MOTUS (developed by Schakel: Schakel, van Arem et al. (2013)) with which already a number of ex ante evaluations of advanced driver assistance systems have been performed.

OpenTrafficSim has been built on top of the open source simulation package DSOL (Distributed Simulation Object Library: *Jacobs (2005), Jacobs, Lang et al. (2002))*. As is elaborated in van Lint, Schakel et al. (2016) "DSOL is a Java-based, object oriented, multi-paradigm simulation environment that prepares for distributed and parallel execution of the simulation model." The DSOL environment enables multi-formalism and offers many advantages for *OpenTrafficSim* (van Lint, Schakel et al. (2016)):

- DTSS – Discrete Time Systems Specification
  Time-stepped models can be used when the system state is evaluated at constant intervals;
- DEVS - Discrete EVent Systems Specification
  The event mechanism is easy to use for scheduling traffic lights, arrivals of vehicles in the system, and models where the state is recalculated at scheduled time instants rather than at constant intervals;
- DESS – Differential Equation Systems Specification
  DESS (sub)models can be used for calculating non-linear acceleration and braking behaviour, which is usually relatively easy to represent as a set of differential equations;

The opportunity to use and combine these update mechanisms in *OpenTrafficSim* allows a scheduling of (driving) behaviour that is explicitly modelled. Each driver can plan a path based on observations and schedule state changes using the event scheduling mechanism of DSOL. The simulated driver computes a continuous path over the infrastructure for the next n time units. The schedule interval *n* can be as short (e.g. one time step) or long (20 seconds) as needed. To compute such a path, the driver needs to make assumptions (predictions!) about drivers around him. The re-evaluation of his path will occur either at the intended re-evaluation interval or as soon as circumstances dictate. The modularity of the *OpenTrafficSim*/DSOL framework makes it possible to incrementally extend microscopic models. An example is the implementation of explanatory mental models, such that new behavioural theories can be tested and shared within the community of transport and traffic research (van Lint, Schakel et al. (2016)).

The provision of ready to use building blocks from *OpenTrafficSim* intends to stimulate and support students, researchers and model-appliers to learn and innovate by:

- providing state of the art knowledge that is already available within the existing model system;
- providing ready to use utilities, that allows users to concentrate on modelling methods and algorithms and thereby use their resources efficiently;
- adding knowledge to the system with new contributions;
- sharing this knowledge with the transport and traffic research community.

1.6 Objectives and research questions

This research relates to the development of *OpenTrafficSim* and focusses on the following issues.

The first issue regards the requirements for developing an open source software project. While there have been numerous publications that are focused on the development of dedicated modelling methods and algorithms, far less attention has been paid to the architecture, design and practical implementation of a comprehensive transport and traffic modelling system. The first research objective is:

- *Derive the key requirements for the development of a successful open source software project for transport and traffic.*

The second issue regards the way of modelling the transportation infrastructure in an open source transport and traffic model. In this thesis the scope is on the modelling of road based traffic. The related research objective is:

- *Create a design of the transportation infrastructure for road based traffic that firstly meets the functional requirements for transport and traffic models and secondly enables an efficient data-exchange between models and external data-sources.*

To address these issues two main research questions and a number of sub-questions are investigated throughout this research:

Question 1:
What is the best approach to create a transport and traffic model that enables model users and developers to reuse the existing code, learn from it, improve existing methods, and/or extend it with new and innovative functionality?

Q1.1: What are the requirements and demands for an open source software modelling environment for transport and traffic?

Q1.2: How to organize this open source software project in terms of software development and governance structure to optimally support the (re-)usability of the system?

Both sub-questions are elaborated in chapter 3 by reviewing literature on open source software projects. It results in recommendations and requirements for the development of successful open source projects. On the basis of these requirements, chapter 8 evaluates the OpenTrafficSim project, by reviewing the project and two case studies.

Question 2:
What is a good design of the transportation infrastructure for road based traffic, in order to fulfil the functional requirements and enable a proper data-exchange with other models and external data sources?

Q2.1: What are the global requirements for the design of the transportation infrastructure when regarding the functional demands from the varying transport and traffic modelling approaches?
This question is handled in chapter 4.

Q2.2: What are specific requirements for a design for the traffic infrastructure that enables a transition between global and detailed representations of a road traffic infrastructure and allows the modelling of all modes of traffic on this infrastructure.
This question is handled in chapter 5.

Q2.3: What are the requirements to achieve a proper exchange of data between the transport and traffic model and external data sources, with a focus on the road based transportation infrastructure?
This question is handled in chapter 6.

The answer to these questions provides the necessary input for the design of the transportation infrastructure for road based traffic, as will be elaborated in chapter 7.

1.7 Main Contributions

Most of the scientific contributions of this thesis are of a theoretical nature and intend to support the design process of a transport and traffic model system based on an open source software approach.

Scientific contributions

The main scientific contributions are summarized in this section. The information from previous sections is combined with more details from the rest of the thesis in order to make the contribution concrete.

This thesis describes the scientifically contributions in the following ways:

- Identifying the requirements for the development of an open source software project for transport and traffic modelling, considering the project governance and the software requirements (chapter 3);

- Determine the main objects of transport and traffic modelling, their relationships, and their representation at various levels of detail and scales, by analysing the modelling process from activity generation onto traffic operations (chapter 4). This architecture provides a basis for the transport and traffic model research environment *OpenTrafficSim*;
- Identify the requirements for the design of the transport infrastructure that enables transport and traffic modelling at various levels of detail (chapter 5)
- An evaluation of the relevant geo-spatial data standards for the transportation infrastructure, and a proposal for improvements in order to match these data-standards with current and future data requirements for transport and traffic models (chapter 6 and 7);
- Some additions to the *OpenTrafficSim* design of the transportation infrastructure. Specifically, the relation with the activity locations (chapter 7);

Practical relevance

The theoretical contributions and insights as discussed in the previous sections do have a practical relevance. As will be discussed in section 2.5.1, a generic transport modelling tool that is specifically designed for research purposes and oriented towards usability and learnability, is lacking. The main practical contribution of this PhD project is the start-up of an open source software project that aligns to the concepts and ideas from this thesis. The OpenTrafficSim modelling environment contains tools, methods and utilities to enable a modular modelling development. As the developments are still ongoing, we refer to the website http://www.opentrafficsim.org/ for further and actual information. A second practical contribution concerns the review of current geo-spatial data standards for the *transport infrastructure*, and guidelines for the improvement of these standards. This knowledge is used to support the development of the Dutch data model for roads and traffic (IMWV)[1] and applied in a pilot for the province of Noord-Brabant with an implementation of the GIS representation of the roads and junctions at three levels of detail[2].

1.8 Scope of this thesis

Transportation and traffic models cover a wide area of topics and modalities with variations in scale (level of detail), size (area) and planning horizon that depend on the research objective. In this thesis, we largely restrict ourselves to road oriented transport infrastructure, including tram railways at roads, and transfer points with rail based traffic and pedestrians. Yet, many of the concepts are applicable to a wider range of transportation types and modes. This specifically accounts for the contents of chapter 4. The remaining chapters describe the infrastructural design and use of *OpenTrafficSim* and focus on road based traffic.

The microsimulation approach models individual vehicles and their interactions on a road. The interaction of this behaviour within a certain configuration of the infrastructure determines traffic flow behaviour. And thus, provides the capacity of a road as an outcome of a simulation.

[1] https://www.crow.nl/thema-s/wegontwerp/imwv-informatiemodel-wegen-en-verkeer

[2] Project carried out by Sweco and commissioned by Stefan van Gerwen from the province of Noord-Brabant

With coarser approaches - the mesoscopic and macroscopic traffic flow models - this vehicle behaviour is captured into continuous formulae that use flow as an input. These equations require attributes and parameters, such as capacity, that are linked to the infrastructure. This means that part of the behaviour now is captured in an attribute that is part of the link or lane. This is depicted in Figure 1.4: the description of the infrastructure within the microsimulation models does not contain "behaviour", but at coarser levels it is incorporated in the modelled infrastructure. The elaboration of methods and algorithms that relate to these issues is not part of this thesis: we restrict ourselves to the representation of the data objects.

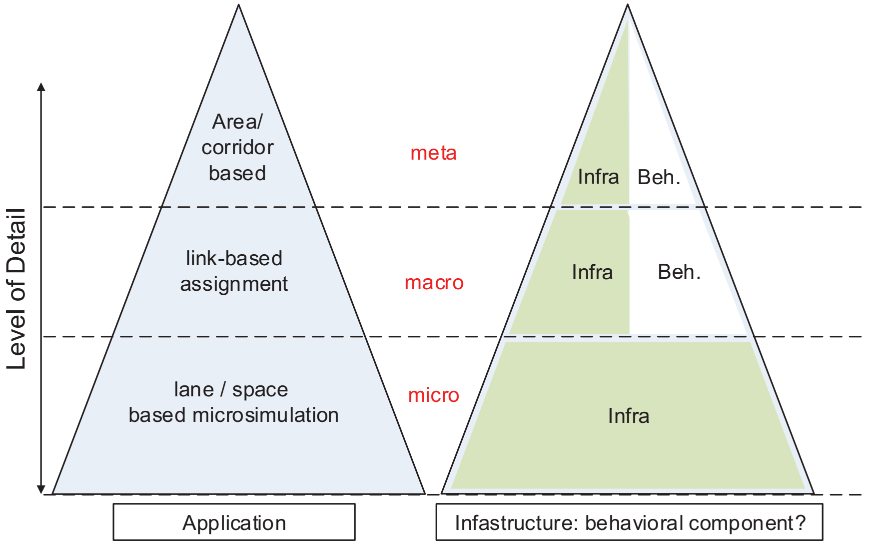


Figure 1.4: Modelling approaches: behavioural components captured by infrastructural attributes

1.9 Outline

In order to answer the research questions that are raised in section 1.6, this thesis is structured in 9 chapters.

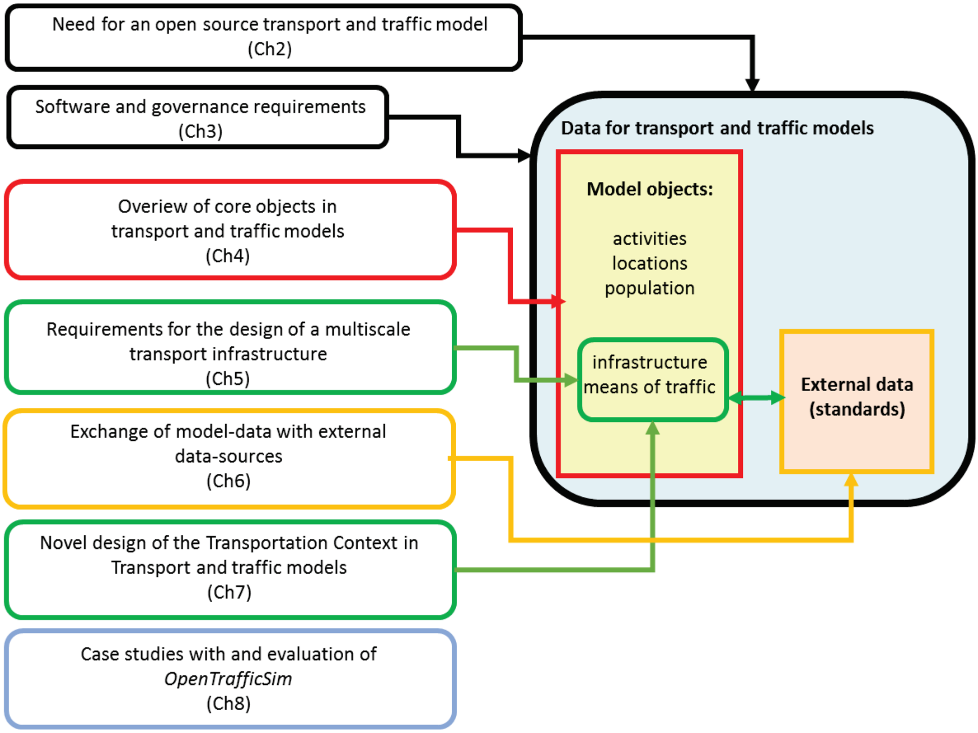


Figure 1.5: Thesis outline

Chapter 2 starts with an overview of the transportation and traffic system, and how this can be structured in models. This is followed by a historical overview of transport and traffic models, and an outlook for future model challenges. Subsequently, the issues raised in chapter 1, are further elaborated. We identify the demands from the intended user group (science and research) for an open source software modelling environment. In addition, we explore the requirements to create a design of the transportation infrastructure for road based traffic that firstly meets the functional requirements for transport and traffic models and secondly enables an efficient data-exchange between models and external data-sources. This is further elaborated in the succeeding chapters.

Chapter 3 investigates the requirements for an academic open source software project in terms of software development and governance structure, given the objective to optimally support the (re-)usability of the system.

Chapter 4 sketches the rough architecture of the relevant objects of the infrastructure within the transport and traffic model system, by analysing the modelling process from activity generation towards the execution of trips. Aspects that are further elaborated in this overview are the variations with respect to the modelling approach (methodological aspects), the level of detail and the scale of modelling, and finally the various modes of transport. Transport and traffic models require input data for the representation of the real word. To facilitate the exchange of data, data objects in transport and traffic models should align to their counterpart in GIS databases.

Chapter 5 explores the design of the model objects of the *transport infrastructure* for road based traffic. This chapter elaborates upon the requirements that are already described in chapter 4.

Chapter 6 focusses on the exchange of model objects with external standardized data sources and compares and aligns the model objects with their counterpart from the open data standards for geo-spatial information. The choice of a suitable geospatial data standard is included in this chapter.

Chapter 7 presents the objects that represent the physical infrastructure in *OpenTrafficSim and* suggests some additions. The main challenge is to design a road infrastructure ontology that enables a smooth transition between global and detailed representations of objects, captures all modes that use the road infrastructure, and allows an exchange of data with internationally accepted data standards.

Chapter 8 provides three case studies with *OpenTrafficSim*. The first case shows the import of a microsimulation network from OpenStreetMap, which illustrates both the opportunities for an automated exchange from data sources to model-input, but also highlights the lack of detail in these types of digital maps. The second case presents a hybrid modelling approach that shows the construction and application a Network Transmission Model for the city of The Hague. The last case presents the N201case and describes a real time simulation model to show (1) the functionality of the simulation and (2) the opportunities to modify and extend the *OpenTrafficSim* functionality. The chapter finishes with an evaluation of the open source software approach of *OpenTrafficSim*.

Chapter 9 presents the conclusions of this research by answering the research questions raised here, highlighting practical and scientific implications and suggesting possible future research directions.

Chapter 2

The need for an Open Source Transport and Traffic Modelling framework

This chapter describes the main motives to initiate the design of an open source software modelling framework for transport and traffic, and starts with an overview of the transportation system and the main phenomena that are captured by transport and traffic models (section 2.1). The design of these models depends on various factors, such as the purpose to apply the model, the scale of the area, the transport elements and modes they describe, and the availability of input data. These factors determine the modelling approach and the level of detail with respect to the description of the phenomena (2.2).

The development of the computerized transport and traffic models started halfway through the twentieth century. Section 2.3 provides a historical overview of the developments of these models. In most cases, commercial firms provide the software to create such models. As this software is basically closed, meaning that the programmed code in not free and open for users, users that want to investigate, modify or extend the model, are unable to do so. This specifically accounts for users from the field of academia, research and ITS (2.4). The open source software approach offers an alternative for these user groups, as it enables users to examine and use, adjust, or extend the source code freely. This is the main motivation for initiating the development of OpenTrafficSim, an open source software package for the microscopic simulation of traffic. As this is not the first attempt to start and create an open source software project, an evaluation of likewise projects, both within and outside the domain of transportation, is presented in section 2.5. Finally, the data formats of transport and traffic models often do not align with internationally accepted data standards, which hampers the reuse of existing models (data, networks) as is elaborated in 2.6.

2.1 Overview of the transportation system

The layer model of the transportation system as described in section 1.2 shows a conceptual overview of the transport and traffic modelling process. Figure 2.1 shows the main steps of this process for a single person and starts with the choice of activities and their locations. A broad range of factors characterizes these choices. First of all, the decisions are related to the person's characteristics and his household, as there are strong mutual dependencies between their activity- and transport choices. In combination with the spatial environment, in terms of the transportation infrastructure and activity locations, this largely determines the opportunities to perform activities.

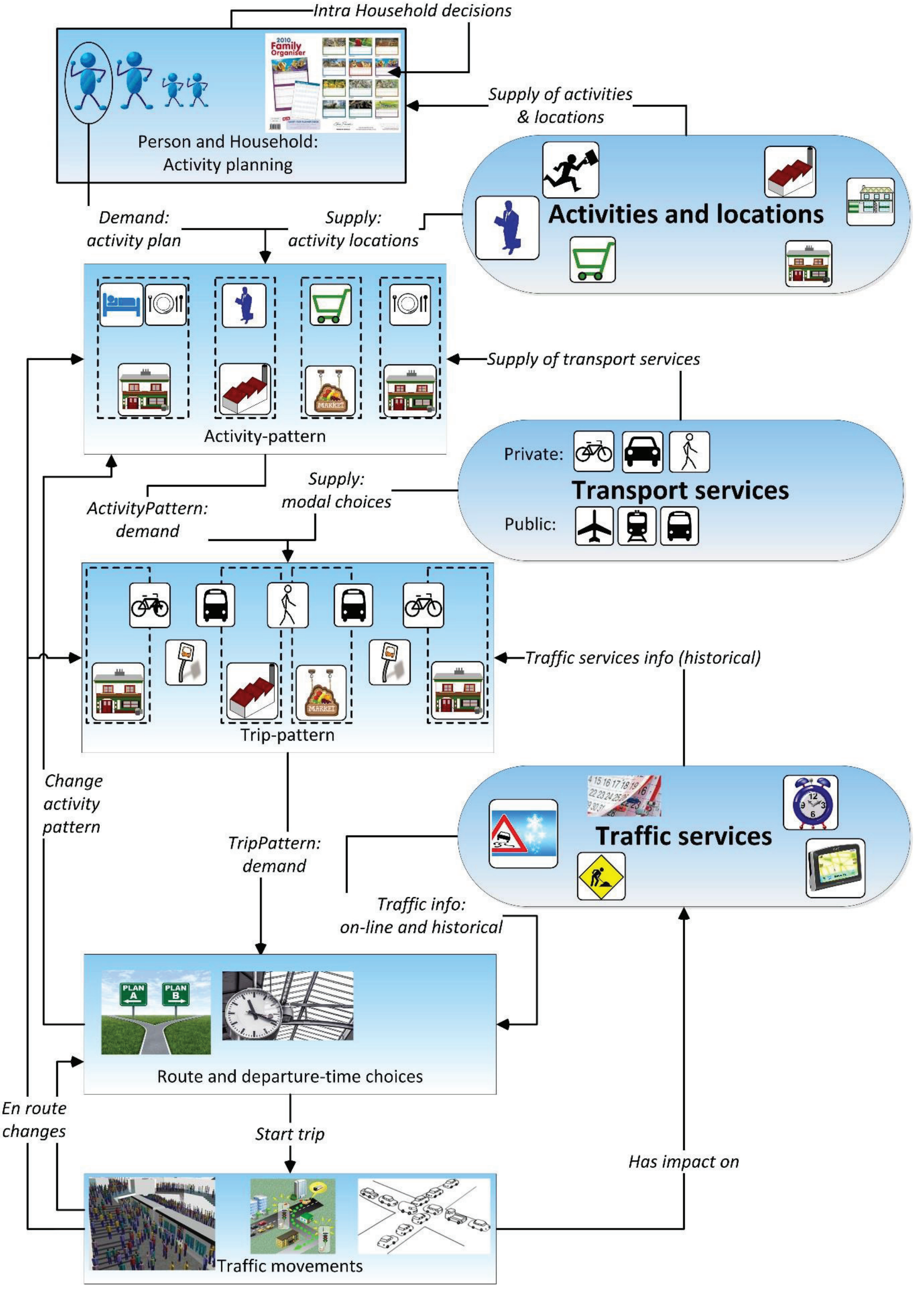


Figure 2.1: Modelling transport & traffic at a person's level

Some of the activities, such as where to live and work, will already be captured for a longer period, while others, such as recreational activities, have a higher degree of flexibility. The activity pattern is a result of the way these activities are planned and scheduled over a specific period and contain the planned transport service to move between the locations. A transport service is defined as a series of trip chains that may differ by mode.

The actual execution of the trips involves an evaluation of the traffic situation and is largely impacted by daily varying factors such as the weather, the actual state of the infrastructure, accidents and other stochastic factors, and usually results in the choice of the departure time and route. At this point, but also during the trip, unexpected changes in the traffic situation, for instance due to exceptional circumstances that deteriorate the service and quality level of traffic, may lead to a reconsideration of the route, or even trip- and activity patterns. In these cases, the process of demand modelling and assignment is not strictly sequential, and follows an iterative approach.

The aggregation of all of these choices on the level of individual households and other (organizational) entities, ultimately results in a collection of activity plans and trip patterns that reflects the total demand for traffic. Relinquishing the complex process of combining individual choices to retrieve a valid collection of all plans[3], the execution of all individual trip-patterns results in movements by modality, leading to traffic volumes that vary by time and place. In peak periods, traffic demand is relatively high and may exceed the supply of infrastructure causing delay and congestion, while demand can be relatively low in other periods.

One of the main aims of transport and traffic models is to provide insight in the (im)balance between supply of infrastructure versus demand of transport, as a starting point for the search towards solutions that generate a better performance. These solutions can be translated into measures in the model, and by applying the model, their impact can be analysed.

2.2 Transport and traffic modelling approaches

There is a wide variation in transportation and traffic modelling approaches, which is partly due to the following factors:

- their purpose or use (van Noort, van Arem et al. (2010));
- the transport element(s) and modes they describe (for instance transport over road, water, rail or side-walk);
- the scale with respect to time (from seconds to decades) and space (segment, lane, link, route, corridor, network) as in Barceló (2010), and by their level of detail (Hoogendoorn (2001);
- the underlying modelling paradigms (for example Arentze and Timmermans (2000) and Ben-Akiva, Bottom et al. (2007)).

[3] In reality, there will be a bidirectional and cyclical process, where the aggregation of individual decisions determines the amount of traffic, and, in case of high peak traffic, possibly leads to a reconsideration of the initial individual travel choices.

The purpose for using a model directs further choices with respect the scale of the model in terms of *study area*, the *level of detail* that is required, the representation of *behaviour*, the *time horizon* and the relevant *transportation modes*.

Two examples reflect the impact of the purpose of the study on the choice for a specific modelling approach. In case of short term operational issues, such as the optimization of an oversaturated junction, the traffic model would only require a model that simulates the junction, with its traffic demand for all turning movements as its input. A second example regards a study to explore the long term city developments and its consequences for transport and traffic. This case requires a large scale transport and traffic model in terms of time and space, and has far more aspects to incorporate. Activity patterns of people, as well as the demographic and socio-economic situation, may change substantially. As a result, trip patterns may change significantly. This requires a model approach that provides insight in the future demand of transport and traffic. The analysis concentrates on insight in changing future demand patterns, but a less detailed modelling of traffic flows (compared to the first case).

Figure 2.2 provides a schematic overview of the main factors that have to be considered, when choosing a modelling approach for a specific case. The use case determines the objective and purpose of the study. On its turn, this further determines the level of detail, the scale, the modes and the phenomena that need to be modelled. After this phase, there still are choices to be made. First of all, there is a methodological choice. Transport and traffic models combine multiple methods and algorithms, there is often no unique answer to the question which model design is best. Perhaps, multiple approaches (approach A or B, as shown in the figure) are satisficing from a methodological point of view. The choice than may depend on other factors such as the availability of data or the financial means to develop the model.

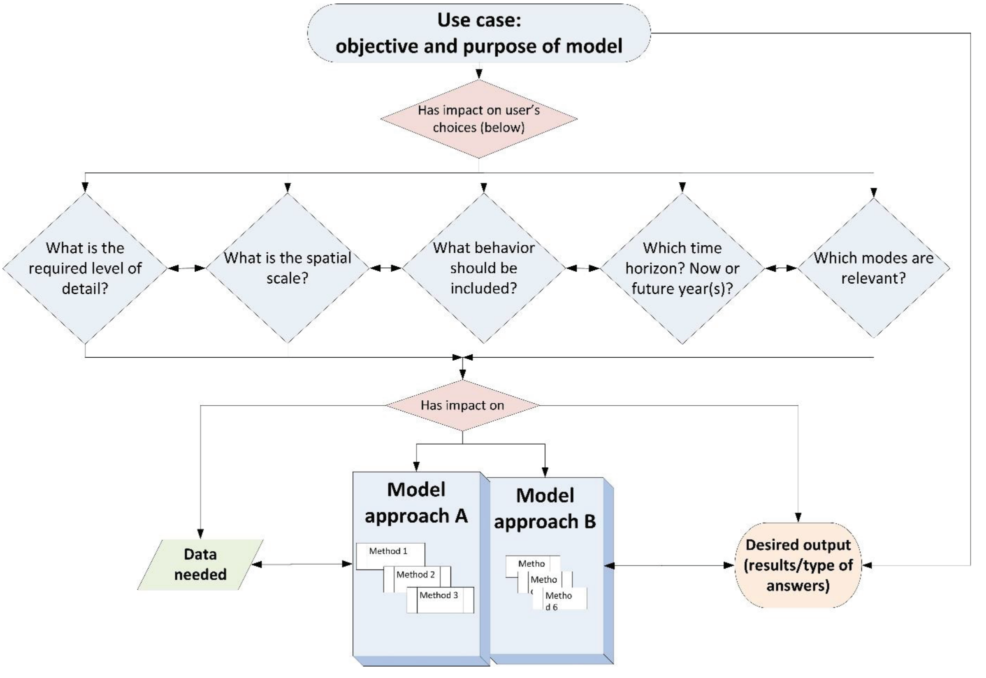


Figure 2.2: the use case, its impact on the phenomena to be modelled, and subsequently the choice for a model approach

2.3 A historical overview of the development of transport and traffic models

Since the second half of the 20th century, transport models have been developed as tools for the support of planning decisions on new infrastructure. The development of computers provided the computational power that is required for applying transport models. Roughly a decade after the introduction of the first computer in 1942, the first transport models that ran on a computer were developed.

In 1955 the Chicago Area Transportation Study (CATS), based on an procedure pioneered in Detroit, was one of the first projects that utilized a transportation model using trip generation, trip distribution, modal split, and traffic assignment models for travel forecasting (Weiner (1986)). "The task of the Study is to analyse the present travel behaviour, to forecast what the future requirements of the metropolitan region will be and, on the basis of this information, to devise a long range plan for needed highways and for mass transportation facilities" (Planning (1959)). These types of models principally were developed to explore the longer term impacts of urban and regional developments on traffic demand and the impact of new infrastructures.

These strategical plans also posed specific challenges for more operational choices, such as the design of the freeways and interchanges. As was already noted in the early sixties of the previous century: "traffic engineers have become concerned over the lack of knowledge about the nature of traffic flow. This lack of knowledge has hampered the design engineer in his ability to design an expressway, which will move traffic smoothly and efficiently at a minimum cost. All too frequently after a highway has been opened it has been found that traffic problems arise, which require extensive redesign and construction". Then already, questions arose that are still relevant today, as is reported by Glickstein and Levy (1961): "

- What should be the length of an acceleration lane;
- What is the effect on traffic flow of locating interchange areas at various distances from one another;
- What is the effect of various speed minimums and maximums on the traffic flow?"

This motivated the development of computerized models to simulate the behaviour of traffic flows in urban areas and cities, and to evaluate alternate design criteria. In 1955 Gerlough developed one of the first computerized simulation models for freeways (Gerlough and Traffic (1956)). One year later, Goode developed a model (Goode (1956)) that simulated the movement of traffic through a signalized junction (Glickstein and Levy (1961)). A few years later the Traffic Network Simulator (TRANS) evaluated the traffic signal settings on traffic flows in a city (Katz (1963)).

Although these initial simulation attempts faced the limitations of computers in terms of computational speed and memory, it was already noted (Goodell (1997)) that: "It is possible to imagine a time in the future when every large city will have its modelling team and computer which the traffic engineer will use as a tool to try out his projected changes, assuring himself of a smooth transition to a new optimum use of his city's traffic facilities. The small communities, unable to afford a large machine, would band together perhaps under state auspices, to use a single large machine. Since the model, once developed, can be changed with relatively little effort, even small communities will be able to develop the technique without having a machine standing by." While computer developments outdated the assumption of the high costs of large computers, the urbanization with a growth and concentration of activities in a relatively small area, indeed stimulated the use of transport models as a tool to support traffic operations and transport planning. In fact, this also created a market for dedicated traffic modelling software tools. Initially, these tools were a spin-off from in-house developed model applications from companies or research institutions. Early examples of models for transport planning are the Strategic and Regional Transport Model (START) model of MVA (Bates J (1990)) and the Emme modelling package as a spin-off from research by Florian in the late seventies. Also other packages like MVA (UK)'s TRIPS, ITS Leeds' SATURN, MINUTP and TranPlan entered the market (Wigan and Drain (2002)). Many of these models were used to support the medium and long term planning decisions on urban and regional developments concerning issues on land use and infrastructural facilities.

Due to the ongoing economic and demographic growth, with an increase in car availability, an increasing number of urban regions encountered situations where traffic became such dense that the available infrastructure could not sufficiently handle the flows at all times of the day. Recurring peak hour congestion with an increase of delays and less reliable travel times emerged. These acute problems required short term actions and policies. One of the remedial measures for addressing the congestion problem in the short term, is the deployment of dynamic traffic control and management systems, supported by Intelligent Transportation Systems (ITS). ITS is the application of current and evolving technologies to transportation systems and the careful integration of system functions to provide more efficient and effective solutions to multi-modal transportation problems (Boxill and Yu (2000)). One of the first instruments that was deployed to optimize traffic flows dynamically, is the control of junctions by traffic lights.

The challenge to diminish the congestion problems supported the development of new and advanced modelling methods within the last step of the four stage model: the assignment of traffic to the infrastructure de Dios Ortuzar and Willumsen (2011). As a result, the quality and complexity of traffic modelling methods to simulate congestion gradually evolved. As for instance Geroliminis and Daganzo (2007) describe, since the 1990s a new series of dynamic assignment methods evolved, ranging between macroscopic and microscopic approaches. Again, the on-going increase of computing power and the possibilities of automated data gathering and processing, together with more advanced software and information technology, enabled and supported the development and implementation of more sophisticated modelling tools. An inventory of micro simulation models at the end of the 20th century showed a large amount of micro simulation tools (Smartest project), few of which survived however. Examples of microsimulation packages that still are being developed are Vissim, Aimsun, and Paramics.

2.4 Commercially transport model packages less suited for academic research

A large share of the current transport and traffic model packages are proprietary software products. In most cases they are developed and distributed by commercial firms. While academia and research institutions often initiated and supported the first developments of these commercial packages, their natural role to concentrate on research and innovation mostly prevents the step from prototype towards a mature product for a broader audience of model-appliers. Therefore, these prototypes commonly remain for internal use and study. Only in a few cases, the step to develop modelling packages for external users is being made.

With regards to the *use* of the transport and traffic model packages, the majority of the applications has the aim to support the policy making process of public authorities. Typical examples are:

- Strategical transport models for planning that provide insight in the long term impact of economic and demographic developments on the demand for transport and the use of both land and roads. And consequently, examine the need for new infrastructure or explore the impact of various land use scenarios.
- Microsimulation models at an operational level to assess the impact of short term measures to mitigate traffic in congested situations, for instance by optimizing traffic lights or adjusting the junction lay-out.

For these projects the primary interest of the modelling packages is to supply user friendly software with proven technology (algorithms and methods) and tools to create and use these models efficiently. The development and addition of new features in these software packages is mainly guided by the demands of the dominant user group and the cost-benefit ratio of investments. As long as the products provide sufficient functionality for their users, the emphasis of the software companies will be oriented towards developments of tools and methods that improve user friendliness and efficiency. Investments in innovative methods and algorithms, or the support of academic research, will mainly be considered if expected returns on these investments turn out positive. Therefore, developers of commercial transport and traffic model packages will not be the main source for methodological innovations.

The natural providers for such innovations are the academic and research institutions, as the development of knowledge is one of their key roles. A first and essential requirement for a transport and traffic modelling platform for research and study, is access to the internal structure of the methods and algorithms. This requirement limits the usability of proprietary software as a modelling platform in academia. The majority of (commercial) transport and traffic model packages primarily offers ready-to-use functionality, but does not facilitate the addition of new functionality by users, nor does it provide a transparent picture of how the underlying components are implemented. As the proprietary software of these packages is 'closed' there generally is no opportunity to review and use the source code.

The next section provides an overview of the main packages from commercial providers and shows their functionality and specific features, and address the mismatches between the supply of these packages and the requirements from a research point of view.

2.4.1 Transport and traffic models from commercial providers

Table 2.1 shows the main packages with a summary of their overall functionality. Four out of six providers supply the typical functions that origin from the traditional four-step model and combine it with a microsimulation module. While the microsimulation models (Vissim, Paramics and Aimsun) originally have been developed separately from the demand and assignment modelling models, providers are starting to integrate these modules into one modelling environment. Aimsun for instance, provides an approach where networks can be exchanged between the microscopic simulation and the macroscopic network.

While all of the packages provide methods from the phase of trip generation onto the assignment phase, there can be significant differences with respect to the collection of methods they offer, their quality, and their flexibility in terms of editing equations and parameters, and the opportunities to change their default methods. A good evaluation and comparison of the packages regarding both their qualities and functionalities is beyond the scope of this thesis. Moreover, one may argue the sense of an extensive survey, as it requires not only an overview of the overall functionality (for instance from manuals[4]), but also skills and experience to evaluate their results.

[4] As an indication: the Visum (version 16) and Vissim (version 8) manuals together count more than 3900 pages.

Table 2.1 Overview of some of the commercial transport and traffic modelling packages

Provider:	PTV	Cube	Caliper	Inro	Aimsun	Sistra
package:	*Visum*	*Voyager*	*TransCad*	*Emme*	*Aimsun*	*not available*
Activity and demand modelling						
Functional features						
Activity planning and scheduling	+/-	+/-	+/-	+/-	-	
Trip generation and distribution	+	+	+	+	+	
Departure time choice	-	-	-	-	-	
Agent based trips	-	-	-	-	-	
Multi-modal	+	+	+	+	+	
Route choice and static assignment	+	+	+	+	+	
Dynamic traffic assignment	+	+	+	+	+	
package:	*Vissim*	*Dynasim*	*TransModeler*	*n.a.*	*Aimsun*	*Paramics*
Microsimulation						
Functional features						
Control (e.g. traffic light)	+	+	+		+	+
En-route choice	+	-	+		+	+
API	+	+	+		+	+
Hybrid simulation (meso/micro)	+	-	+		+	-
Equilibrium assignment	+	-	+		+	-
One-shot assignment	+	+	+		+	+

Yet, we do notice some shortcomings of these model packages with respect to the requirements for research.

Firstly, many of the commercial packages provide some approaches for activity and agent based modelling, but require additional programming to provide truly activity based modelling approaches (Transport (2009)). While some of the packages provide methods to derive tour-based trips, they still use OD-matrices for the assignment and simulation of trips, and do not provide the means and methods to model and schedule daily activity and trip patterns for individual agents.

Secondly, the commercial packages provide closed software, which implies that the code is not available to the users. As a consequence, there is no opportunity to check the equations directly from the software, nor is there an opportunity to adjust or extend the methods that are supplied. In practice, the modelling methods and equations are only partly documented. An explicit description of the equations merely occurs when it is relevant for the user, in terms of offering opportunities to tune and adjust the parameters. The Vissim manual for instance, provides a description of the Wiedemann equations for car following behaviour, and describes the role and impact of the various parameter settings that can be adjusted by the user, but does not further elaborate on the lane changing algorithms. For part of the research questions, the problems with the closed software can be overcome by an application programming interface (API), which does allow customization of some of the internal workings during simulation (e.g. invoking different car following or merging algorithms or retrieving and processing of raw simulation data). Although API's hence provide access to run-time simulation data and functionality, its use is restricted to actions supported by the underlying program structure, which may not necessarily be suitable for addressing a particular research question. The dilemma for the transport and traffic model software designers is that there is no commercial incentive for opening up and standardizing their internal data and network formats and structures with other software designers, even though this may lead to much faster advances in the underlying algorithms and models and hence better products.

While this is sufficient for many applications, some of the use cases from academia do require further opportunities to adjust or replace methods. Due to the absence of a modelling package for transport research, the de facto option then is to code transport and traffic models (or parts thereof) from scratch. There is, however, a large cost involved in building such a platform. First of all, starting from scratch involves designing and coding much common and auxiliary functionality required for the model to run. Besides the obvious costs in terms of time and effort, much of this repetitive work is relatively mundane and does not contribute to scientific advances. Secondly, since code reuse is rarely a requirement in research projects, it is often cumbersome and time-consuming for others (including fellow academics) to utilize and extend models developed by their academic peers. As a result, a large portion of academic traffic simulation software exists for as long as its creator supports it, after which only a limited amount of supporting documentation (including scientific articles) is available for successive users. This situation frustrates the exchange of modelling experiences and, as a result, the implemented knowledge cannot be easily re-used and exploited by a broader audience. As scientific progress depends on communication that can be trusted, reproducibility of results from model studies is also a requirement that should be safeguarded.

This situation hampers both innovative research, due to the lack of an open and free to use transport and traffic modelling platform, and quality improvements in commercial software packages. The dilemma for public authorities (road operators, policy makers) is that they have no means (and often not the expertise) to judge whether or not a certain model is a valid choice for the projects they commission, but de facto, these are the choices they have to make on the basis of budgetary or *availability* arguments. The dilemma for the traffic simulation community and researchers in particular, is that they are forced to either *reinvent the wheel* (i.e. code their own models), or commit themselves to one single modelling environment and work around the fact that this tool cannot be tweaked fully to their needs.

2.4.2 Open source software as an alternative

The open source approach can provide a suitable alternative for commercial packages. The main audience for such an open source platform consists of users, who want to investigate the underlying methods or are looking for possibilities to adjust current or develop new functionality (algorithms and methods) that is not sufficiently provided by existing packages. Within this audience the following target groups can be distinguished:

- Education and study
  An open source system can be useful for learning and experimenting. Teachers can use the software in their courses. By exploring and altering parts of the code, students can reveal its impact on model behaviour and results.
- Scientific and applied research
  The software can be used to develop and test new or improved methods.
- Innovative companies
  The ITS and automotive industry can use the software to simulate and test innovative applications, or embed the software in their own products.

Contrary to users of the closed and proprietary model packages, the intended users have a principle interest to investigate and edit the components that are usually "under the hood". The design of the platform requires specific attention for these user requirements. While traditional model-appliers aspire tools that are user friendly and provide efficient work-flows, the *developers* that are targeted at, have a greater interest in the exploration of the underlying code of the functional methods. A basic requirement for a scientific modelling system is a free and open use of the source code. Code reuse is a form of knowledge reuse in software development that is fundamental to innovation in many fields (Haefliger, von Krogh et al. (2008)). Code reuse however, requires more than open source alone. While tools and utilities that enables them to use the model are necessary at a basic level, the following requirements are of major importance:

- A clear and well-structured design of the source code that enables users to understand and read the software;
- A modular design to efficiently modify the code, for instance by replacing modules, adjusting methods or altering parameters;
- An incremental design that allows the extension of the program with new functionality in terms of modules and methods;

2.5 Review of open source software projects

As can been seen from hosting sites, there already is a huge amount of open source software projects. Part of these projects are related to scientific research, and can be observed in various domain of interest. There are various incentives to start and develop these projects. An important incentive from scientific research is that the knowledge implemented in the code is controllable and enables re-use by other users. For reasons of convenience and efficiency, the opportunity to automate part of the work is a second incentive. It enables researchers to concentrate on science rather than the repetitive handling tasks of data preparation. But perhaps the most important aspect is the opportunity to use the software as a basis for new and innovative developments. If these developments are implemented in the existing software environment, the innovations can directly be open to other users.

A review of articles that describe and/or evaluate open source projects reveals the importance to not only concentrate on the functional requirements, but also on the *architecture and design* of the software and the *project governance*. The main items of concern that appear from these articles are:

- Maximal reuse and re-usability by using third party code and services (Roure and David (2009)).
- Knowledge of users' requirements, incremental design approach, usability testing, and knowledge of User Centred Development methods (Abras, Maloney-Krichmar et al. (2004)) are positively associated with the usability of an Open Source Software (OSS) project (Raza, Capretz et al. (2010)).
- Governance of the project: "The most effective configuration, which we labelled the Defined Community, appears to be successful due to a balance between freedom and control, as evidenced by the presence of participative decision making and the existence of a well laid-out and specific software development process" (Di Tullio and Staples (2013)).

While these requirements account for all open source software projects, a specific notion for science is that significant software contributions to existing scientific software projects are not likely to be rewarded through the traditional reputation economy of science. This provides a reason to expect the over-production of independent scientific software packages, and the under-production of collaborative projects in which later academics build on the work of earlier ones (Howison and Herbsleb (2011)).

2.5.1 Examples from the agent and activity based community

Within the research community, already some successful open source software projects for modelling transport and traffic have been initiated and developed. A large part of these initiatives regards the agent and activity based modelling. A likely reason is the lack of attention for these type of models in most commercial packages. This section provides an overview of some of the main research projects for transport and traffic models.

TRANSIMS

The TRansportation ANalysis and SIMulation System (TRANSIMS[5]) project is one the first open source projects in the field of transport and traffic modelling (Barrett, Berkbigler et al. (1995)). Its development was led by the US Los Alamos National Laboratory and is currently directed by the USDOT. As is described "The TRANSIMS Project objective is to develop a set of mutually supporting realistic simulations, models, and data bases that employ advanced computational and analytical techniques to create an integrated regional transportation systems analysis environment. By applying forefront technologies and methods, it will simulate the dynamic details that contribute to the complexity inherent in today's and tomorrow's transportation issues. The integrated results from the detailed simulations will support transportation planners, engineers, and others who must address environmental pollution, energy consumption, traffic congestion, land use planning, traffic safety, intelligent vehicle efficacies, and the transportation infrastructure effect on the quality of life, productivity, and economy." As can be seen from literature (Lee, Eom et al. (2014)) it is still being used and

[5] TRANSIMS studio download: https://sourceforge.net/projects/transimsstudio/

developed within the field of the activity based modelling and emergency evacuations, but mainly for small scale test projects involving only partial elements of its overall structure (Transportation (2009)).

MATSim
Subsequently more initiatives followed, mainly aiming to create model systems with functionality that lacks in existing models. A nice example is MATSim (Meister, Balmer et al. (2006, Balmer, Rieser et al. (2009)) that has been developed as a substitute of the traditional four-step process for transportation planning, and with the longer term goal to have an agent-based system for all aspects of urban and regional planning. While both TRANSIMS and MATSim provide a wide range of functionality, the MATSim organization actively involves the user community. The MATsim website (http://www.matsim.org) for instance, provides reports on projects, has an agenda with regular user meetings, and offers tutorials to support users who want to learn the code and become a MATsim developer. An overview of the functionality and applications of MatSim is provided in Horni, Nagel et al. (2016). As is stated in the introduction the book serves multiple goals:

- to give new users a quick start in running MATSim.
- provides more experienced MATSim users and MATSim developers with information on how to extend MATSim by plugging in available modules (e.g., the contributions), or by programming against the MATSim API (Application Programming Interface) to implement their own MATSim extensions.
- to contextualize the methods used in MATSim within a broader theoretical background. By compiling our conceptual insights on MATSim gained over the years, the book also contributes to methodological discussions on joint microsimulation of travel demand and traffic flow, a relatively new field, or, more generally, spatial demand and its congestion generation."

These last two bullets typically underpin the added value of such projects for research as they provide opportunities to add knowledge to the system and provide a transparent insight in methods.

POLARIS
One of the more recent open source and agent based packages is POLARIS (Auld, Hope et al. (2016)). An interesting aspect is the activity based demand model which is based on the ADAPTS model (Auld and Mohammadian (2009)). As is reported in Langerudi, Javanmardi et al. (2017) this module is specifically dedicated towards the planning and *dynamically* scheduling of activities. Apart from its qualities, the integration of ADAPT by reworking and reorganizing the software, is a nice example of the reuse and further development of knowledge that is stored in the programming code.

The next module is the network simulation model that combines (1) an individual traveller's route choice model, (2) a route generation model, and (3) a mesoscopic traffic simulation model that uses a variant of the Lighthill–Whitham–Richards (LWR) traffic flow model, which is a combination of a conservation law defined via a partial differential equation and a flow-density relation diagram. The demand and network module are accompanied by the Intelligent Transportation System/ Network Operations simulation.

2.5.2 Examples of traffic microsimulation projects

In the agent based modelling projects, the primary aim was the development of modelling methods that appeared absent in existing transport and traffic model packages. Contrary to the agent and activity based models, there appear less open source projects in the field of *traffic flow simulations*. Apart from *OpenTrafficSim*, the most notable exception is *SUMO* ("Simulation of Urban Mobility") that has been developed since the year 2000 at the Institute of Transportation Research (IVF) at the German Aerospace Centre (DLR). The idea behind developing an open source traffic simulation was "to support the traffic research community with an open simulation in order to: a) make the results of investigations more comparable and b) ease the testing of algorithms developed within traffic research." (Krajzewicz, Bonert et al. (2006)). The authors specifically address the opening of the source "in order to ease implementation of own algorithms" as one of the targets. Recently, a suite of applications has been added to help and perform the simulation (Krajzewicz, Erdmann et al. (2012)). As with MATSim, there is an active interaction with the user community by organizing user conferences, providing documentation and developing utility tools to increase user friendliness. SUMO is specifically designed for micro simulation, and is not designed as a more generic modelling platform.

A second example is the multi-model opens-source simulator MovSim that implements the intelligent driver model (IDM) for car following and combines it with the MOBIL lane changing model (Kesting, Germ et al.)) which has for instance been used to simulate congestions and stop-and-go waves of 15000 cross-country skiers participating the Vasaloppet ski-marathon (Treiber, Germ et al. (2015)).

2.6 Improving data exchange by implementing international standards

Apart from the mismatch between commercially available transport and traffic modelling packages and the specific needs of the research community, there is another problem that in our view adversely affects the entire transport and traffic modelling community. The problem is that most transport and traffic model packages have dedicated data input and output formats, which is mainly caused by choices from the past. As a result, it is often time-consuming to exchange data from one package to another. Currently, providers are working on improvements to facilitate a more efficient use of their data. Often this is for optimizing the exchange of data within the packages, for instance to enable an exchange of data between macroscopic and microscopic networks (PTV, Aimsun, Caliper). In addition, most of the packages provide functions to import data from external data sources, for instance infrastructure networks from OpenStreetMap or other GIS networks.

Still, there is no facility to export these data to a standardized format that could be used by other packages. Although some of these data can be read with other software (e.g. text files), this limits the user in reusing existing models (data, networks) with other packages, and requires a significant workload of data preparation for any transport study. This specifically accounts for data-intensive applications such as microsimulation networks. The following quote nicely illustrates this point (Li Zhang (2010)): "The data from TransCAD network ... are used for building the DynusT network. The process is tedious and time consuming ... The research team needed to contact the model developer in order to decode and interpret the data. For example, in TransCAD, a link's number of lanes listed as 31 does not correspond to 31 actual lanes ... Finally, correcting the movement of Junctions must be accomplished manually as well intersection by intersection." This data and network preparation phase consumes a significant share of the total effort (in working hours) spent on each study, and this does not include the actual calibration and validation of parameters and e.g. OD matrices.

The bottom line is that these limitations in the exchange of data between different models frustrate the portability of data (inputs, networks, ODs, etc.) between transport and traffic model packages. This in return may have serious consequences for the quality and validity of simulation studies. Once a model (i.e. a network + all input- and output data) for a certain case has been established with a specific package, future simulation studies are likely to be done with just that simulation package. It is, however, likely that in terms of model validity, other simulation packages may have been the more appropriate choice for this specific project. In fact, from a statistical point of view, in most cases it would be highly desirable to have an ensemble (a committee) of different models, simulate the same situation (for a significant amount of times). This of course, is not commercially feasible in the current case where the bulk of simulation costs are dictated by data and network preparation and calibration.

Part of this problem is the absence of a data standards that is required to facilitate a uniform data storage system. While there is a growing supply of GIS data all over the world, there still is no mature open data standard for transportation infrastructure. The transport and traffic modelling community therefore needs to address the standardization of all the data that goes in and out of traffic simulation models. For instance, the author from Queensland University of Technology defined subsets of data that are needed, called data marts, from a simulation user perspective, as guideline for road authorities and operators to which data is necessary to perform traffic simulations (Miska, Gajananan et al. (2011)). Such a standardization of data definitions and formats could improve efficiency, both for model developers and users.

With respect to the exchange of data between and within transportation and traffic model this section presents the following requirements:

- A matching of objects between models and data sources
- The availability of a mature and complete geospatial standard for transportation infrastructure

2.7 Conclusions

The transport and traffic modelling packages from commercial companies provide proprietary software meaning that users cannot freely examine and use the source code. This limits the use for specific user groups (academia, research). The open source software approach typically addresses these issues, as it allows the user to freely access the code. As has been shown in the POLARIS project, the implementation of existing software modules, though requiring rework, appears a feasible approach for reuse of knowledge in the form of programming code (Auld, Hope et al. (2016)).

In this thesis we provide requirements and guidelines for the development of open source software, and evaluate if and to what extend the *OpenTrafficSim* development meets these requirements. From section 2.4.2 we derive the following requirements for open source software. These requirements will be further elaborated in chapter 3:

- A *clear and well-structured design* of the source code that enables users to understand and read the software;
- A modular design to *efficiently modify* the code, for instance by replacing modules, adjusting methods or altering parameters;
- An incremental design that allows the *extension* of the program with new functionality in terms of modules and methods;
- Maximal reuse and re-usability by using *third party code and services* (Roure and David (2009)).

In addition, section 2.5 shows that a successful open source software project requires a well-balanced *governance* of the project and knowledge of *users' requirements*, usability testing, and knowledge of User Centred Development methods (Abras, Maloney-Krichmar et al. (2004)). The requirements with respect to governance and usability will also be elaborated in chapter 3.

As has been further noticed (section 2.6), transport and traffic models are not properly designed for the exchange of data. This accounts both for the exchange between transport and traffic models as well as the exchange with external data sources. A further step to improve the collaboration between various packages could be obtained by creating common definitions for objects of the infrastructure. This thesis develops a detailed design of objects of the transport infrastructure for models at different levels of detail. An important aspect of the design of data objects, is the exchangeability of data between transport and traffic models. As will be shown in the subsequent chapters, there is not yet a fully elaborated open data standard for describing the objects of the transportation sector. Part of this thesis therefore concentrates on the exploration of the requirements for an open data standard that is suitable to act as a durable basis for the exchange of data between transport and traffic models. With respect to the exchange of data between and within transportation and traffic model this section presents the following requirements:

- A matching of objects between models and data sources
- The availability of a mature and complete data format for transportation infrastructure

The succeeding chapters elaborate on these topics. Chapter 3 concentrates on the specific software and governance requirements from open source software projects, while chapter 4 to 7 further explore the design of data objects to define the road based transport infrastructure within transport and traffic models at various levels of detail.

Chapter 3

Requirements for a reusable academic transport and traffic model platform

From the previous chapter, we observed that most transport and traffic modelling software packages can be characterized as proprietary and closed source software. Usually the source code of proprietary software is not made available, which opposes the possibilities to study, modify and share the code. As its consequence, the underlying knowledge is largely inaccessible. This is in contrast to the objectives for a transport and traffic modelling system for science and research that requires the ability to access all code and the opportunity to add or modify existing code. An open source transport and traffic modelling platform appears a suitable alternative, but poses specific demands to the software in terms of quality, manageability and project governance.

In this chapter we further elaborate these requirements with respect to the software approach and project governance. The chapter starts with an overview of the main differences between open versus gated source software projects (section 3.1). As will be concluded, there a three factors for creating a successful open source projects: the software quality and platform design (3.3), the project governance (3.4), and a sufficient user participation (3.5). The choice of an appropriate software programming language is discussed in section 3.5, followed by the conclusions (3.7).

3.1 Open source versus gated-source software projects

The governance structures for software development essentially fall within the range of two extremes: *open source* and *gated source*. The key distinction between these two is that in the open source community, anyone can download, use, modify, and distribute the code. In a so-called *gated source community,* a corporate sponsor owns the code and retains the right to make project decisions. Only those who have agreed to a license with the corporate owner can download, use, or (partially) modify the code. The license stipulates that the code may only be shared with other licensees. Finally, licensees who use the code for commercial purposes must pay a royalty to the corporate owner. Between the extremes of open and gated source, hybrid forms of governance are possible.

Research among participants from both open and gated source communities shows that the possibility of opportunistic (*unfair*) actions by those holding control rights can both decrease and alter the character of voluntary participation (Shah (2005)). Gated source participants focus on the actions of the sponsoring firm, and worry that the needs of the corporate owner are more important than needs of the participants. Most notably, ownership by the firm creates the possibility that the developer will not have access to the code at a later date. Also, participants may worry that their specific needs deviate from the needs of the firm, when software related decisions are made. Many open source participants cite reciprocity as a key reason to contribute their work, trying to act fair and equitable. Moreover, many open source participators continue to create code because they simply enjoy programming. In an open source environment, participants, working at many different locations and organizations, have free access to previously developed knowledge (code), extend this by providing contributions, which evolves in a joint development of software programs (Lerner and Tirole (2002)).

The open source activity seriously started in the early 1990s with the diffusion of Internet access (Lerner and Tirole (2002)) and increased rapidly. Several open source software hosting facilities offer a source code repository. The most popular ones, in terms of the number of users, are *GitHub* (with 19.4 million projects (2016)) and *Source Forge* (430.000 projects). The increase of projects hosted by *SourceForge* from 105,000 in October 2005 to 324,000 projects in January 2014, and 430,000 in August 2017 shows the rapid growth of open source projects. Among them are more than 4,600 within the category of *Science and engineering (*January 2014). The main motivations for open source in research projects are sharing of knowledge, reproduction of results, extension of software functionality and adaptation of code (Delp, Anderson et al. (2007)).

3.2 Evaluating the requirements

In the remainder of this chapter we further focus on the various factors that determine the success of open source software projects. Ideally, a requirements analysis would elicit the main requirements and classify them into functional requirements (what it must do) and the non-functional or performance requirements (Nuseibeh and Easterbrook (2000)). These requirements need to be understandable, unambiguous, comprehensive, complete and concise. The requirements analysis must clarify functional requirements and design constraints (such as the availability of resources for development). Of major importance is the identification of stakeholders (developers and users of the system) and the goals (what should it do). In this thesis we do not apply a systemic requirement analysis, but provide a qualitative evaluation of the requirements and combine it with the experiences from two case studies.

The subsequent sections explore these requirements, and focus on three intertwined factors that determine the success of an open source project: (1) the quality of software and platform design (coding philosophy), (2) engagement of the community, i.e. the supply of sufficient opportunities to attract users, and (3) project governance and organization. These topics are elaborated in the subsequent subsections.

3.3 Software requirements and platform design

Software quality is categorized by the International Electrotechnical Commission ISO/IEC 9126-1 into six attributes: *functionality, usability, reliability, efficiency, maintainability, and portability* (Raza, Luiz F. Capretz et al. (2010)).

Functionality and usability mainly relate to functional requirements, but may have an impact on performance. *Functionality* (and related: the underlying modelling and coding philosophy of the modelling platform) is addressed in the next sections. This clearly relates to the objectives of the users and their objectives. *Usability* can be described as the "total user experience". Key usability factors are user requirements, an incremental design approach, and usability testing Raza, Luiz F. Capretz et al. (2010).

The other attributes tend to relate more to performance. From a viewpoint of *maintainability, usability, reliability and efficiency*, a high quality of the software architecture is required. Maintainability strongly depends on the software design. A component oriented approach isolates pieces of code in separate components. A software component is a software technology for encapsulating specific functional behaviour coded in the form of one or multiple objects. These objects may function autonomously, so that they can be treated as black boxes. This might for example be a module that converts a GIS map into a transport and traffic model network format, or a module that visualizes the simulation of moving objects (vehicles, pedestrians). The (re)use of software and maintenance can be supported through this modular software architecture. By isolating functionality in separate modules, only those parts of the system where functionality changes need adjustment, whereas the other modules remain as they are. A decomposition of functions into modules is useful as long as it improves usability and maintainability. There is however, a trade-off between the level of decomposition and complexity of the design rules. Every cross module dependency must be understood and addressed via a design rule (Baldwin and Woodard (2008)). An increase in *sub-modules* necessitates more communication between modules. In general, a further decomposition is useful as long as it does not lead to an exponential growth in complexity of interdependencies with other modules.

An important characteristic of modularity is that internal changes in modules do not affect the behaviour of other parts of the system. So, within a car following module, the algorithms can be adjusted, substituted or extended as long as the externally visible input and output *variables* remain unchanged. Modularity therefore enables model developers to concentrate on specific topics of interest without the need to consider or understand other parts of the system. Moreover, it creates the possibility to jointly develop models at geographically different locations by splitting tasks.

The core platform, that provides basic functionality and utilities, needs to be the stable element, in order to connect the complementary modules that are encouraged to vary and be exchanged (Tiwana, Konsynski et al. (2010)). A module can be understood as an add-on software subsystem that connects to the platform to provide functionality. Ideally, the core platform exhibits low variety and high re-usability. Its design needs to anticipate not only on current models and methods, but also on future developments, unforeseen at the time. Design rules organize and define the communications between the platform and the modules. Again, rules should be stable so that module developers can make the same assumptions at different times.

Finally, the availability of the source code of all modules offers the opportunity to learn from previous work. In essence, modularity creates *option value* with respect to new and improved approaches (MacCormack, Rusnak et al. (2007)).

3.4 Governance requirements for open source software projects

Governance is defined as the process that is responsible for the control of project scope, progress, and continuous commitment of developers: who makes what decisions about the model environment (Capra, Francalanci et al. (2008)). A central challenge is that a platform owner must retain sufficient control to ensure the integrity of the platform, while relinquishing enough control to encourage innovation by the platform's module developers. Whereas a good cooperation among code developers is crucial, too tight control structures may inhibit personal motivation to join. There are several ways to increase cooperation without raising the level of control.

The first is to eliminate as much as possible non-coding developments, for instance by self-documenting programming approaches and by creating a clear, logical and intuitive modular design. The Open Source software philosophy requires the ability to allow multiple users, possibly at different geographical locations working jointly at various parts of the software system. So without the ability to parcel out work to programming teams in different areas, the effort is likely to be unmanageable (Lerner and Tirole (2002)) (more on this point further on).

Secondly, project governance will have to be flexible over time. At the initial stage of development, the project team is primarily responsible for the decisions with regards to design and functionality of the complete platform. In this initial phase, conscious direction is necessary, since decisions on the design of the project may have far reaching consequences later on. Once there is a 'critical mass' offering sufficient functionality, the project structure and organization may change. Although rules will remain intact, in this stage they merely intend to describe the organizational processes. Formally the project team will settle on contested project decisions, however, control will be settled as much as possible by informal processes (Tiwana, Konsynski et al. (2010)), leading to a spontaneous and self-selected production of modules. A bidirectional control mechanism where active users take part in discussions with the developers will enhance the quality of the product and motivate users. This implies a movement of decision rights to those with knowledge (Langlois and Garzarelli (2008)).

3.5 Prerequisites for user participation

A critical factor for open source projects is to attract a sufficient number of users that contribute knowledge in the form of new or adjusted modules. Therefore, the involvement of new-entering members is one of the major issues. The decision of users to enter the project will be based on a trade-off between the entrance efforts (costs) and the benefits of available functionality (and implicitly knowledge). This relates closely to the need for "a critical mass of code to which the programming community can react". Enough work must be done to show that the project is doable and has merit (Haefliger, von Krogh et al. (2008)). Most of the intended contributors to transport and traffic models can be characterized as *need driven participants* (Lakhani and von Hippel (2003)), that is, the middle-ware is an instrument to be used for the development and testing of new approaches or functionality within the field of transport and traffic modelling. Some of the reasons to contribute are the availability of tailor made functions, the ability to use and modify existing methods and a proper architecture. The software should be usable, in the sense that it can be understood, learned, used and is attractive to the user (Raza, Luiz F. Capretz et al. (2010)). Only the pieces of code that are relevant for research can then be studied, evaluated, adjusted and hopefully end up in innovations. One of the advantages of open source software is that after finalizing a new module, the costs of sharing and distributing it are nearly zero, which supports the spreading of knowledge.

Participants aiming to add substantial new parts of code, require a basic level of programming knowledge in order to provide sufficient quality. Proper documentation of the software (preferably self-documenting) and accurate programming of the existing modules are prerequisites to support entering users. Furthermore, again project and platform modularity will help users and developers, as they only need to investigate the relevant modules, while the other modules may remain a black box for them. Assuming that not all contributors of programming code or users act at the same level of knowledge and experience, and certainly not at the level of a developer, attention has to be paid to the needs and expectations of the different types of users. The incremental design approach that introduces advancing levels in a gradual and progressive way is a way to make starting users more comfortable. This can be realized by providing different entry levels (Raza, Luiz F. Capretz et al. (2010)). For instance, by providing a front-end application for a traffic simulation that allows new and inexperienced users to enter and change parameter values and study the impact of these changes. More experienced users may dive into the code themselves and change formulae and methods within the source.

3.6 Choice of the programming language

Clearly, the choice of the programming language is an important issue. A review of literature shows that there appears not to be a single best programming language. As Dwarampudi, Dhillon et al. (2010) concludes, "every language has its own ups and downs. Every particular language has a purpose but can be extended or revised to accommodate the current needs of programming...".

Therefore, a selection is made by comparing and selecting the language that best suits the requirements for scientific open source software projects. This is based on multiple criteria. *First*, it needs to support modular (or object oriented) programming. An empirical comparison (Ferrett and Offutt (2002)) shows that object-oriented languages such as Java and C++ have more and smaller modules with fewer parameters, than procedural programs. *Second*, the compiled code should run on all common operating systems, making it platform independent and thus support *portability*. While the C++ code can be written platform independent, the compiled code needs to be re-compiled for every platform, the compiled Java code runs on almost every operating system. *Third*, the code should be fast and memory efficient. Here C++ is a little faster than Java, while Python appears to be significantly slower (Stein and Geyer-Schulz (2013)). Finally, as an argument to attract external participants, the language should be one of the popular programming languages (see for instance Tiobe.com). Some of the main programming languages that meet these criteria are Java and C++. With respect to a platform for Open Source Software that opts for modularity, documentation and platform independence Java and C++ are suitable, whereas other languages lack some of the characteristics that are required.

As the intended users are no professional software developers, and often do not have programming skills, the *ease of learning* and the *use* of a language has been chosen as the final selection criterion. From literature we notice that for not professional and relative inexperienced programmers (students) Java appears to be a better option than C++ (da Cruz Vieira and de Lelis (2015)). The main arguments are that (1) Java is relatively easy to learn, and (2) supports the user with built-in utilities such as garbage collection, exception handling, concurrency and does not permit the explicit use of pointers.

3.7 Conclusion

This chapter shows that the development of an open source software project requires high demands for the software, both in terms of quality and manageability. As the contributors often operate at different locations, both geographically and organizational, it is necessary to eliminate as much as possible non-coding developments for instance by automated and self-documenting programming approaches, and a clear modular design. This enables new and unexperienced users to learn and understand the software without the involvement of an experienced user. With respect to the software quality we list the following requirements (based on section 2.7 and 3.3) and classify them by their purpose:

- **R-OpenSource.1.**
  A *clear and well-structured design* of the source code that enables users to understand and read the software;
- **R-OpenSource.2.**
  A modular design to *efficiently modify* the code, for instance by replacing modules, adjusting methods or altering parameters;
- **R-OpenSource.3.**
  An incremental design that allows the *extension* of the program with new functionality in terms of modules and methods;
- **R-OpenSource.4.**
  Maximal reuse and re-usability by using *third party code and services*.
- **R-OpenSource.5**
  Software with a high code quality in terms of *functionality, reliability, usability, efficiency, maintainability, and portability*
- **R-OpenSource.6**
  A stable core platform that provides basic functionality and utilities with low variety and high re-usability

Section 3.4 shows that an open source software project requires a project team that consciously elaborates the design and functionality of the complete platform and develops a first release that is attractive for external users. At a later stage when more users enter the project, the governance may change as active users take part in the development process and gain influence at the cost of the initial project team. In fact, self-organization becomes the ideal structure as the project becomes mature. The main requirements from project governance are:

- **R-OpenSource.7.**
  A flexible project governance with conscious direction at the initial phase of the project and, once there is a critical mass, a movement of decision rights to those with knowledge
- **R-OpenSource.8.**
  Proper documentation of the software (preferably self-documenting)

As section 3.5 concludes, a critical factor for open source projects is to attract a sufficient number of users that contribute knowledge. This requires a high quality of the software code, as has already been elaborated. In addition, the following specific requirements appear:

- **R-OpenSource.9.**
  The involvement of new entering members, to attract a sufficient number of users that contribute knowledge in the form of new or adjusted modules
- **R-OpenSource.10.**
  The provision of different entry levels for more and less experienced users

The choice of the programming language is one of the decisions that has to be made at the start. Open source projects specifically require a modular approach, portability, reliability, usability and efficiency that can be offered by several languages such as Java and C++. The final choice has fallen on Java, due to its *relative* ease of use compared to C++. As the intended users are often no professional programmers, this consideration is decisive in choosing Java as the favourable programming language. While users still should develop a basic level of Java programming knowledge, well written and documented code will enable them to understand the functional parts of the code, and learn from it. Yet, it has to be realized that extending the project with new code requires further programming knowledge and experience.

The evaluation of the *OpenTrafficSim* project according to these requirements is elaborated in chapter 8 (section 8.5 and 8.6).

Chapter 4

Overview of main modelling objects in transport and traffic models

Over the past decades, transport and traffic models have become powerful tools for transport and traffic practitioners and academics all over the world. Chapter 2 showed an overview of transport and traffic model approaches, ranging from global and strategic towards very detailed and operational, meanwhile applying different methodological approaches. As a result, there is a broad collection of model types, containing various interacting methods and components. This chapter provides an overview of the main modelling objects from these models, covering both the phase of demand modelling and assignment, and explores if they are relevant for the design of the infrastructure. The analysis concentrates on road based transport and traffic, including transfers to and interactions with other modes (such as rail based and pedestrians). This results in an overview of objects that are significant for defining the transport infrastructure.

4.1 Introduction

Transport and traffic models consist of objects from the real world such as people, houses and activity locations. These models contain abstractions of these objects and let them interact to reflect the process of transport and traffic. The demand for transport services is a result of the interplay between the *demand* for successive activities at different locations and the *supply* of transport and traffic services between these locations. This results in demand for transport, in terms of trips by various transport services such as cars, bikes, trains and busses. The assignment phase of transport and traffic models reflects the actual route choice and trip making process.

In this chapter, we provide an overview of this modelling process and identify the most relevant objects and their interactions. The definition of objects strongly depends on the level of detail, and also relates to the various modes of traffic. We aim at a design of objects that allows to efficiently switch between various levels of detail. As research from the field of GIS shows, a proper data model that represents the most detailed level, is the first requirement to enable multiple levels of detail. From this base level, all other levels of detail are realized by *generalization* procedures, such as aggregation (for instance simplify intersections) and collapsing (for instance combine dual ways into a single line) (Ulugtekin, Dogru et al. (2004), Weibel and Dutton (1999)). Therefore, the first assignment is to explore and define this highest level of detail. To obtain this required level of detail, the analysis starts with an overview of transport and traffic models that represent the highest level of detail, describe its main modelling steps, and the requirements in terms of objects, relations and input-data. Figure 4.1 provides an overview of the subsequent modelling steps, starting from activity generation towards the simulation of traffic: the left side of the figure shows the modelling steps at a high level of detail reflecting the agent and activity based modelling approach, whereas the right side shows a less detailed and more aggregated approach, reflecting the traditional four stage transport model approach de Dios Ortuzar and Willumsen (2011).

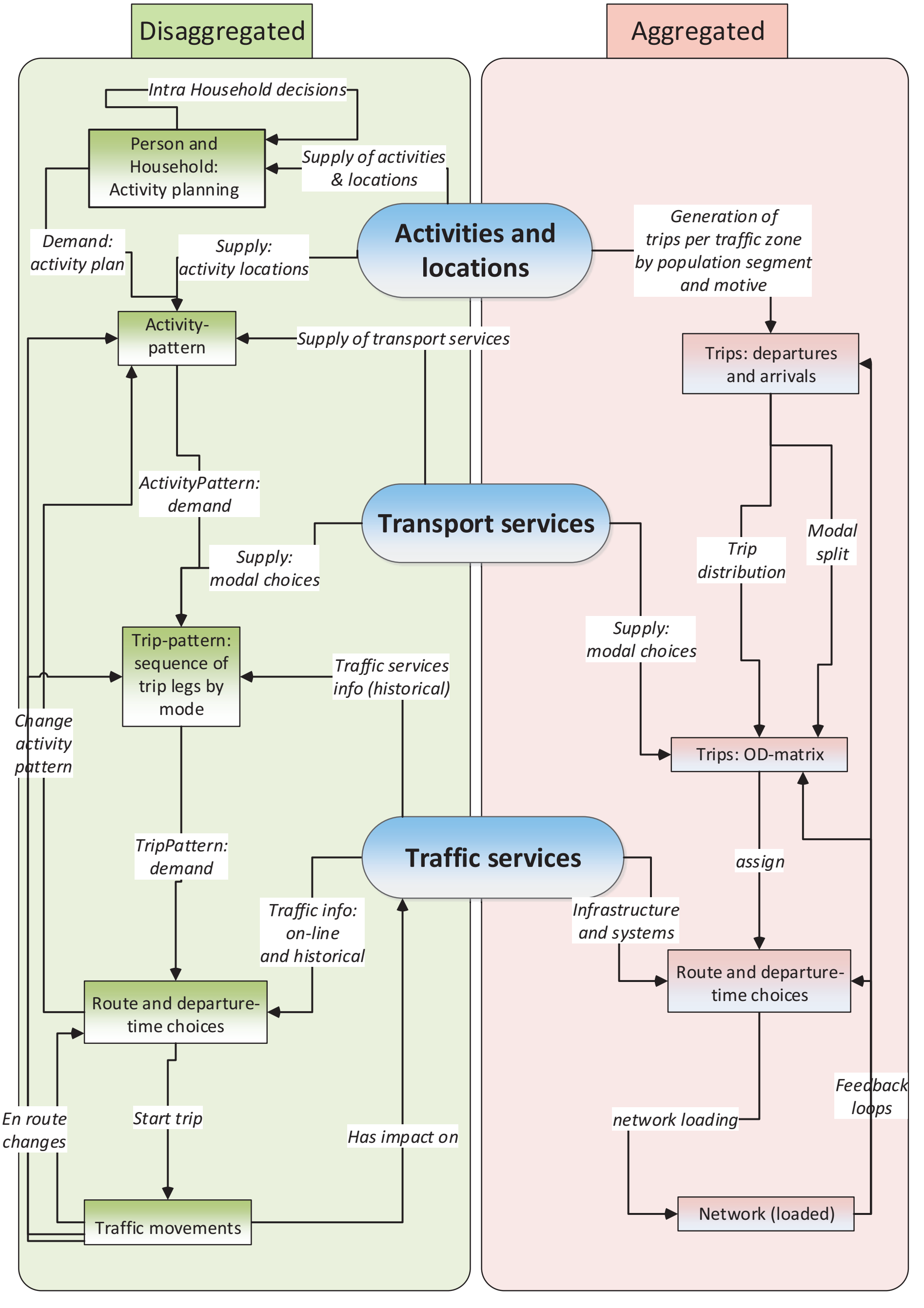


Figure 4.1: overview of the transport and traffic modelling process (left: detailed, right: coarse)

Section 4.2 considers the identification of the main objects of the transport infrastructure, by observing the modelling steps at the highest level of detail. The analysis distinguishes the phase of demand modelling (4.2.1) and the assignment of traffic, addressing vehicle based trips (4.2.2), public transport (4.2.3) and pedestrians (4.2.4). The results from these sections are combined into an overview of the core objects of the transport infrastructure, and includes the main objects to create a modelling environment for multiple modes of transport and traffic at the highest level of detail (section 4.2.6).

4.2 Core objects for modelling at a high level of detail

The transport and traffic modelling approaches can vary by (1) the *representation of behaviours and choices* and (2) the *level of detail*. Both have their impact on the representation of the objects. By overviewing the various modelling approaches at a high level of detail, we identify the relevant objects that make up the 'context' in which people live, schedule their activities and travel. The main object collections of the transport and traffic models are the transport infrastructure, the population, the facilities and their locations, and the means of transport. This analysis includes the following aspects, and points the models that represent this phase with the highest level of detail:

- *Transport and traffic demand:*
  The *activity and agent based models* provide the highest level of detail with regards to *demand* modelling, by simulating the behavioural processes of individuals with respect to activity and travel choices.
- *Traffic assignment:*
  Microsimulation models mimic traffic behaviour at the level of individual vehicles, and requires the highest level of detail with respect to the transportation infrastructure.

In addition to these aspects, specific attention is provided for:

- *Public transport*
  The modelling of trips by public transport requires a description of the transportation infrastructure that supports the transfer between various modes of transport.
- *Pedestrian traffic*
  Contrary to vehicles, pedestrians often operate in open spaces that do not provide a clear, constraining navigation structure for movement. The description of the infrastructure therefore has other characteristics than typically link based representations.

4.2.1 Transport and traffic demand

The activity based models are part of the broader family of agent based models. Agent based models are used for both the understanding and designing of *artificial* agents, to reflect human social and organizational behaviour and individual decision-making. The agent is an independent component with behaviour ranging from simple if-then rules to the complex (Arentze and Timmermans (2000, Macal and North (2009)). This allows the modelling of issues such as congestion pricing, ride-sharing incentives, alternate work-schedules and telecommuting. The limitations of the *statistically oriented* trip-based travel modelling approaches, has resulted in the development of the *behaviourally orientated* activity based modelling approach with an explicit modelling of the decision processes of individuals (Balmer, Axhausen et al. (2006), Pinjari and Bhat (2011)). Due to the complex and dynamic process of activity planning, fully operational activity based models are still scarce (Han (2009)) and remain at the level of academic research such as reported in Jayakrishnan and Kim (2006) and Auld, Mohammadian et al. (2009). Most transport models apply approaches that simplify (such as in Hunt and Stefan (2007)) or skip the activity scheduling process, and assume fixed activity plans, estimated from external data sources (Meister, Balmer et al. (2006)). Most of the operational activity-travel demand models have been developed to predict changes in activity-travel patterns in relation to land use and transportation policies. Typical examples are *MATSim* (Balmer, Rieser et al. (2009*))* and *Albatross (*Arentze and Timmermans (2000*))*. Short term dynamics such as the (re-)scheduling of activities or non-stationary environments are addressed more recently and implemented in models such as *Feathers* (Arentze, Timmermans et al. (2006), Han (2009) Gupta and Vovsha (2013)).

Overviewing the modelling process, the following types of choices with regards to activities appear (Balmer (2007)):

- Activity type choice;
- Activity chain choice (order of activities);
- Activity starting time choice;
- Activity duration choice;
- Activity location choice;
- Group composition choice (who should I take along?).

While many of these choice processes incorporate information about the available transport and travel services and routes, the development of ICT and automated vehicles entail additional choices (Liao, Arentze et al. (2014)):

- Activity substitution of location based activity by ICT based counterpart;
- Fragmentation of activities, where part of an activity can be replaced (part-day homeworking);
- Multi-tasking where activities are combined, such as working in the train or online shopping during work.

The activity based models specifically focus on the modelling of activities by persons. As person decisions are often strongly related to intra-household relations, the *Household* is one of the key-objects to represent the population (Gupta and Vovsha (2013), Katoshevski, Glickman et al. (2015)). To explain the activity decision there is a growing interest in social networks that relate social interactions with activity patterns (Arentze and Timmermans (2006, Hackney (2009)).

An overview of the main objects of the activity based models is presented in Figure 4.2. The names of the objects are based on the MATSim (Multi-Agent Transport Simulation) model (Horni, Nagel et al. (2016)). A comparable structure is found in the agent based model Feathers (Bellemans, Kochan et al. (2010)).

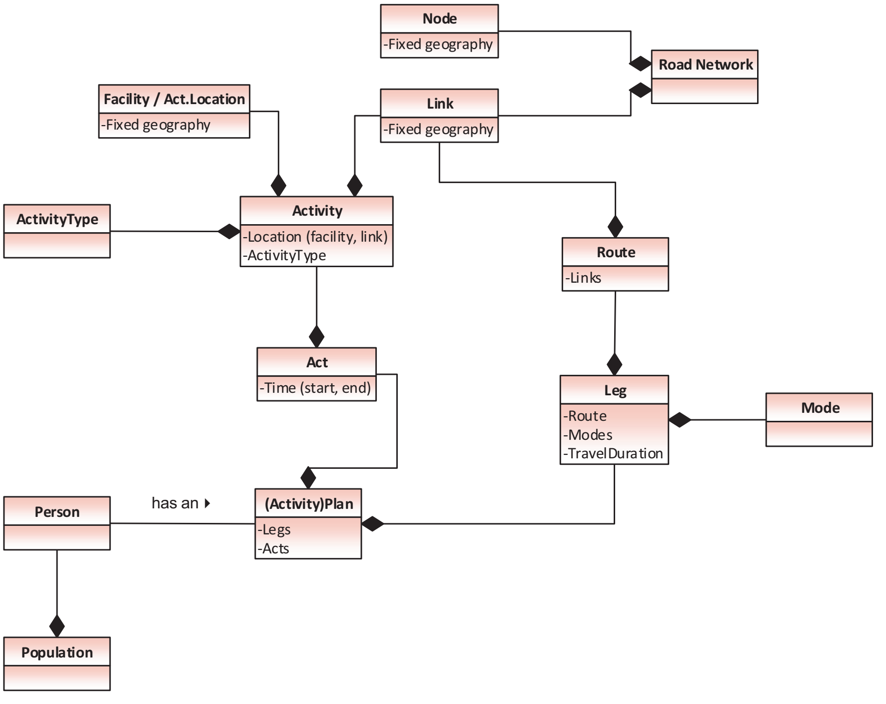


Figure 4.2: Pseudo UML-diagram with the main objects of an activity based model[6/7]

In the initial phase of activity planning *Persons* combine their *Activities* into an *ActivityPlan*. Most activities are carried out at specific locations, for instance sleep at home and work at a factory. The *Facility* defines the location (for instance a building or an area) where activities are performed. In addition the these geographic characteristics, the *institutional* context is important, as it describes where and especially when (opening hours) activities can be conducted (Arentze and Timmermans (2000)). This context is further defined by *ActivityTypes*. While a description, such as work or recreational, is helpful to indicate the type of activities, it does not necessarily provide the attributes that are relevant for activity choice and scheduling. Properties that do provide relevant information for the *activity types* are *frequency*, *duration*, *temporal flexibility*, *spatial flexibility*, *function*, *financial importance* and *interpersonal flexibility* (Doherty (2006), Mohammadian and Doherty (2006)). The actual relationship between these attributes and a certain activity is not necessarily fixed, but relates to cultural and technological circumstances. The spatial and temporal flexibility of work for instance, significantly increases with the availability of telecommunication, which in turn provides more flexibility for scheduling work related activities. To accomplish these activities, every person chooses a suitable way to travel between the successive activity-locations (or Facilities). The planning of activities requires information about the availability of *transport services,* the

[6] names of objects based on MATsim: http://www.matsim.org/the-book

[7] This, and all subsequent diagrams, looks similar to the UML representation, but does not follow the UML conventions

activity locations and the *activity types*. The *ActivityPlan* defines the sequence of activities, and every trip (*Leg*) to move between the activities. A *Leg* is composed of one or more transport services. The actual modelling of these activity plans is complex, as there are many relations, uncertainties, incomplete knowledge and/or data, and recursive processes that all affect the final choices with respect to route choice and departure time. The level of detail of a model determines the composition of a Leg. A trip to work might for example start by walking from home to the location of the car, drive to a train station, park and walk to the platform, continue by train, and finally walk to the office. The simulation of trips at such a level of detail, would require the combination of a pedestrian, road and rail based simulation (Balmer, Rieser et al. (2009)).

4.2.2 Traffic flow operations

While the activity based models provide a detailed approach to model travel and trip demand, the modelling of travel and traffic services and their operations, is a topic that is not dedicated to the activity based approach. The execution of these activities requires decisions about the trips between the successive activity locations. The main choices in the trip preparing process within transport and traffic models regard the combination of the mode(s) of transport, the departure time, and the route alternatives. The choice of the mode(s) of transport is primarily based on (1) the availability of traffic services (private vehicles and public transport), (2) knowledge and perception of these alternatives, (3) personal trip preferences and expectations, (4) the valuation of the alternatives (expressed in terms of *utility*) based on variables such as trip duration (including variations and uncertainties), costs, and comfort, and ultimately (5) an assessment of the preferred alternative. For chained trips with multiple modalities, the modelling of this process requires insight in a set of trip alternatives, characteristics of the separate trip legs and transits, as well as a valuation of the (dis)utility of all components (Ye, Pendyala et al. (2007), Ciari, Balmer et al. (2008), Cats, West et al. (2016)).

After choosing the preferred trip alternative, route choice applies an approach that is comparable with the mode choice process. The *global*, or *pre-trip*, route choice, is generally based on prior knowledge from (digital) maps, traffic information and previous experiences. During the trip, changes of the traffic situation, such as congestion, may lead to adjustments of the route (Gao, Frejinger et al. (2010)). This is often denoted as an *en-route* change of a route, also called *adaptive* or *local route choice* in pedestrian modelling (Duives, Daamen et al. (2013, Hoogendoorn, van Wageningen-Kessels et al. (2015)). Specifically for *public transport* with chained trips and multiple modes, there are numerous unpredictable phenomena that affect both mode and route choices (Gentile, Florian et al. (2016)). This may occur at transit points where multiple transport alternatives are available. As departure times often deviate from scheduled departure times, the choice of the specific trip alternative will be uncertain and depend on daily varying circumstances. Another example of such uncertainty is the availability of rental bikes at PT-stations. If all bikes are in use, this will require to look for an alternative and thus the choice for another mode and route to a next destination.

In the final stage of trip executions, traffic operations cause *movements*. The modelling of movements and *delay in movements* is one of the key challenges in traffic flow models. The movements of vehicles and pedestrians, in the remainder denoted as *traffic units*, are influenced by (Rieser (2010), Wilkie, Sewall et al. (2015), Cats, West et al. (2016))):

- The characteristics of the drivers/pedestrians;
- The position of other traffic units;
- The physical infrastructure (such as curves, slopes, speed bumps);
- Traffic rules (such as right of way, legal speed limits and prohibitions);
- Vehicle capacity (public transport: in case of boarding).

Figure 4.3 presents an overview of the most significant situations that have an impact on speed of vehicles and traffic throughput.

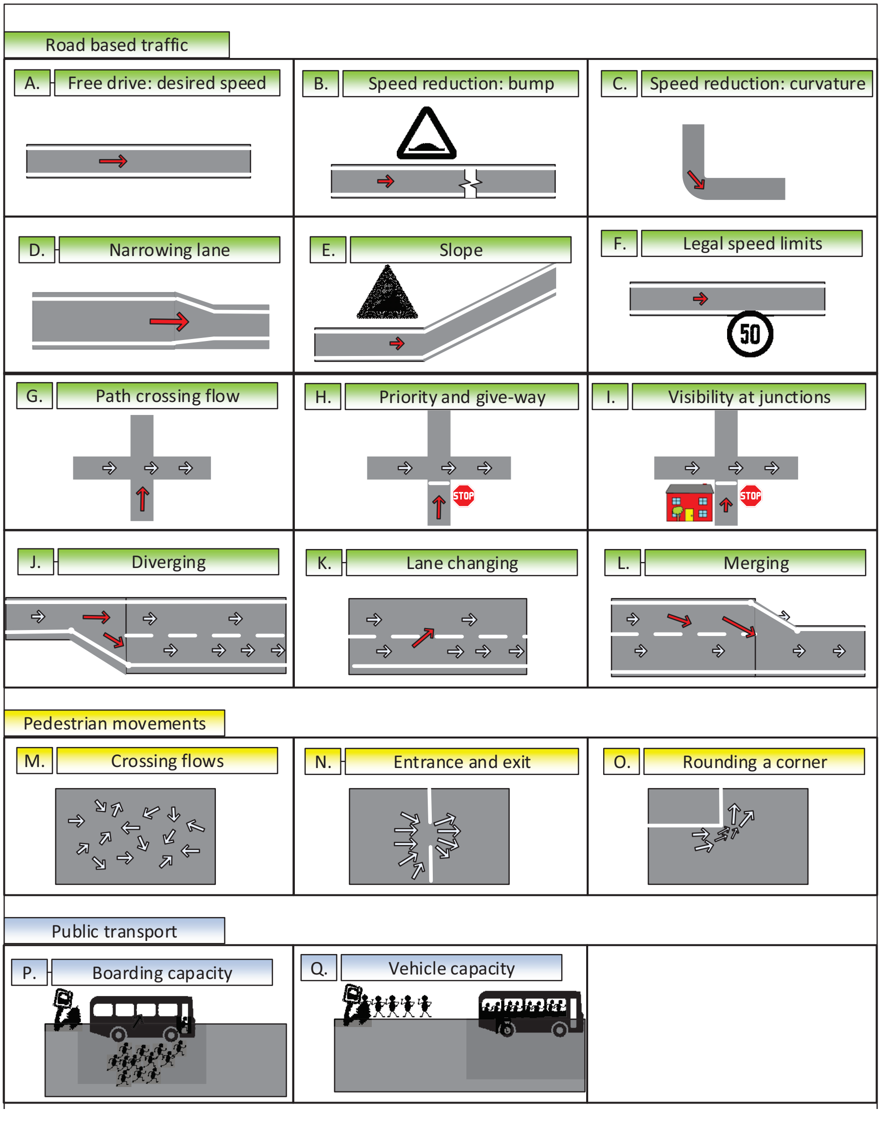


Figure 4.3: overview of typical situations that have an impact on speed and capacity (pedestrian movements based on Duives, Daamen et al. (2013))

Specifically, regarding the transport and traffic operations, the microsimulation models provide a high level of detail with respect to the choices of movement. The agent based simulations distinguish the agent (person) that chooses its mode, route and driving choices individually. By distinguishing the vehicle and its driver, the impact of human behaviour on driving can be modelled explicitly. This would for instance be relevant for cases where human behaviour deviates from average conditions, such as evacuations with panicking, or cases where driver tasks are handed over to the vehicle, and the role of the driver changes. For future situations, the role of in-car systems will become increasingly important, and in case of self-driving vehicles may fully take over the role of the driver.

The behaviour of traffic in all of these models is mimicked by simulating vehicle-driver combinations at an individual level. The interaction with other vehicles and the infrastructure requires a detailed representation of traffic behaviour. To understand driving behaviour and subsequently model it, driving decisions are often distinguished into a strategical, tactical and operational level (Hoogendoorn and Bovy (2004, Kesting, Treiber et al. (2008)). At a *strategic* level, knowledge of the route, either from the map, previous experiences or from a navigation device, provides information about choices at a link and lane level, specifically when nearing decision points such as on- and off-ramps and turning movements at junctions. In the *tactical* stage the decision to make a lane change is prepared by looking for a suitable gap at the parallel lane(s) and adjust speed to enable a proper lane change. The *operational* decisions merely are intended to carry out the tactical decisions safely or anticipate on unexpected behaviours from other cars.

All of these actions and decisions are based upon and influenced by multiple factors. The tactical and operational decisions depend on the driver and vehicle (system) characteristics, his awareness of the situation, and the behaviour of other vehicles in his vicinity. The awareness of his situation depends on information that is provided by various sources. At first, there is the mental map of the driver about his route, based on previous experiences (Zheng (2014)). The mental map can be supported and updated in various ways, for instance by in-car navigation devices, that provide drivers support on necessary lane changes or turn movements. Additionally, traffic signs, road markers and information panels along the road provide visual information about routes, turnings, speed limits and accessibility, and thus need to be part of the definition of the modelled infrastructure. The role of in-car systems is of growing importance, as ITS developments will provide increasing amounts of information, for instance about current and future traffic conditions (congestion) or expected time to green at signalled junctions.

For specific cases, for instance from the automotive industry and ITC research institutes, supplementary data are required (Reichardt, Maurer et al. (2011)). An example is the simulation of new communication technologies, where the propagation of waves from wireless LAN, requires a representation of the environment such as buildings and trees.

Figure 4.4 shows the main components of micro-simulation models such as Vissim, Aimsun and Paramics. Compared to the components of the activity based models (see Figure 4.2) it provides a further refinement of the roadway network with an explicit description of the driving lanes. On the other hand, less attention is paid to the description of the facilities where trips depart or arrive, as well as the person who is making the trip.

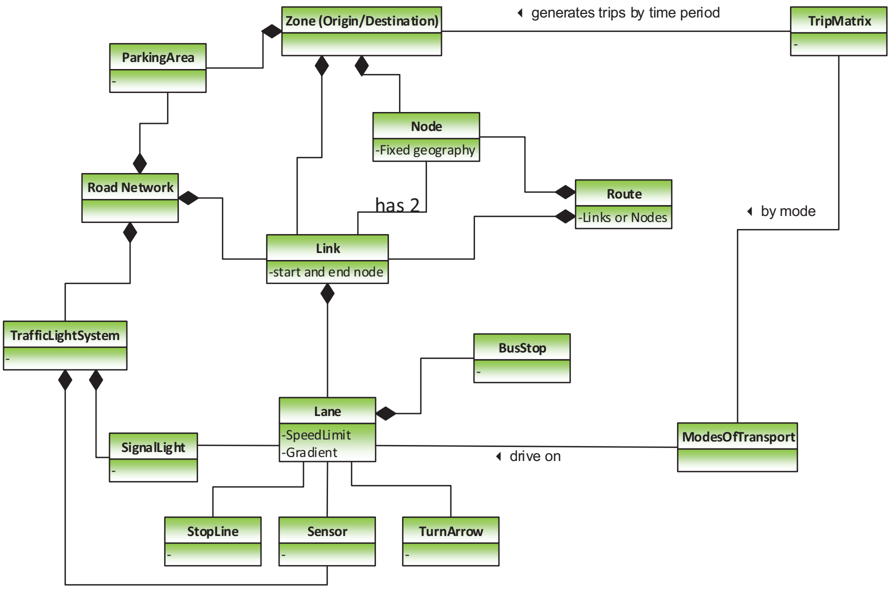


Figure 4.4: common components of microsimulation traffic and transport models

4.2.3 Public transport

Transit assignment or simulation models are used for predicting the distribution of passengers over a transit network (Cats, West et al. (2016)). In addition to modelling private traffic, they require additional objects to represent their physical environment. Specific characteristics of public transport models that deviate from private traffic are:

- Fixed routes and time-schedules of vehicles
- Duration of the stops of vehicles that is influenced by the number of boarding and alighting passengers
- Passenger capacity of the vehicles

In addition to congestion that is caused by capacity constraints at roads and junctions, public transport adds congestion phenomena that are caused by a high density of passengers on board of a vehicle. One of the effects of congestion is crowding, which refers to lower on-board comfort as the on-board load increases (Cats, West et al. (2016)). In case the bus is full, boarding passengers are denied and have to wait for a next bus or look for other alternatives. The modelling of these phenomena at a high level of detail is presented in Cats, West et al. (2016). The process of boarding and alighting requires a dwell time function that reflects the boarding and alighting capacity of the corresponding vehicle type. Relevant attributes of the vehicle type are the number of doors, payment procedure and boarding regime, the number of seats and the total on-board capacity.

Figure 4.5 shows the framework for the bus transit system based on the hybrid mesoscopic-microscopic traffic simulation model Mezzo (Toledo, Cats et al. (2010)) and the MATSim framework (Balmer, Rieser et al. (2009)). As can be seen, the description of the transport infrastructure is extended with typical object for public transport.

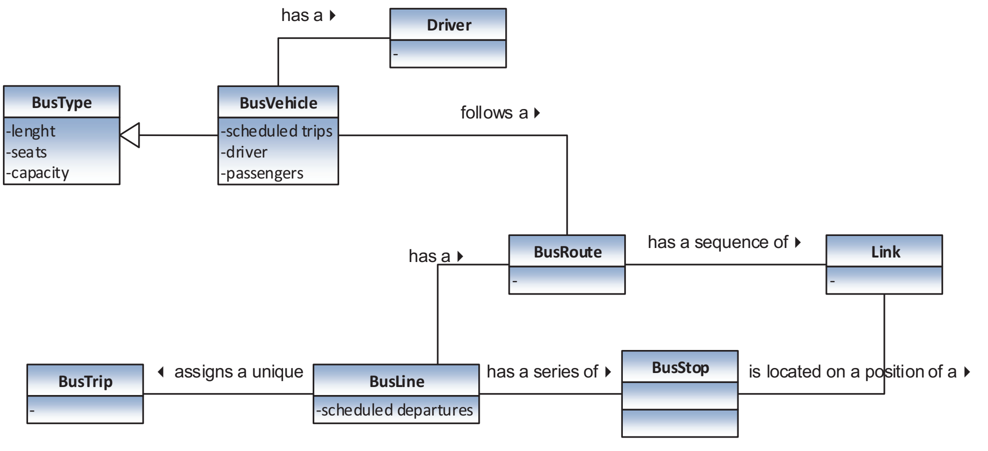


Figure 4.5: object framework for public transport

In addition to the transit trips, transfers to, from or between public transport vehicles are generally part of the entire trip from origin to destination. These transfers can be done by all kinds of modes and requires the modelling of transfers between these trips.

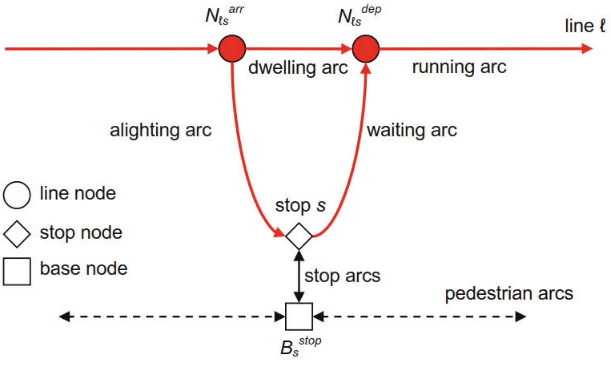


Figure 4.6: objects to model public transport transfers (source: Gentile, Florian et al. (2016))

The transfers generally contain the following phases (Gentile, Florian et al. (2016)):

- Accessing a transit stop (usually by walking or bicycle).
- Waiting at the stop for the dwelling vehicle
- Boarding a dwelling vehicle
- Travelling in the vehicle (on board) through a sequence of stops;
- Alighting the vehicle at another stop;

Figure 4.6 shows the components of a transfer that are typically relevant for the valuation of the separate phases, and are used to assess the costs of all trip components. In addition to walking time, a walking discomfort coefficient may be used, that varies by personal characteristics, for instance to model a higher discomfort for elderly people.

At the highest level of detail, the modelling of such transfers requires a pedestrian simulation with a detailed representation of the locations where public transport vehicles arrive and leave. Typically, a platform at a train station can be represented by a *HorizontalLevel* (see Figure 4.7), where it "touches" the network based rail infrastructure along this platform. A description of the topology of these kind of objects in addition with proper semantics, is a first requirement to facilitate the modelling of transfer locations between multiple modes. In addition to these pedestrian transfers, other modes such as bicycles or cars provide a link with public transport and onward journeys. In the context of modelling the physical infrastructure, this requires specific attention for the definition of the accessibility of roads by type of vehicle.

4.2.4 Pedestrian traffic

Literature on pedestrian modelling and navigation, reveals a representation of the infrastructure that deviates from most other modes of transport. Rüetschi (2007) classifies car, rail and public transport trips as primarily *Network* based, because they focus on node, link and lane structures. Pedestrians often operate in open spaces that do not provide a clear, constraining navigation structure, allowing for far less constrained movements. The movements of the pedestrians depend on their plan and goals, and the perception of other pedestrians and the environment (Campanella (2016)). The environment consists of the infrastructure and information via signs or communication, and comprehends both the walking areas, the obstacles and the non-walking areas. In an outdoor environment, these so-called *Scene Space* based movements are typically influenced by infrastructure and urban design, and are dominated by nested open spaces. Environments like stations (with its halls, squares, platform areas, etc.), airports, harbours, shopping malls, or public parks are samples of these walk environments. Many of them require a 3D representation, for instance because of different levels. Escalators and stairs have the function to connect these levels, and, in case of escalators are unidirectional (up or down). The environment can be specified as follows (Schaap, Zlatanova et al. (2010)):

- Provide a sufficient number of objects to be included in the Scene Spaces
- Store spatial characteristics of objects relevant for pedestrian use of Scene Spaces
- Define accessibility constraints of Scene Spaces
- Combine indoor and outdoor Scene Spaces in one geo data model
- Connect Scene Space with the Network Space

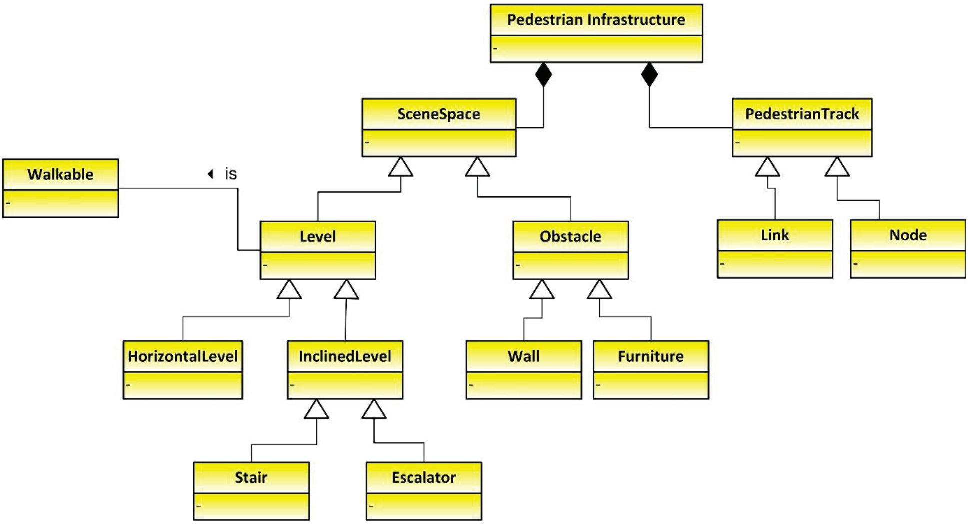


Figure 4.7: Class diagram of the infrastructure in for a pedestrian model (names based on the Nomad pedestrian model)

Figure 4.7 shows the basic objects to model the pedestrian infrastructure. These objects can be further extended and specified to construct a *Scene Space*.

4.2.5 Future developments of the transport and traffic system and their impact on the modelling objects

While the analyses of the previous sections are based on models that mainly reflect the current situation of transport and traffic, the ongoing and successful further development of the computerization of vehicles up to fully self-driving vehicles, has the potential to change the transportation system drastically (Milakis, van Arem et al. (2015)). Two of the major trends with respect to the technological developments are:

- Wireless communication technologies that increasingly enable the development of systems that support interactions between cars, and between cars and the transportation infrastructure;
- Development of in-car systems that support and gradually hand over driving tasks from the driver to the vehicle.

With respect to transport and traffic models, some of the main changes of the self-driving vehicle and the ability of enhanced vehicle communication are (Milakis, van Arem et al. (2015)):

- Travel choices:
    - In addition to the traditional role (move to the next activity location), self-driving vehicles enable people to perform activities while driving. This changes part of the travel costs expressed by their value of time (VOT) and may change the role of transport services significantly. On the other hand, the costs per distance unit

of the self-driving vehicles are still uncertain which makes it impossible to estimate total costs.
    - The role of public transport by road may change significantly, specifically when self-driving vehicles are able to supply dedicated transport services. Instead of fixed lines and time schedules, flexible systems with ride- and vehicle sharing emerge.
    - With the opportunity of the vehicle sharing alternatives, the availability of private means of transport (car) within households is not restrictive for activity choices anymore.

- Driving behaviour and control
    - Decoupling driver and vehicle: the role of a driver may become absence.
    - Driving behaviour of automated vehicles deviates from human driving, for instance by accepting shorter headways. This may lead to an increase of lane capacity.
    - New traffic management and control systems may emerge. Examples are controlled junctions without traffic lights, optimized routing advice and guidance to the nearest available parking place.

- Location choice and land use
    - Vehicle and ride-sharing reduces the fleet size and requires less parking capacity. Land use of parking can be replaced by other functions
    - Enhanced accessibility of rural areas may attract people to live, work and recreate at more distant locations.

While these developments may have a significant impact on the methods of transport and traffic models, we only expect a minor impact on the design of model objects is . With respect to the transport infrastructure, the main change appears the introduction of self-driving vehicles, which in the longer term may lead to changes in road geometry (smaller lanes), and the introduction of dedicated infrastructure for self-driving vehicles. These infrastructural changes can still be modelled with current models. A second change regards the modelling of the means of traffic, which necessitates the possibility to implement automated driving systems as components of a vehicle. Again, these opportunities are already possible in current transport and traffic models. Finally, the potential changes of the public transport system with fixed lines and schedules into more flexible taxi-like and car sharing systems can still be modelled with the current objects.

4.2.6 Summary of the core objects

The integral infrastructure of the transport and traffic models is summarized in the component diagram (Figure 4.8). The various colours relate to the figures in the previous sections, and show the various modes of transport: pedestrian, public transport and road based private transport and their mutual relationships. For multi modal trips, a transfer of mode requires that the different types of infrastructure are connected to each other at the transfer locations. For a trip-leg this could be a sequence of: (1) walk from the house to the car-park, (2) drive over the road infrastructure, (3) park near a train station, (4) walk to the platform to enter the train etc... The modelling approach determines the required level of detail for the description of the infrastructures. The description of the trips and their means of transport are described by generating a series of trip-legs (by person) by mode.

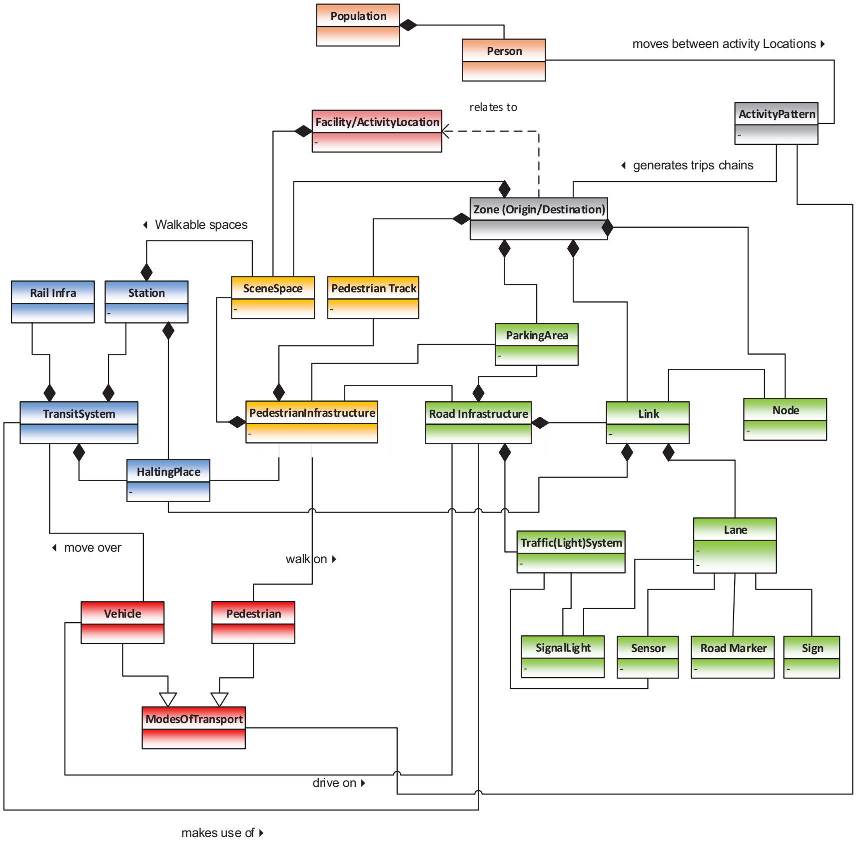


Figure 4.8: Overview of core components of current transportation and traffic models

The components of the transport and traffic models can be grouped into the following categories:

- The *population* making *trips* by modality;
- The *means* of *transport*;
- The *activity locations;*
- The *transportation infrastructure*.

4.3 Conclusions

The modelling of traffic demand starts with the choice of a person's *activities* and ends up in *trip patterns*, that describe the successive activity types and their locations. The highest level of detail describes individual agents (persons), individual activity locations and their traffic access points (parking facilities, or entrances).

The next modelling step describes the trip choices with regards to the *modes of transport* and their *routes*. At the highest level of detail these are individual trips between activity locations. To provide access to all of these locations, a detailed representation of the transportation network and traffic services is required. The transportation network should provide connectivity between the roads and junctions, to enable the construction of a graph to create feasible routes. Additionally, the mode and route choice require insight in the (dis)utility of the trip alternatives. As there can be differences in costs and the valuation of time and comfort for different trip segments, detailed insight in these separate parts is therefore required.

The actual *execution* of these trips leads to choices with respect to movement. At the highest level of detail, microsimulation models represent the behaviour of individual agents which requires a detailed description of the transportation infrastructure. The modelling of the longitudinal and lateral movements of road based traffic requires a detailed representation of the lanes, including lane markings, sensors, and their geometry (including curvatures).

The subsequent chapters further explore the requirements for the design of the transport infrastructure for road based traffic. This includes the exchange of data between models and external data sources. From the previous sections, we have identified a first list of aspects that are relevant for the design of the modelled transportation infrastructure:

- A representation of the activity locations;
- Description of the access points towards the activity locations;
- A representation of the traffic services;
- Connection between the means of transport and the individual agents (persons);
- A detailed representation of the transportation network, including lanes descriptions;
- Connectivity between the roads and junctions, to enable the construction of a graph;
- Ability to determine costs for different trip segments, including the valuation of time and comfort.

Chapter 5

Requirements for the design of a multi-scale transport infrastructure

After determining the core objects for transport and traffic models at the highest level of detail, this chapter focusses on the requirements for the design of the transport infrastructure for road based traffic. An important aspect of this analysis regards the impact of modes of transport, the variation in the level of detail of transport and traffic models, and its consequences for the definition of model objects. Furthermore, the exchange of data between models and external data sources is crucial to feed the models with data. Chapter 6 concentrates on this topic. In chapter 7, the results and requirements from both chapters are used to elaborate the design of objects from the transport infrastructure.

5.1 Introduction

The object design of the transport infrastructure for road based transport and traffic models relates to three main factors.

Firstly, the methods of transport and traffic models determine (1) the objects, (2) their functional requirements, and (3) the relationships between the objects in order to represent the transport infrastructure. To avoid an extensive survey, exemplar representations of objects that cover the broader collection of road infrastructure objects will be used, to indicate how similar types of objects can be implemented in the network design. The possibility to model multiple modes of traffic will be part of this exploration.

Secondly, the design will take into account interactions of the road based transport infrastructure with other non-road based modelling environments. Pedestrian (crossings) or waterway models (bridges) for instance, may interfere with the road infrastructure. Moreover, transfers between different road-based modes that require pedestrian movements, or intermediate transfers by ferries, should be part of the design. To provide the opportunity to combine (parts of) these infrastructures, the network design meets these requirements of relevant non-road based trips. Section 5.2 discusses these topics and presents the relevant objects of the transport infrastructure for road based transport and traffic models.

Thirdly, data are needed to actually shape the objects of the transport infrastructure with their geometry and attributes. The availability of data and its format are therefore important factors to consider in the design process. As will be elaborated in *chapter 6*, the use of (preferably) open and standardized data formats will be an important constraint.

The ability to switch between levels of detail of the transport infrastructure is one of the main challenges within this design process. This level of detail depends on the requirements of the various transport and traffic model approaches. Within the spectrum of road based transport and traffic models, the agent and activity-based, and the micro-simulation approach apply the highest level of modelling detail, while for many other approaches (mesoscopic, macroscopic) a less detailed description of the transport infrastructure is sufficient. More global approaches, such as the network transmission model, require data about areas instead of roads. Figure 5.1 shows these various types of application and their level of detail. For all of these levels, a description of the transport infrastructure is required.

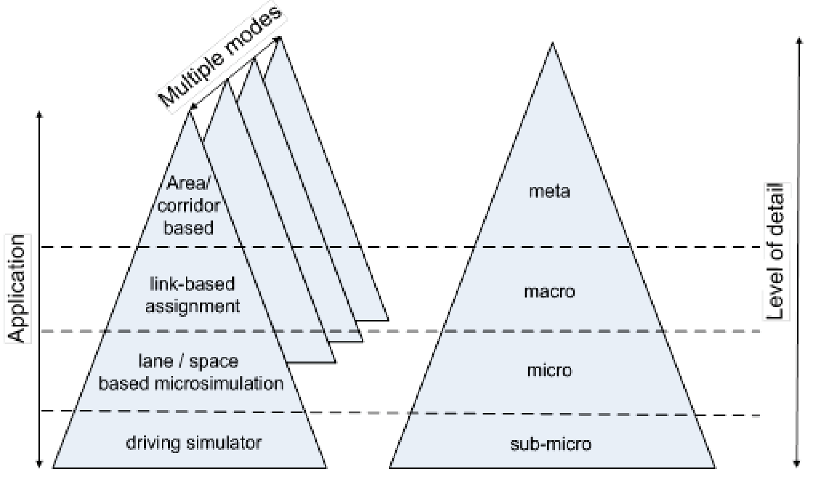


Figure 5.1: Switching between levels of detail while combining various modes[8]

As has been described in chapter 4, a proper data model that represents the transport infrastructure for microsimulation models at a high *level of level,* is a solid base. From this base level, the less detailed representations of the infrastructure in coarser transport and traffic models are realized by aggregation (Ulugtekin, Dogru et al. (2004)). This requires the application of generalization operators to switch to lower levels of detail (Kilpelainen (1997)).

Figure 5.2 visualizes this process, and shows an aggregated link based representation of the road infra that is derived from the highest level of detail. While the geometric details such as length and width can be generated, this does not account for attributes whose values depend on traffic behaviour. *Capacity* for instance, is a variable that results from both topographical and behavioural features. The opportunities to switch between levels of detail are elaborated in section 5.3.2 and 5.4.

[8] *based on brainstorm sessions in the OpenTrafficProject team*

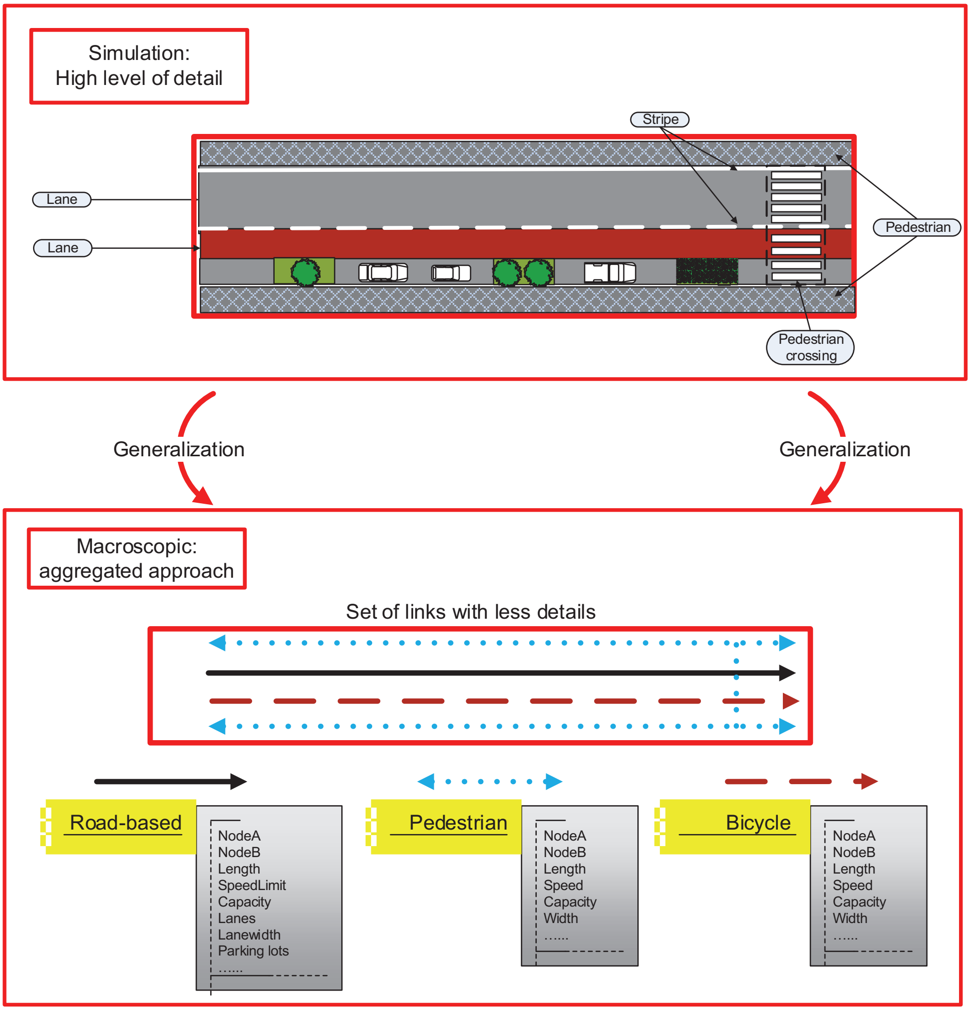


Figure 5.2: From high level of detail to coarser representations of a road stretch

Before presenting the conclusions, section 5.5 explores the opportunities to add *more* detail. Opposite to aggregation, a second principle for changing the level of detail is *extensibility*. This principle that originates from the object oriented software design, is a systems design principle, where the implementation takes future growth into consideration. This allows an extension of objects, with new attributes and functions.

5.2 Main requirements for objects of the road infrastructure at the highest level of detail

The modelling of the trip making process, which includes choices about activities, departure time, mode, routing and the movements of vehicles, requires information about the *road infrastructure, including the access to the activity locations* and the *means of transport* that utilize this infrastructure. Based on the analysis in chapter 4, and on Ruas, Gold et al. (2008), Wilkie, Sewall et al. (2015), Zhu and Li (2008), Hong, Jianping et al. (2008), Willemsen, Kearney et al. (2006, Zhu and Li (2008) and van Lint, Schakel et al. (2016), the following topics appear relevant for the design of the transport infrastructure at the highest level of detail.

1. Interaction of vehicles with the infrastructure

For a microsimulation the following vehicle characteristics appear relevant:

- The type of vehicle such as person car, freight or bus;
- Static vehicle information such as length, width and height;
- Attributes that describe the dynamics of vehicles such as maximum speed and acceleration/deceleration;
- The person who is driving;
- On board units that either enhance the vehicle or assist the driver.

2. Determine trips and their costs

This aspect distinguishes two subjects. The first concentrates on a description of the relevant infrastructure characteristics that determine the costs of route-alternatives:

- Features that provide localized information, such as a sign that indicates a legal speed limit;
- Range attributes that supply information for a certain region of a road;
- Accessibility of roads by vehicle types and/or user groups;
- Turning opportunities and prohibitions at junctions;
- Characteristics of junctions such as priority, give-way and traffic light control.

Secondly, the model requires the opportunity to determine route-alternatives, and includes:

- The ability to create routes between the activity points, which requires a valid representation of the topology of the roads and junctions.
- The opportunity to include multi-modal trip legs and transfers between modes;

3. Ability to model vehicle operations

The third aspect partly overlaps with the previous aspect.

- A description of the relevant infrastructure characteristics and situations that specifically determine driving/vehicle behaviour:
    - Features that have an impact on driving behaviour, such as a speed bump or a traffic light signal;
    - Lane markings that signify whether or not lateral movement are prohibited;
- The representation of traffic management systems:
    - The functional units such as traffic light control;
    - Features of these units such as signal heads and sensors;
- 3D information for a rich description of both the road and the environment to:
    - Derive the visibility at a junction;
    - Estimate the visibility in case of overtaking;

4. Public transport, transfers between modes and access to activity locations

Elements that relate to the activity choices or relate to changes to other modes of transport relate to:

- The identification of access points for the activity-locations;
- Characteristics of parking opportunities;
- Transfers between modes or trip legs;
- Halting locations for public transport.

The succeeding sections further explore these topics, and extract the main requirements for the design of the infrastructure for a microsimulation model. Section 5.2.1 describes the interaction of vehicles with the infrastructure and its consequences for the design of the infrastructure (topic 1). Section 5.2.2 discusses the requirements to determine trips and their costs (topic 2) as well as the modelling of vehicle operations (topic 3) and derives additional requirements for the road infrastructure. Section ? regards the transfers between modes and access to activity locations (topic 4).

5.2.1 Interaction of vehicles with the infrastructure

The relationship between the road infrastructure and traffic behaviour is not unambiguous. There appear large variations in traffic composition and behaviour between different countries and regions. In most developed countries, traffic is quite homogeneous with a large proportion of cars that behave according to the rules of the road. Most drivers intend to keep within their lane and follow priority rules on junctions. In less developed countries traffic is often more heterogeneous, and even in cases with a large share of cars there is often less lane discipline. Under these circumstances vehicles have the freedom to occupy any lateral position on the road leading to more frequent and unwarranted lane change manoeuvres (Bains, Ponnu et al. (2012)). In such cases lateral behaviour is essentially determined by the physical space of the road and the position and movement of neighbouring vehicles (Bains, Ponnu et al. (2012, Schönauer, Stubenschrott et al. (2012)). The same kind of behaviour is apparent in urban environments with a Shared Space approach, where multiple modes of transport interactively decide their moving behaviour, as there are no lane markings and no explicitly defined driving directions.

For the design of the transportation infrastructure, in a situation where traffic behaviour shows lane discipline, the requirements for the modelled infrastructure depend on the type of traffic that is regarded. Three categories of road based traffic are distinguished:

1. The primary traffic units that use roads:
    - Vehicles for:
        - private transport such as car, taxi, freight carrier, coach
        - public transport such as bus or trams (when driving lanes and rail-tracks share the road)
    - Other motorized means of transport (motors, scooters, mopeds).
    - Non or partly motorized two-wheelers (bicycles, e-bikes)
    - Pedestrians: at roads that allow pedestrians such as in shared space

2. Traffic of conflicting infrastructures (such as pedestrian crossing and level crossing). Two main categories are distinguished:
    - Traffic modes that cross roads at predetermined locations:
        - Rail based transportation
        - Waterways crossing roads: movable bridges
        - Pedestrians and bicycles at cross-overs
    - Traffic modes that cross roads at undetermined locations, for instance pedestrians that cross to a sidewalk or destination at the opposite side of the road.
3. A last cluster of traffic services regards the connection of road based traffic with other modes, such as intermediate transfers over waterways (ferries) that are part of the routing alternatives.

To include these aspects, the transportation infrastructure requires:

- Ability to distinguish various vehicle types, and relate this vehicle type to the attributes of roads and lanes, such as speed limit and accessibility;
- Allow an overlap of mixed use of the infrastructure such as driving lanes and a rail track for trams;
- Implement the infrastructure of other modes, such as rail or waterway crossings, when it interferes with road traffic;
- Provide intermediate transfers that are part of a road based trip, such as transfers over waterways.

For situations where lane discipline is lacking, an additional requirement holds:

- a description of the road infrastructure that reflects the factual 2D topology of the area where vehicles and (in case of Shared Space) pedestrians can move.

5.2.2 The road infrastructure: allow for vehicle operations and trips

Traffic flow models aim to provide insight in the performance of the transport and traffic system. The performance is an outcome from the modelled behaviour of drivers/vehicles and their interactions at the road infrastructure, and is expressed by performance indicators such as travel times, intensity and density. Microsimulation models provide the highest level of detail within the transport and traffic models. Not all details of a road infrastructure have to be necessarily relevant to model the traffic performance in a microsimulation model.

Detailed information about the road surface for instance, is less relevant, as for most microsimulation models the physical experience of 'driving' is not part of the model. Thus, objects only need to be included when they have a significant impact on traffic flow behaviour. The relevant aspects that need to be considered are the determination of trips and their costs and the ability to model vehicle operations, in order to allow route and modal choice, and the choices of movement.

Basically, this first requires a design that allows for a *valid topology of the roads, junctions and lanes*, in order to create (multi-modal) routes between activity locations and at a more detailed level a continuous path for driving (i.e. a trajectory for the next 10 seconds). This requires:

- The ability to create routes between the activity points, along roads and lanes, including multi-modal trip legs and transfers between modes (de Dios Ortuzar and Willumsen (2011), page 474);
- Provide occupancy information by giving the locations of nearby objects on the road infrastructure, and allow for geometric computations such as calculating the distance between cars (van Lint, Schakel et al. (2016));
- Topological information of junctions, with internal lanes that connect the entrances and exits of a junction (Wilkie, Sewall et al. (2015), Willemsen, Kearney et al. (2006));

Secondly, the route and path creation requires information from the *relevant objects of the road infrastructure* such as speed limits and accessibility. By analysing the functional requirements of the microsimulation models to perform route choice and path planning while driving, the necessary information of the relevant objects of the road infrastructure is derived. This selection process starts by identifying and clustering objects into homogenous groups. Table 5.1 provides an overview of these clusters with some exemplar representations, and shows their impact on driving behaviour.

The *first* cluster describes the "lay-out" of the road. The *accessible* part of the road contains the driving lanes and additional space for vehicles such as parking spots or bus stops. The microsimulation model requires information about both the geometry (for movement) and physical characteristics such as markings along the lane and speed bumps. The *non-accessible* part of the road contains elements as barriers and traffic bollards that provide further information for driving (warn and/or inform). Not all of these elements will (and need to) be implemented in a microsimulation model. The relevance depends on their impact on driving behaviour. Obviously, some types of objects have a clear impact on lane choice and overtaking. The type of lane *marker* (broken versus solid) for instance, has a direct impact on lane choice and overtaking, and a pedestrian crossing impacts drivers' speed behaviour (Varhelyi (1998)). *Physical* road characteristics such as curvature, speed bumps, gradient and road surface clearly have an impact on speed choice.

Table 5.1 Overview of road infrastructure objects and their impact on traffic behaviour

Cluster	Type	Example	Speed choice	Lane choice	Over-taking	Route choice
I: Road design	Road markers along					
		Lane marker		+	+	
	Road markers across					
		Turning arrow		+		+
		Cross-over	+			
	Physical road characteristics					
		Curvature	+	+	+	
		Road/lane width	+	+	+	
		Road gradient	+		+	+
		Road surface	+			
		Speed bumps	+			
	Road furniture					
		Guard rail/ safety fence	+*	+*		
		Median barrier	+*	+*		
	Lane design					
		Parking spot	+	+		
		Bus/tram stop	+	+		
		Bus/tram lane	+	+		
II: Road info	Static signs					
		Max. speed	+	+	+	
		Priority/give way	+			
		Direction		+		+
	Dynamic signs					
		Variable speed limits	+	+	+	
		Variable message sign	+	+		+
III: Control	Traffic management					
		Traffic lights	+	+		+
		Dynamic route info panels				+
		Speed cameras	+			

** The lateral distance from the driving lanes to these objects slightly impacts behaviour*

For instance, measurements show a speed reduction in the presence of a curb (Cruzado and Donnell (2009)) or speed bump (Pau (2002)) and lower driving speeds on a slippery road surface (Rämä and Kulmala (2000)). Overtaking is impacted by factors such as the horizontal curvature (Farah and Toledo (2010)). For some other characteristics the impact of driving behaviour is less straightforward. An example is the impact of the slope of a road in hilly areas on route choice, as is described in Pingel (2010). Studies towards the capacity of roads (for instance Henkens and Tamminga (2015), Chitturi and Benekohal (2005)) reveal that lane capacity (as a result of driving behaviour) is influenced by lane width, speed-limits, and the absence of emergency lanes. The latter is supported by studies that indicate (1) a slightly higher speed on roads with a right shoulder lane and (2) a tendency of vehicles on roads with a right shoulder lane to drive relatively more to the right, while in case of a guardrail vehicles tend to drive more to the left (Bella (2013)), and (3) an averagely higher speed (Cruzado and Donnell (2009), Lewis-Evans and Charlton (2006)) as well as a higher capacity (Chandra and Kumar (2003)) on wider lanes. Finally, the impact of bus stops and parking lots will have a serious impact on car driving behaviour (speed and lane choice) as has been reported by Bonsall, Liu et al. (2005). The same accounts for the appearance of bus and tram lanes that interfere with private cars (Arasan and Vedagiri (2008)).

The *second* cluster comprises traffic signs on (i.e. turning arrows), above (i.e. dynamic traffic information panels) or along a road (i.e. legal speed limit). The messages on these signs can be subdivided into categories such as mandatory, information and priority signs. The relevance to make this distinction is their difference in response. *Mandatory* signs generally have an immediate and clear impact, for instance imposing a speed limit or prohibiting the entrance of a road. *Priority* signs provide drivers knowledge about priority at junctions, and thus influence behaviour when nearing a junction. With respect to *warning* signs response is rather diverse. A sign warning for animals crossing the road may have less impact on behaviour than signs warning for road works or sharp turns (slow down). As there is no clear and generic behaviour by type of sign, the impact of the main types of signs that do have a clear and relevant impact on driving will be illustrated by exemplar. Turn-arrows for instance, not only have an impact on lane choice, but also on route choice: the absence of turn arrows shows higher rates of drivers selecting a wrong-way turning lane (Schrock, Hawkins Jr et al. (2005)). The impact of a speed limit has an impact om the speed choice, but also on lane changing behaviour (Knoop, Duret et al. (2010)) and on overtaking (Farah and Toledo (2010)). Variable speed limits do impact the same behaviours, but its impact is different. As Knoop, Duret et al. (2010) shows, the speed limits can be used to increase the usage of under-utilized lanes, and therefore increase road capacity. The signs indicating the priority rules of the road (give way or priority) do have a direct impact on speed behaviour for traffic: in case of a give-way cars will decrease speed when nearing junctions as they may encounter traffic from opposing directions.

The *third* cluster covers the *dynamic traffic management and monitoring systems*. Already from 1900, traffic lights have been used to control traffic. The traffic light system controls the actual traffic lights, and uses sensors to detect nearing traffic. These *sensors* can be shaped in various ways. Most common for traffic lights is an inductive loop, but alternatively, detection is possible by means of other sensors, such as radar, bluetooth or camera. While communication systems allow vehicles to connect with road side systems, and can be considered as “moving sensors”, they are not part of the physical infrastructure. Traffic light controllers have a direct impact on speed behaviour, but can also be used to guide traffic through a network by providing more green time for specific directions. In the latter case, dynamic route information panels can be used to support the route choice decisions at junctions (Vreeswijk and Blokpoel (2012)).

Whereas most aspects of driving behaviour are only related to the road infrastructure, overtaking behaviour on roads with opposing traffic, as well as driving behaviour when nearing a junction, requires additional information of objects along the road that have an impact on the *field of vision* of drivers. In most traffic microsimulation models, such as Vissim or Paramics, the field of vision is defined as a variable that defines the distance to a junction, from where the driver can see nearing traffic from all other arms of the junction. To model these behaviours explicitly, a 3D representation of objects along the road, such as buildings, vegetation and safety fences is necessary.

Summarizing, we distinguish the following requirements:

- Represent the geometrical design of a road as characterized by the horizontal and vertical alignment, and the cross section of the road, including the road marking (stripes) to distinguish lanes and the shoulder;
- Road and lane related objects that have an impact on behaviour driving;
- Allowing the inclusion of traffic management and monitoring systems when they have an impact on traffic behaviour.

5.2.3 Public transport, transfers between modes and connectivity with activity locations

Trips by *public transport* consist of multiple trip chains and thus require a sequence of travel decisions. In case of multiple path alternatives, the path choice depends on a consideration of the joint (dis)utility of all trip related decisions for every path. While the movement of the public transport vehicles at the road is comparable to private cars, the main difference regards the fact that passengers choose a specific combination of trip legs that are composed of different modes. With respect to the public transport legs, the activity of boarding and alighting in relation to the vehicle capacity, requires additional modelling. In addition, the changes between modes require the modelling of transfers. The transport assignment models are usually classified into frequency-based and schedule-based models differing in their network supply representation and their implications on the passenger loading procedure. In addition to these two approaches, agent-based simulation models more recently emerged as an alternative approach. In all cases, public transport requires the additional modelling of some specific vehicle characteristics: boarding and alighting capacity, number of seats and passenger capacity. With respect to the road infrastructure, the main additional aspect regards the modelling of transfers by means of walking. There are various means of modelling these pedestrian movements. Most pedestrian models are either microscopic (cellular automata and social force) or macroscopic (continuum: speed, density and flow) (Campanella (2016)). At the highest level of detail, the microscopic approach models pedestrian behaviour in a 2D space. The movements of pedestrians are directed by their route, and the lay-out of the spaces. Local route choice is in turn governed by the pedestrian's tendency to move away from high-density areas and delay, caused by local densities. In the microscopic models this behaviour is a result of the decisions of the individual pedestrians. A macroscopic representation of pedestrian movements in a 2D space is presented by Hoogendoorn, van Wageningen-Kessels et al. (2015). The multi-class pedestrian model is using speed density relations to govern behaviour, and thus relates movement behaviour also to the geometry and topology of the 2D-infrastructure. Other approaches solve crowd movements by means of mathematical approaches developed in graph theory (Duives, Daamen et al. (2013)). An example is provided in Daamen and Hoogendoorn (2003) where the infrastructure elements at a train station (walkways and stairs) are represented by a network of links. The movements of the pedestrians depend on link parameter (density) and attributes of the pedestrians (their free speed, and familiarity with the environment). To allow for these transfers to other modes of traffic, the connections between different infrastructures need to be part of the design of the transport infrastructure. At the highest level of detail, this requires the representation of the transfer between a road based network and a rail based network, for instance by modelling the places where vehicles park and the transfer to the halting locations of the public transport system. The design depends on the modelling approach and can either be designed as a graph that captures the topological relationships of the walking routes between the parking location and the halting locations, or as a 2D space with entrances and exits.

Apart from the roads themselves, trips have to start and end somewhere, sometimes including stops at transfer locations. In the real world, vehicles will be parked at some location. While some of the models enable the use of parking spots and areas, many transport and traffic models often simplify this process by adding Traffic Analysis Zones (TAZ) where (1) traffic is dumped at or taken from the network or (2) feeding links connect the TAZ to the network. For the connection with the activity locations, again the parking spot is the location where travellers start of finish a trip. As this thesis regards the design of road based infrastructure, we only present the design of parking spots, and do not include the infrastructure of the transfers from and to the public transport system.

In some cases, partial road networks that are divided by waterways, are connected through ferries. To implement such connections, the ferry can be modelled as a separate mode that moves vehicles over a river. In this case, the operations of the ferry should be defined in terms of: frequency of crossing, capacity and time to cross the waterway.

Regarding the impact for the infrastructure, we add the following requirements for connecting road based trips with public transport and activity locations.

- the connection between the road and public transport hubs by representing the pedestrian transfers, either by the geometry of the 2D space or a link based network;
- Include parking spots or parking areas for multimodal trips and the relation with activity locations;
- Representation of ferries that connect roads that are divided by waterways.

5.3 Modelling traffic flows at various levels of detail

While the previous section regards the representation of the infrastructure for microsimulation models, this section presents the types of traffic flow modelling at coarser levels of detail. We first describe the various modelling methods, and then present the impact on the representation of model objects an attributes.

5.3.1 From micro to macroscopic approaches

When the trip starts, movement requires operational choices such as the level of speed, lane choice and headway to leading vehicles. The representation of these choice behaviours are modelled at different levels of detail, and are often referred to by terms as microscopic, mesoscopic and macroscopic. This categorization is merely based on two factors (Hoogendoorn (2010), Handford and Rogers (2011), Duives, Daamen et al. (2013)):

- The representation of the traffic units:
  1. Individual traffic units
  2. Groups of traffic units
  3. Aggregated: traffic units as flows
- The representation of behaviour:
  A. Individual agents and vehicles modelled separately
  B. Individual vehicle/driver combination
  C. Averaged Aggregated: multi-class or totalized

The combination of both factors determines at what level choices and movements are modelled. Table 5.2 presents the modelling approaches by their level of detail.

Table 5.2: levels of aggregation in traffic flow modelling

Description	Traffic Units/ Behaviour
Agent based microsimulation	1A
Vehicle/driver combi based microsimulation	1B
Mesoscopic	1C, 2C
Macroscopic traffic assignment	3C

The term mesoscopic refers to any model between the microsimulation and macroscopic models, and thus includes various approaches. Some examples are:

- Mezzo, Dynasmart (1C)
  These mesoscopic packages model vehicles individually, but do not represent lanes individually. A Link is divided in a running and a queuing part, which extends upstream from the end of the link when capacity is exceeded (Burghout (2005));
- Dynameq (1C)
  As the previous models, this package models traffic flows by combining individual traffic units with an event based approach. Vehicle attributes are assumed identical over the entire traffic stream, and each vehicle adopts a link specific free flow speed, and uses the triangular flow-density relationship. This model uses a lane based representation (Mahut and Florian (2010));

- Contram (2C)
  This form of models groups vehicles into packages and routes it through the network based on sped-density relationships (Burghout (2005)).

While the mesoscopic approaches may describe lane-specific behaviour, it does not explicitly use the detailed description as in microscopic traffic simulation.

Microscopic simulation

The microsimulation models require a detailed representation of the transportation infrastructure, as has been described in section 4.2.2. Within the simulation, most of the situations (as shown in Figure 4.3) are modelled through the interaction of the driver with other vehicles, and the characteristics of the infrastructure. Yet, some behaviours are already defined exogenously, such as the visibility (or perception) of the environment that is generally implemented as an attribute of a link or lane, or the impact of objects such as speed-bumps (modelled as a lower speed-limit at a certain point at a link). Figure 5.3 provides a schematic overview of the main characteristics of the microsimulation model, showing the units of traffic, the infrastructure and the model (methods). The traffic flow phenomena thus are largely determined by the driver/vehicle behaviour.

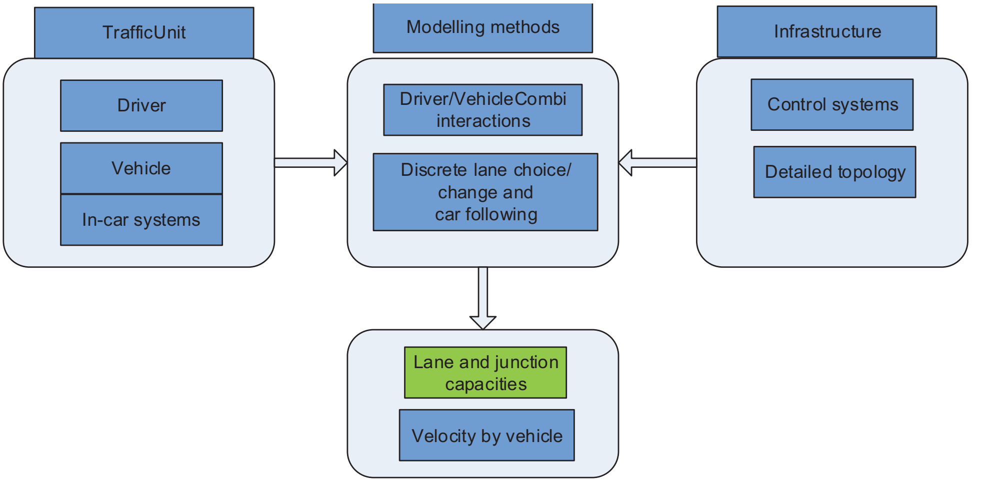


Figure 5.3: Key characteristics of micro simulation model

Meso- and macroscopic approaches

Advantages of the meso- and macroscopic approaches are twofold. Firstly, the applications require less computational effort than microsimulation models and thus are faster, which allows the modelling of large networks. Secondly, they require few parameters and hence can be calibrated more easily. A disadvantage of these meso- and macroscopic models is their difficulty to represent detailed traffic flow behaviour at junctions that are controlled by traffic lights.

The mesoscopic models either averages behaviours or aggregates traffic units, whereas macroscopic models do both. In all cases, the description of the transportation infrastructure is less detailed with regards to the geometry of lanes and junctions: these aggregated approaches typically apply link and node based models (Tampère, Corthout et al. (2011)). The choices of mode, route and movement are not made by the individual traveller, but are based on aggregate functions. Individual agents and traffic units are not registered anymore, but implemented in flow based descriptions at the link and node level. The impact of the situations A to L (Figure 4.3) on speed and/or capacity now do not depend on the behaviour of the individual driver anymore, but are *abstracted* and "transferred" to the attributes of the infrastructure (links and nodes). The resulting impact on speed now depends on these link/node attributes and the size of the traffic flows.

Figure 5.4 shows the fundamental diagram that describes the flow as a function of the capacity of a link and the density of vehicles on a link. Instead of modelling and estimating car driving behaviour (in micro-simulation models), this requires the estimation of link based attributes and parameters.

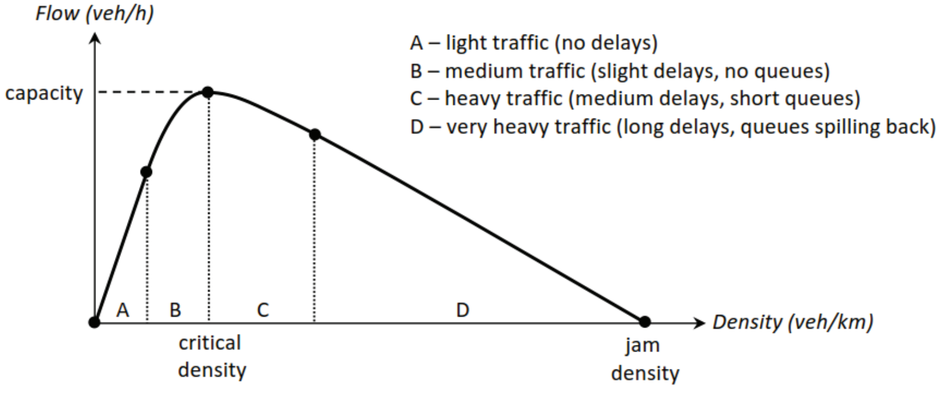


Figure 5.4: fundamental diagram (source: Bliemer, Raadsen et al. (2015)).

The capacity and critical density are variables whose value are a result from driver and vehicle behaviour that on their turn depend on factors such as weather, information technology and so forth (Hoogendoorn and Knoop (2012)). As Figure 5.5 shows, the lane capacity that is one of the results of a microsimulation model, now is an attribute that is part of the description of the infrastructure: an input instead of a result.

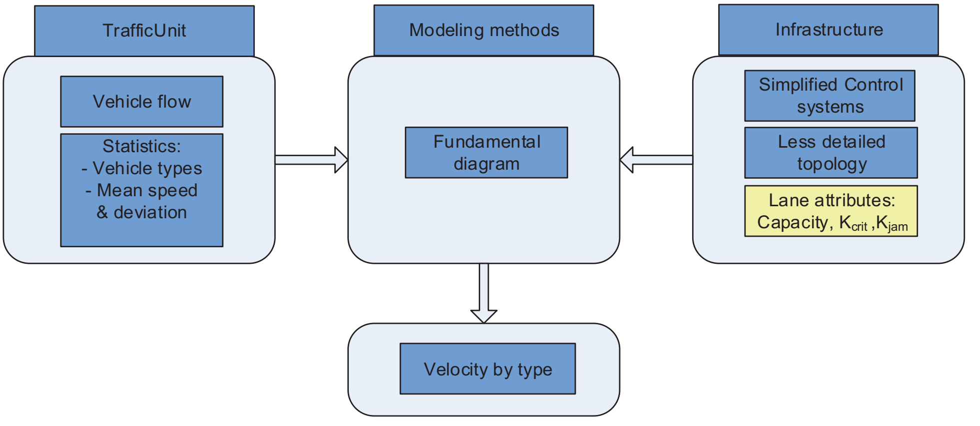


Figure 5.5: Key characteristics of macroscopic model types

Extended approaches however, do provide methods to determine the effective capacity drop. Leclercq, Marczak et al. (2016) for instance, provide analytical expressions to estimate the capacity drops for discretionary lane changes and merges at multilane freeways. This requires the addition of only few parameters such as the wave speed, the truck fraction and the length of the on-ramp. Likewise, Marczak and Buisson (2015) introduces a macroscopic approach to determine the capacity drop at weaving areas. These extensions introduce analytical methods to improve the representation of traffic flow phenomena. As Figure 5.6 shows, the drop in capacity now becomes a result instead of a fixed attribute of the infrastructure.

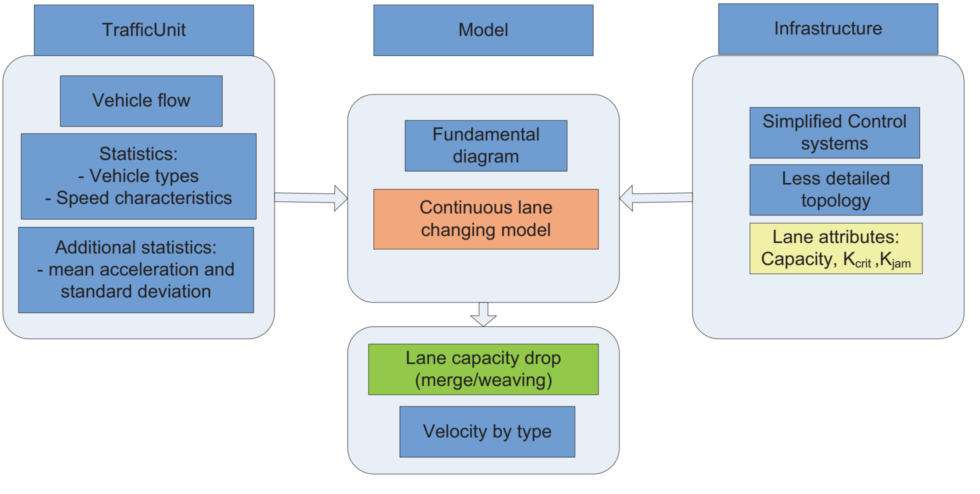


Figure 5.6: Key characteristics of extended macroscopic model types

Network Fundamental Diagram

While the mesoscopic and macroscopic approaches require a *network* of links and nodes as the basic entity for traffic assignment, the *Network Fundamental Diagram* approaches traffic at a lower level of detail by modelling *Zones* that represent a neighbourhood with a collection of roads. For every zone a network fundamental diagram is defined that is tuned to the zone-characteristics. Figure 5.7 shows examples of NFD´s for zones with different characteristics. The maximum flow from areas that have roads with averagely lower speed limits (30 versus 50 km/h) is lower. Moreover, there can be other reasons for differences in outflows, for instance in city centres with a lot of slow moving traffic. Additionally, Figure 5.8 shows that the outflows not only depend on density, but also on the distribution of traffic over the area.

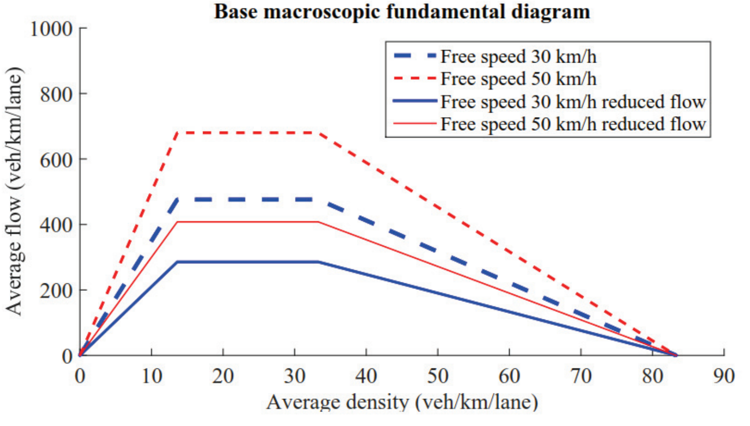


Figure 5.7: base network fundamental diagram (source: Knoop, Tamminga et al. (2016))

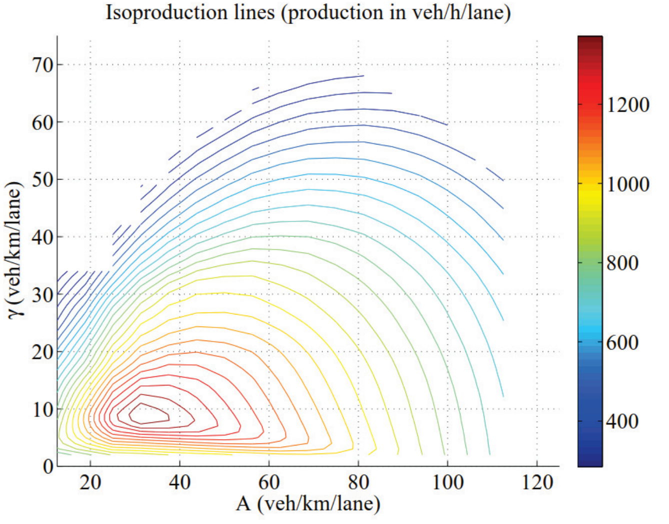


Figure 5.8: Iso-production lines from the generalized macroscopic fundamental diagram (source: Knoop, Hoogendoorn et al. (2013))

Just like the meso- and mesoscopic model approaches, the objects that represent the infrastructure, contain attributes that can be directly related to infrastructure objects, but also behavioural components. In chapter 8.3, a further elaboration of the NTM-approach is provided by a case study for the city of The Hague.

5.3.2 The representation of the transportation infrastructure at various levels of detail

At a detailed level, the cross section of a *link* essentially describes the road as a collection of lanes, with their longitudinal and lateral connectivity features, and its topographical characteristics in terms of horizontal curvature and vertical slope. In addition, a *node model* represents the connectivity of lanes at junctions and merges. The modelling of junctions in the microsimulation models (situations G-I in Figure 4.3) requires a representation of turning lanes with the location of stop lines, and the connection with the exit lanes from the junction. In case of traffic lights, additional information about the control algorithm, in combination with the specific objects such as vehicle sensors, becomes relevant.

As has been sketched in Figure 5.2 the representation of a road in macroscopic traffic flow models is generalized by discarding topographical details. There is less need for a detailed representation of the lane and junction layout and other characteristics, such as the curvature of a road. Instead of describing the topography of lanes explicitly, the lane characteristics now become an attribute of a *link* without information about their longitudinal and lateral connectivity. Despite these differences in detailedness, these models all require a valid topology that accounts for the relations between road network elements, in terms of connectivity between separate road stretches, and entrances and exits at junctions. In essence, this requires a combination of links and nodes that describes the connectivity of lanes and roads within a transportation network, or a set of planes if the infrastructure is described by 2D objects (for instance at pedestrian models). The description of *junctions* still requires information about the number of turning lanes and the connection with exiting links (see for example Tampère, Corthout et al. (2011)), but does not necessarily require the exact connection between an incoming and exiting lane. Moreover, the location of sensors and stop lines is not relevant anymore.

We can conclude that a microscopic model of the transportation infrastructure contains all topographical and topological information to transform it into a macroscopic description: the network characteristics (topology) of roads and junctions, with details of the approaching lanes and the turning movements, can directly be derived from the more detailed description in microsimulation models. But attributes that contain information that relies on behaviour, such as capacity, cannot be directly derived, and require additional actions.

With respect to pedestrians, the dominance of 2D movements requires a transport infrastructure that describes the physical walking space and objects such as doors and stairs and elevators that are relevant for movement. As there are no specific markings to guide walking and manage pedestrian movements, less details are required to model the objects of the 2D space. The representation at lower levels of details therefore requires less transformations.

5.3.3 Attributes of objects from the transportation infrastructure

In addition to the topological and topographical features, attributes are required to further describe the transportation infrastructure. Typical examples are the legal speed limit, vehicle types that are allowed to drive at a road, or priority rules of links entering a junction.

While the generalization of the topography and topology leads to a reduction of objects, it also requires an addition and/or adjustment of attributes for the macroscopic models. The absence of an explicit lane object forces the introduction of a link attribute that describes the number of lanes. The description of the connectors at a junction can be implemented as a table attribute of a node that describes the connections between ingoing an exit links.

For other link-attributes, the values can be derived from the higher level of detail, in case the attribute does not have a behavioural component. Examples of such attributes are the legal speed limit, the road type and the allowed vehicle-types.

An example where a conversion from microsimulation to a macroscopic traffic flow model is less straightforward, is the impact of speed bumps. Within a microsimulation a speedbump can be modelled as an object at a link with a lower maximum speed. The resulting link speed in the microsimulation model is the result of the average speed of all vehicles traversing this link (lowering their speed at the speed bump). In a macroscopic traffic flow model this impact can be represented by an equation that computes the additional travel time as a function of the difference in driving speed on the undisturbed part of the link, versus the lower speed near the speed bumps. The impact of curvature, slopes, and speed bumps can thus be transformed into a generic *speed level attribute* at a link level.

A key variable that is required in the meso- and macroscopic traffic flow models, but absent in microsimulation, is the ***capacity*** of links. Within microsimulation models, the capacity of lanes and links is a result of the model, and thus cannot be derived from the microsimulation transportation infrastructure. Therefore, other approaches are required to model the capacity in macroscopic traffic flow models. A possible approach is to derive capacity by means of indicators that are provided by handbooks such as the Highway Capacity Manual. These figures are based on measurements for a great amount of road configurations. Attributes that may influence the capacity are the configuration (number of lanes, weaving areas), lane width, speed-limit, absence of emergency lanes, road-works, tunnels and bridges. As many of these attributes are available at the highest level of detail, the derivation of the lane capacity from road characteristics appears a suitable approach. Such an approach should be dealt with carefully, as there appears a high level of variation between measurements of comparable configurations. Moreover, in many cases, this variation is also due to the fact that there are multiple factors that impact road capacity. For road-works there will often be a combination of factors such as narrow lanes, the design of the road works at the start and end (curvature), unfamiliarity with the situation, and distraction from the road works.

In addition to the use of measurements, there appear two other approaches to derive capacity: the simulation based and the analytical approach. An example of the simulation based approach is reported in Liu, May et al. (2011). The article describes the derivation of both performance and supply curves for urban networks. Specifically, the impact of differences in demand patterns on them is estimated using microsimulation. In this case, the use of measurements is insufficient to obtain all the data needed. Figure 5.9 shows an example of a curve that is derived from microsimulation.

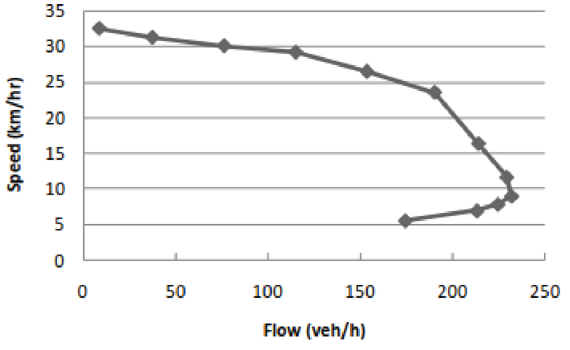


Figure 5.9: Speed-flow relationship based on microsimulation (source: Liu, May et al. (2011))

In addition to the simulation based derivation of capacity, the analytical derivation transforms the impact of discrete events into continuous formulae that use flow as an input. An example is provided in Leclercq, Marczak et al. (2016) that presents an analytical formulation of the effective capacity of a multilane merge. Figure 5.10 illustrates the lane changing manoeuvres at the merge. With only few parameters, the lane capacities are estimated from the flows at the main road and the merge. Regarding the description of the infrastructure, this requires the definition of the lane changing area (L_{DLC}) and the length of the on-ramp (L).

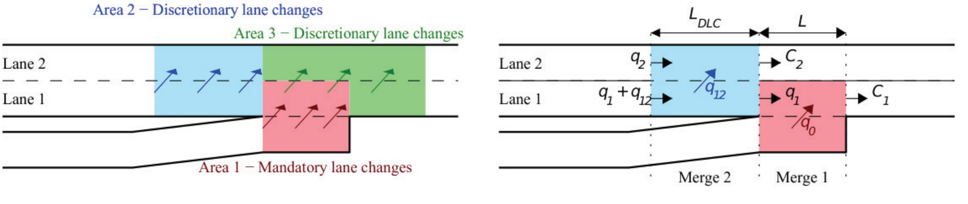


Figure 5.10: transforming discrete lane changes into continuous descriptions (source: Leclercq, Marczak et al. (2016))

Similar approaches are required to determine the junction capacity. Again, behaviour of traffic ultimately determines the capacity, and thus it cannot be directly derived from the detailed network characteristics alone. As reported by Tampère, Corthout et al. (2011)) this still requires information about the number of turning lanes and the connection with exiting links. Other aspects of the node model to determine the capacity are:

- Un-signalized
    - Conflict points at crossing flows or merges
    - Turning angle
    - Visibility
- Signalized
    - Limiting supply by signal controller
    - Red and green phases

All of these items are usually part of a microsimulation model.

5.3.4 The impact of various levels of detail on the requirements for the design of objects of the infrastructure

The main conclusion from the previous sub-sections is that the meso- and macroscopic models require less detailed information of the cross section of a link than the microscopic simulation models. Instead of a detailed description of the lane topology, a link description of a road with attributes that describe the lane-related information suffices. The same reasoning accounts for a junction. It requires a less detailed description, but still requires information about the number of turning lanes and the connection with exiting links.

In order to efficiently model networks at various levels of detail, we derive to the following requirements:

- There should be a correspondence between the links at various levels of detail, in order to exchange information between the levels of detail.
- There should be a relation between the junction-description at various levels of detail.

5.4 Demand modelling at various levels of detail

While the previous section described the infrastructure, this section discusses demand modelling. The agent and activity based modelling approaches provide the highest level of detail with regards to demand modelling. Demand is represented by a trip-pattern object that contains the sequence of trips between activities performed by a specific person. Every separate trip requires a departure time, an origin (activity location) and a destination. A trip is further decomposed in a chain of trip-legs. Again, every trip-leg has an origin and destination, and is specified by its mode of transport. At such a high level of detail, the location of the activities can be represented by their actual address and entrance(s). For large scale applications however, this information is generally aggregated by applying traffic zones. The challenge for a flexible demand modelling framework, is to combine this variety of levels of detail. Balmer, Axhausen et al. (2006) provides an example to attain data at the individual level by disaggregation procedures, and shows that many of these procedures follow a conversion from larger size areas towards more detailed levels. The other way around, the aggregation from individual addresses towards lower levels of spatial resolution uses the same kind of geographical entities. These can be raster representation, administrative areas, or other areal classifications. By applying *hierarchical zoning systems*, a switch from a high level of detail towards coarser description by means of geographical areas, as a basis for aggregated model approaches, appears suitable. The aggregation of the Person objects within a geographical area, if required classified by attributes as age, gender and others, results in an aggregated description of the population. In the same way activities can be aggregated, and classified by sub-classes to reflect the zonal characteristics that can be used to compute trips.

With respect to the modelling of trips, many transport and traffic models at a coarser level of detail, apply so-called *origin destination matrices* as input for static or dynamic traffic assignment models (see: Balmer, Rieser et al. (2009)). By means of aggregation procedures, the trip-pattern can be modified into such mode specific origin-destination matrices. In case of trips with more than one mode, the aggregation should start with a decomposition of every trip into trip-legs per mode. As only the departure-time of the first trip-leg is provided, this requires an assessment of the departure times of all subsequent trip-legs. This can be done by assessing the travel times by mode for all separate trips-legs. Subsequently, these trip-legs can be grouped by time interval into origin-destination matrices by mode and time of day.

We propose to apply the following objects, to allow for these aggregation procedures:

- The Traffic Analysis Zone (TAZ) object is the entity where trips depart and arrive. The zone is defined as a polygon that contains a set of activity locations. At the highest level of detail, a TAZ can be represented by an individual facility/activity location and its traffic access point with information about 33:
    - the parking location(s) (in case of trips by car);
    - the entrance(s) of a house or facility (in case of a walk trip);
    - the characteristics of the activity location.

  At coarser levels of detail, the TAZ provides aggregated information to describe the characteristics of the activity locations (such as number of inhabitants, working places by type, number of parking spots).
- The Trip-pattern object is the basic entity to describe traffic demand, as it reflects the dependencies between trips and activities. Aggregation procedures allow representations at coarser levels of detail, for instance a towards an origin-destination-matrix.

With respect to the design of the road infrastructure, there are no additional requirements that result from this demand analysis.

5.5 Extending to higher levels of detail

While the definition of *highest level of detail* is derived from the requirements of microsimulation models, other applications, such as driving simulators or traffic safety analysis, require even more details. A typical example is a driving simulator where the behaviours of surrounding vehicles are modelled with a microsimulation model (Olstam (2009, Jiang, Miska et al. (2010)). For driving simulators and the simulation of tires or vibration, physical aspects such as the road, surface properties are necessary to reproduce a real driving experience as closely as possible (source: Dupuis (2015)).

A first notion is that, in addition to aggregation, a second principle for changing the level of detail is *extensibility*. This principle that originates from the object oriented software design, is a systems design principle, where the implementation takes future growth into consideration. This allows an extension of objects, with new attributes and functions. While a road lane may have the surface described by the type of material, it could be extended by providing further details. An example of such an approach is the *OpenCRG* project for the object modelling of road surface descriptions (Czerwionka and Wang (2011)). By means of a curved regular grid (CRG) the road elevation is represented at a detailed level (Dupuis (2015)). This description can be linked to the road description as an ingredient for a driving simulator environment.

5.6 List of requirements and objectives for the design of the modelled transportation infrastructure

From the previous sections, we list the following requirements for the design of the modelled transportation infrastructure:

- **R-InfraDesign.1**
  Ability to distinguish various vehicle types, and relate these vehicle type to the attributes of roads and lanes, such as speed limit and accessibility.
- **R-InfraDesign.2**
  Allow an overlap of mixed use of the infrastructure such as driving lanes and a rail track for trams.
- **R-InfraDesign.3**
  Allow the implementation the infrastructure of other modes, such as rail or waterway crossings, when it interferes with road traffic.
- **R-InfraDesign.4**
  Provide objects to model intermediate transfers that are part of a road based trip, such as ferries that connect roads divided by waterways.
- **R-InfraDesign.5**
  Represent the geometrical design of a road as characterized by the horizontal curves and vertical slopes, and the cross section of the road, including the road marking (stripes) to distinguish lanes and the shoulder.
- **R-InfraDesign.6**
  Define road and lane related objects that have an impact on driving behaviour.
- **R-InfraDesign.7**
  Allow for the inclusion of traffic management and monitoring systems insofar they have an impact on traffic behaviour.
- **R-InfraDesign.8**
  Include parking spots and parking areas as intermediate destinations for multimodal trips (such as P&R) and final destinations at activity locations.

For the various levels of detail:

- **R-InfraDesign.9**
  There is a correspondence between the links and junctions at various levels of detail, in order to exchange information between the levels of detail.

In addition to the necessary road related objects, the design of the infrastructure requires the ability to create valid topology and derive the trip costs. This requires:

- **R-Topology.1**
  The ability to create routes between activity points, along roads and lanes, including multi-modal trip legs and transfers between modes.
- **R-Topology.2**
  Provide occupancy information by giving the locations of nearby objects on the road infrastructure, and allow for geometric computations such as calculating the distance between cars.
- **R-Topology.3**
  The design provides topological information of junctions, with internal lanes that connect the entrances and exits of a junction.

5.7 Conclusion

The objective of this chapter is to determine how objects of the transport infrastructure for transport and traffic modelling approaches are represented at various levels of detail, and how they relate to each other. We start from the highest level of detail to determine which objects need to be included in the transport and traffic model. Firstly, we analyse the modelling of the interactions of vehicles with the infrastructure and show its impact on the representation of the infrastructure. Secondly, the vehicle operations and interactions require an infrastructure that provides information about lanes and their connectivity, both in lateral and longitudinal direction, in order to create a valid representation of the allowed movements at roads and junctions. In addition, it requires the relevant objects for the operation of traffic management systems, such as traffic lights and detectors. Thirdly, these models require a transport infrastructure that enables the creation of routes and their costs to determine route alternatives.

Three clusters of objects are distinguished to provide this information: (1) the road design that describes the physical road infrastructure and includes the public transport facilities (2) the traffic signs that provide functional information to the traffic, and (3) traffic systems and controllers that are used to regulate traffic. To enable trips over this infrastructure, the activity locations can be represented by their addresses and entrance(s) as a part of the road network.

For the representation of the transportation infrastructure, coarser levels of detail require less details to describe the network topography. The lane representation is not required in macroscopic approaches, and instead, the number of lanes becomes an attribute of a link. A combined representation of a link (macroscopic) and its lanes (microsimulation) appears a flexible design to exchange data between various levels of detail. With respect to the attributes of the objects we distinguish two mechanisms to switch from a high level of detail to lower levels of detail. The *first* group of attributes simply suffices a straightforward exchange for attributes that have the same impact in microsimulation and macroscopic traffic flow models. Examples are the link speed limit or the allowed vehicle types on a link. The *second* group consists of attributes that cannot be transferred without transformations. A key variable within this group is the *capacity* of a lane or road. Within a microsimulation model, capacity is a model result, and depends on vehicle interactions[9] in combination with the road lay-out. For *capacity* a straightforward exchange is therefore not possible, and requires additional data from capacity measurements, in combination with simulations based and/or analytically derived methods.

The traffic assignment zone is the object to represent the activity location, and can be defined by a polygon that captures the entrance(s) of a zone. In case of coarser transport and traffic models with lower levels of detail, a hierarchical zoning system with larger polygons (that do not dissect the smaller zones) enables a flexible exchange of zonal data.

Finally, a further refinement of the level of detail, for instance to create an environment for driving simulators, appears possible by means of the principle of *extensibility*. This allows an extension of objects with new attributes and functions, for instance to refine the surface of a road by a curved regular grid.

[9] Which in their turn depend on the underlying equations and parameters for modelling vehicle behaviour

Chapter 6

Aligning transport and traffic model objects with data standards from GIS

This chapter focusses on the exchange of transport and traffic model objects with external standardized data sources, aiming to align these model objects with the open data standards for geo-spatial information. This requires that every transport and traffic model object has a counterpart in the external database. A second requirement is the opportunity to define multiple levels of detail for the GIS objects that again align to the modelling counterpart.

6.1 Introduction

Models require data input to actually shape the model objects. Oppositely, models produce information that can be used as input for external databases again. This mutual exchange of data between models and geo-spatial databases requires a proper mapping of data objects at various levels of detail. This chapter provides an analysis of data standards and definitions, to determine how such an exchange can be achieved, and what actions are required. By means of mapping of concepts, equivalent classes of objects in the transport and traffic model will be searched for in the data sources, such that objects within the transport and traffic model can be automatically derived from the data sources. Also, when model objects are changed, the updated information can be returned to the data sources.

Section 6.2 introduces the exchange of infrastructural objects between external databases and transport and traffic models. Section 6.3 explores the various data standards and proposes the open data format *CityGML* as a feasible standard to describe the transportation infrastructure and its embedment in the urban and rural space. Section 6.4 provides a more detailed investigation into the suitability of CityGML to specifically describe the required details for the transport infrastructure. As will appear from section 6.4, not all necessary objects are fully defined. We therefore explore CityGML and several other open data standards to distil how they (1) describe the transportation infrastructure and if this matches the requirements for transport and traffic models, and (2) handle various levels of detail (section 6.5). This results in a proposal to adjust CityGML for transportation based on a bi-level representation of objects that provides both space and line descriptions (section 6.6).

6.2 Aligning transport and traffic model objects with data standards from GIS

Traffic and transport models require input data for the representation of the real word. Many of these objects can be denominated as geographically fixed entities. This regards the transport and traffic infrastructure, such as roads with signs and traffic lights, rail, stations, airports and harbours, and spaces for pedestrian movements. In addition, information is required for other fixed objects such as the activity locations (houses, offices, shops). These kind of data are supplied by databases from geographical information systems (GIS). Oppositely, models produce information that can be used as input for GIS databases, for instance to predict future traffic flows that are used as input for dynamic traffic management systems. This mutual exchange of data between transport and traffic models and GIS databases requires a proper mapping of data objects (see Figure 6.1) as has been concluded in section 2.7:

- **R-GIS.1**
  Data objects in transport and traffic models should align to their counterpart in GIS databases.

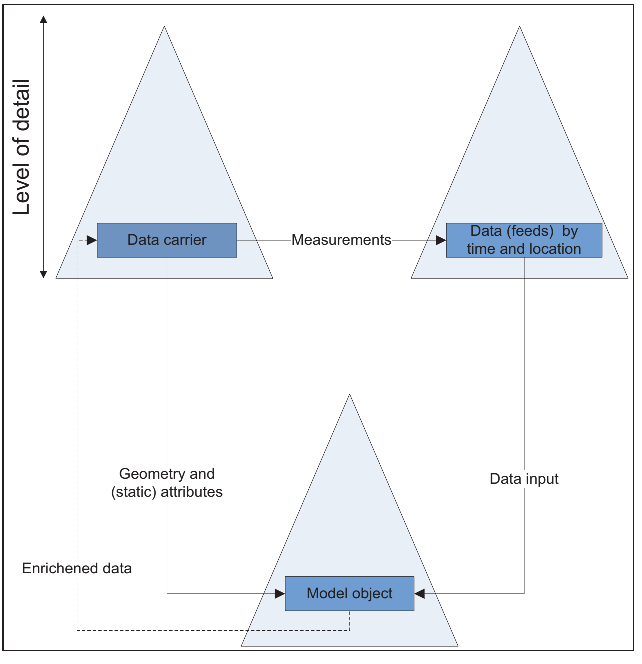


Figure 6.1: Exchange of infrastructural objects between external databases and transport and traffic models

In order to understand the dynamics and dependencies of networks of different modalities, and their connection with the activity locations, and to allow the propagation and visualization of traffic (phenomena), a first identified requirement on multi modal networks is that:

- **R-GIS.2**
  A geometrical, topological, and functional embedding of these infrastructures into the urban space must be done.

This will also allow the joint visualization of virtual 3D city data-models and transport and traffic networks, which would be for instance be very helpful for (1) driving simulations, or (2) to understand the locations and spatial relations of infrastructures in the context of city objects (analogous to utility networks: see Becker, Nagel et al. (2011), Becker, Nagel et al. (2013)). In addition to the geometrical, topological, and functional embedding of multi modal infrastructures into the urban space, we already identified some requirements that also hold for the GIS counterpart (section 5.3.4):

- **R-GIS.3**
  The external database should provide the data for the modelled infrastructure at various levels of detail, in order to allow the exchange information at these levels of detail.
- **R-GIS.4**
  There should be a correspondence between the links at various levels of detail in order to exchange information between the levels of detail.
- **R-GIS.5**
  There should be a relation between the junction-description at various levels of detail.

In addition to these functional requirements, section 2.6 already identified the following requirement:

- **R-GIS.6**
  The exchange of data between models and external databases requires the availability of a mature and complete geospatial standard for transportation infrastructure

The geospatial standards should be internationally accepted, in order to facilitate developing, sharing and using GIS data. The common use of accepted data standards support is essential to harmonize technical specifications for developers, business partners and users.

6.3 Transport modelling standards

In this section we explore the most relevant data standards that cover transport networks, and explore their suitability for a detailed representation of the transport infrastructure that meets the requirements from the previous section.

6.3.1 CityGML

As is described at the website of the OGC *CityGML* is "an open data model and XML-based format for the storage and exchange of virtual 3D city models. It is an application schema for the Geography Markup Language version 3.1.1 (GML3), the extendible international standard for spatial data exchange issued by the Open Geospatial Consortium (OGC) and the ISO TC211. The aim of the development of CityGML is to reach a common definition of the basic entities, attributes, and relations of a 3D city model. This is especially important with respect to the cost-effective sustainable maintenance of 3D city models, allowing the reuse of the same data in different application fields."

CityGML combines the virtual (i.e. geometrical) description of 3D objects with semantic and topological aspects, enabling the possibilities of queries, analysis and data mining. CityGML defines classes and relations for most occurring topographic objects in cities and rural areas, i.e. built structures, but also elevation, vegetation, water bodies, city furniture, and more. Included are generalization hierarchies between thematic classes, aggregations, relations between objects, and spatial properties. "CityGML ... can represent the terrain and 3D objects in different levels of detail simultaneously. Since either simple, single scale models without topology and few semantics or very complex multiscale models with full topology and fine-grained semantical differentiations can be represented, CityGML enables lossless information exchange between different GI systems and users."

CityGML distinguishes five consecutive levels of detail:

- LoD0 -- regional, landscape
- LoD1 -- city, region
- LoD2 -- city districts, projects
- LoD3 -- architectural models (outside), landmarks
- LoD4 -- architectural models (interior)

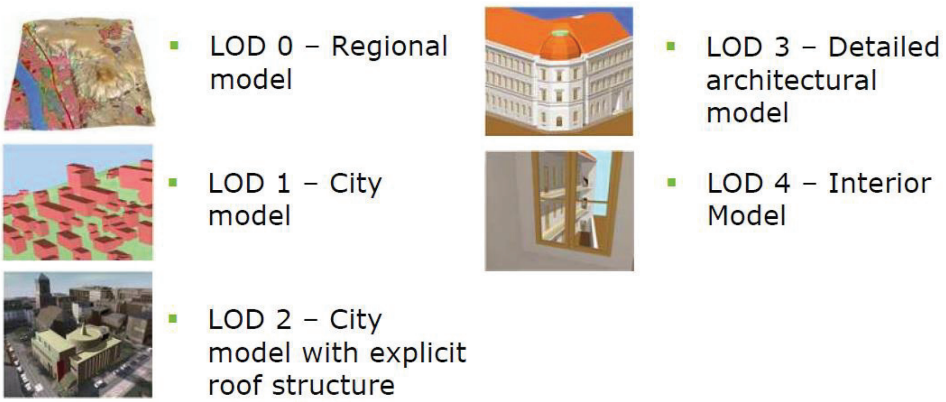


Figure 6.2: levels of detail in CityGML

This distinction into levels of detail and the topological consistency is appealing, as it provides the opportunity to apply a similar approach in the data design of transport and traffic models. CityGML covers both the transportation theme as well as the surrounding environment which allows the supply of all relevant data objects. Another useful feature is the possibility to extend the definition of objects with country specific settings and other geo-data.

6.3.2 *GDF: a basis for navigation*

The *GDF* standard is typically developed for routing, and specifies the data model and exchange format for geographic databases for Intelligent Transportation Systems applications (Cagdas (2007)). It uses a collection of objects for linear networks that resembles the LoD0 in *CityGML* (van Essen and Hiestermann (2005)), but offers a far further elaboration. The conceptual data model distinguishes three distinct layers. The lowest *Level-0* (topology) defines the basic entities nodes, edges or polylines and polygons or faces, and provides the geometric foundation of a road network. These entities are building blocks for the features at *Level-1* that represent real world objects such as a road element that is made of one or more edges and a start and end node (named junction). At *Level-2* complex features such as intersections or roundabouts are defined as an aggregation of *Level-1* features. *Level-2* provides a more generalized representation of the network and represent a functional unit between two intersections. and can be used for way-finding ("turn right at junction East"). GDF is mainly used by commercial mapping companies for navigation and is not designed to be integrated with larger city models.

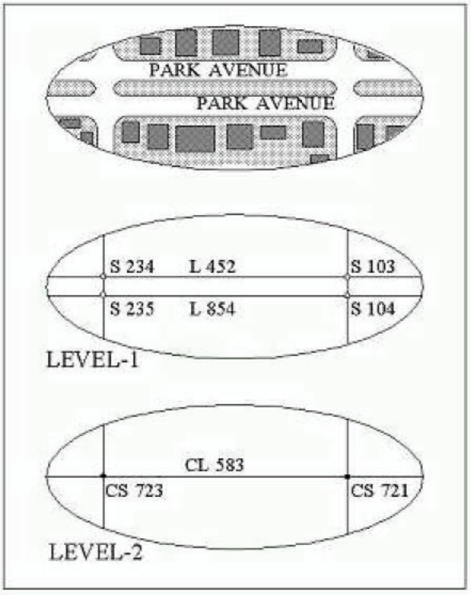


Figure 6.3: Example of a two tired level architecture. Source: Lorenz, Ohlbach et al. (2005)

At *level-1*, the lane information is provided by tables that identify the lanes of a link, its connectivity with succeeding lanes, and the lateral connections. Road related traffic signs are defined as separate objects, and related to a certain lane, direction and location on that element by tables. The *level-1* description does not completely resemble the detailedness that is required for transport and traffic models at a high level of detail. Specifically, information about (1) the width and offset of lanes from the heartline of a road, and (2) details about the junction area are not fully included.

6.3.3 OpenDRIVE

OpenDRIVE is an open xml-based file format for road networks, and has been developed for driving simulators, in order to interface the visual databases from the driving simulator with the vehicle dynamics of the autonomous traffic. Therefore, an additional description of a road network's logics had to be delivered with each visual database. Because OpenDrive is specifically designed for traffic simulations where the required information is clearly defined, there is no concept of Levels of Details. While the *OpenDRIVE* standard provides far less details about road features and attributes than *GDF*, the interesting part is the higher level of detail with respect to the width, curvature and elevation of the road and its lanes. In *OpenDRIVE*, a road is defined by a reference line. Every road contains lane sections to describe the lane geometry. It uses straight line, spiral (linear change of curvature), curve (constant non-zero curvature along run-length), cubic polynom and parametric cubic curves to model all details of the lane configurations in terms of curvature. The description of the cross section of a road into driving lanes is defined by *road lanes records* and *road lane section records,* it merely uses the lateral offset from the reference line to describe a lane´s width and offset. The *road mark record* describes the lane changing rules. In addition, the lateral profile of a cross section as shown in Figure 6.4, can be included.

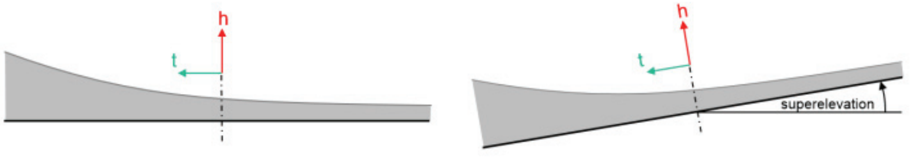


Figure 6.4: lateral profile of a road (source: Dupuis (2015))

Parking spots and comparable elements are modelled by means of *RoadObject* records. The geometry of these objects is less detailed than in *CityGML*, as the objects can either represented as circles or rectangles. Figure 6.5 shows an example for parking spaces with different offset angles. By means of a Repeat Record multiple instances of an object with a regular pattern can be defined to avoid multiple definitions of the same object. By means of an *Object Outline* record also polygonal areas are allowed.

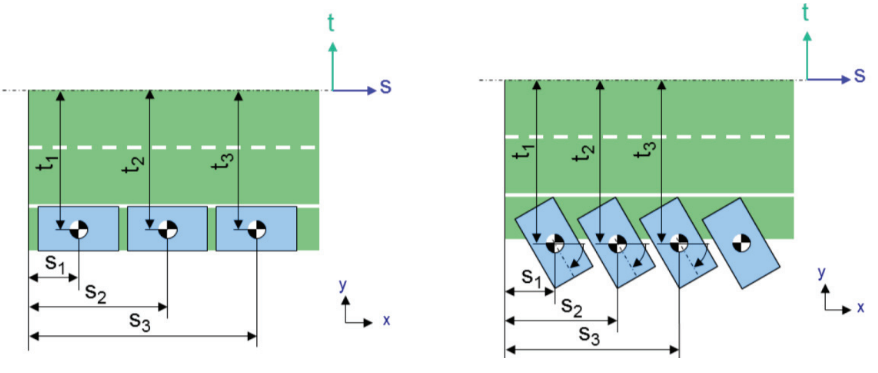


Figure 6.5: Modelling of parking lots in OpenDrive (source: Consortium (2012, Dupuis (2015))

6.3.4 OpenStreetMap

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. OSM provides free access to map images and underlying map data wiki.openstreetmap.org (2017). The OSM network is characterized by so called *way* elements that provide information of a road by its direction. Every way may contain information about lane configuration and turning movements, but information about these attributes is optional. In OSM elements are coded by means of tags in an open tagging system. Guidance is supported by providing best practice and typical coding schemes (Hochmair, Zielstra et al. (2015)). Each tag consists of a key and a value, for instance “maxspeed=>50” and “lanes=>3”. As OSM is a free, community driven map (Goebel, Skuballa et al. (2016)), its quality and completeness depend on voluntary contributions. Comparisons between OSM map data and real world observations, show that not all road details are fully provided. While OSM provides the number of lanes as an attribute of the road element, it lacks details about the lane topology. A further issue regards the description of junctions. OSM provides information about the approaching roads by the "turn:lanes" tag that defines the turning lanes. For instance, a three lane approach with two straight and one combined movement is defined as "turn:lanes => through|through|through;right". A quick comparison of the OSM map with for instance the map-images from Google show that the turn information is not provided for every junction approach. While this incomplete provision of lane and turn attributes can be overcome in due time, a more structural problem regards the OSM object design that lacks a complete description of the road infrastructure as input for a fully-fledged microsimulation model. It has insufficient level of detail.

6.3.5 Conclusion: CityGML provides a good basis

As reported in Labetski, van Gerwen et al. (2018), a review of these standards leads to the following conclusions:

- There is no well-established concept of level of details for roads;
- There is an overwhelming focus on network representation and;
- Integration of roads within a wider city model is not always a consideration.

Furthermore, the standards are often tailored towards a specific application only, e.g. driving simulation or navigation.

While none of the standards addresses all requirements, CityGML addresses part of the shortcomings of the other standards. First of all, it defines classes and relations for most occurring topographic objects in cities and rural area, and thus allows for the embedding of transportation in the urban and rural environment. A further appealing feature is the definition of objects at various levels of detail. At the highest level of detail, it also provides information for indoor, though not all detailed semantic objects that are relevant for pedestrian navigation are defined. If these objects are added, CityGML could form a good basis for pedestrian models (Schaap, Zlatanova et al. (2010)), and, as Gunduz, Isikdag et al. (2016) reports, indoor Application Domain Extensions (ADE) already have been implemented. This opportunity to extend the data standard, also allows further refinements for other domains.

6.4 Comparison of data objects from transport and traffic models with CityGML

CityGML provides an open data model for the storage and exchange of virtual 3D city data-models. The distinction into levels of detail and the topological consistency is appealing, as it provides the opportunity to apply a similar approach in the data design of transport and traffic models. This section investigates the suitability of CityGML as a database for transport and traffic models.

6.4.1 Thematic modules at different levels of detail in CityGML

The CityGML data model[10] is decomposed in a core module and different so-called thematic modules. The core module comprises the basic concepts and components of the CityGML data model and, thus, must be implemented by any conformant system. For transportation and traffic, the relevant thematic modules are: *Transportation*, *Tunnel* and *Bridge*. In addition, some of the transport and traffic modelling relevant information, e.g. for pedestrian or evacuation modelling or detailed traffic simulations, is available in other modules like *Buildings*, *Cityfurniture* and *Landuse*.

The *Transportation* class in CityGML has representations in all five CityGML Levels Of Detail, as is shown in Figure 6.6. The lowest Level Of Detail (LoD0) uses line objects to represent the centreline of a transportation complex. This basic representation aligns to the structure of networks that are used for mesoscopic and macroscopic assignment models, but does not provide details about lane and junction configurations. In LoD1 the surface geometry is provided, and distinguishes the transportation complex from other surface objects like terrain and water. It surrounds all the objects at the road that can be relevant for traffic, but is in itself not directly suitable for modelling traffic operations. For microsimulation models additions that further specify the lay-out of the transportation complex are essential. Within CityGML a further sub division within LoD2-4 for transportation is not apparent, but provides only one level. This LoD2-LoD4 adds the so-called *TrafficArea* and *AuxiliaryTrafficArea* classes. The *TrafficArea* is the part of the road infrastructure which is used for traffic and includes elements such as the driving lanes and parking lots. The *AuxiliaryTrafficArea* describes elements like barriers, green spaces and berms.

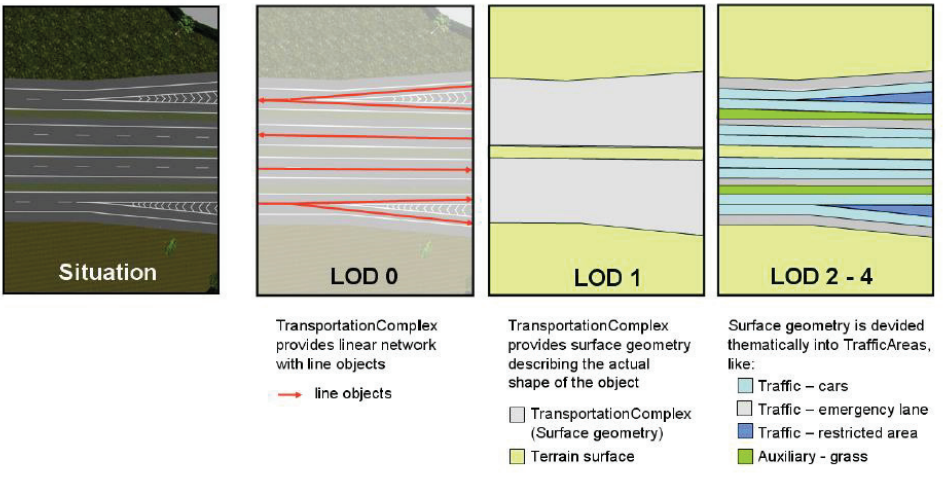


Figure 6.6: Transportation Levels of Detail in CityGML

---

[10] All references to CityGML in this thesis are based on version 2.0.

One of the main objectives of the road network design is the opportunity to switch between levels of detail. For transport and traffic models, these levels of detail have to align to the various types of assignment. We have already shown that *CityGML* combines different levels of detail. The design approach to model the data objects of the transport infrastructure for road based traffic therefore starts with two principles:

- *Relationships between different Levels of detail*
  As is shown in Figure 6.6, a road segment can be represented at various levels of detail. Within CityGML(2.0), the lowest level (0) represents a road as a line object whereas higher levels of detail (1-4) add information about its geometry and functionality. The aggregation / decomposition is supported by providing an explicit generalisation association between city objects (Consortium (2012)).
- *Coherent modelling of semantics and geometrical/topological properties*
  This coherence is described in the CityGML manual (Consortium (2012)): "At the semantic level, real-world entities are represented by features", such as roads, lanes, barriers and traffic lights. "The description also includes attributes, relations and aggregation hierarchies (part-whole-relations) between features. Thus, the part-of-relationship between features can be derived at the semantic level only, without considering geometry. However, at the spatial level, geometry objects are assigned to features representing their spatial location and extent. So the data model consists of two hierarchies: the semantic and the geometrical in which the corresponding objects are linked by relationships (Stadler and Kolbe (2007)). The advantage of this approach is that it can be navigated in both hierarchies and between both hierarchies arbitrarily, for answering thematic and/or geometrical queries or performing analyses. If both hierarchies exist for a specific object, they must be coherent (i.e. it must be ensured that they match and fit together)."

Within CityGML, the *TransportationComplex* represents a superclass for four distinct types *Track, Road, Railway,* and *Square*. In addition to the *class,* every *TransportationComplex* has the attributes *function* and *usage*. The attribute *function* describes the purpose of the object, for example a national motorway or an urban road, while the attribute *usage* can be used, if the actual usage differs from the *function*. An example is a country road where the admittance of bicycles is typically situation dependent. The *TrafficArea* and *AuxiliaryTrafficArea*, being the components of the *Transportation-Complex*, provide information about the specific components of the *TransportationComplex*. The *function* again describes the object's main use, for example a *driving lane* or a *parking area*, while the *usage* attribute indicates which specific modes of transportation can use it. This allows various specifications of a lane: a bicycle lane for instance, could be used by both bicycles and pedestrians, or just by bicycles alone. The attribute *surfaceMaterial* specifies the type of pavement and may also be used for *AuxiliaryTrafficAreas* (e.g. asphalt, concrete, soil, rail, grass). The *AuxiliaryTrafficArea* defines the part of the road that is generally not meant for movements, but are elements that characterize a road, for example a kerb, a median, or green areas (see Figure 6.7).

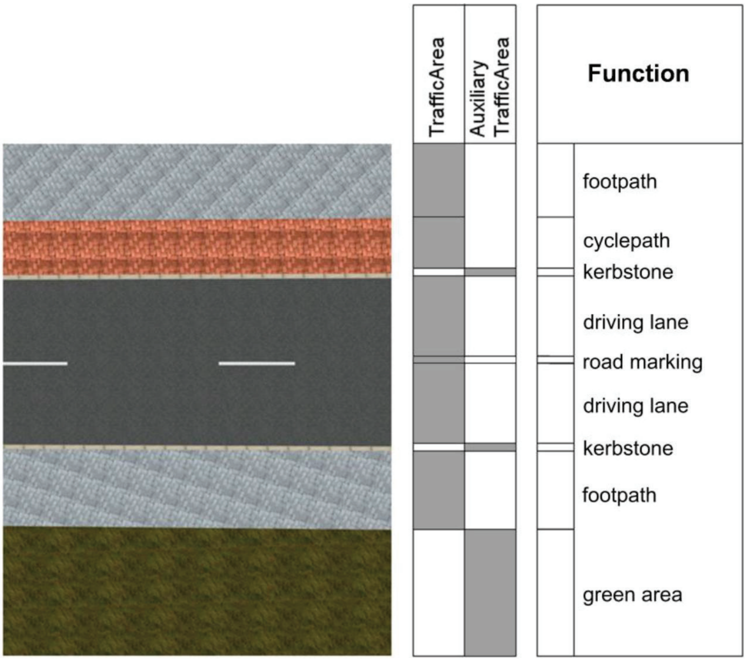


Figure 6.7: example of a Cross Section profile of a road (source: Consortium (2012))

The attributes *function* and *usage* are provided by enumerated lists. While CityGML provides code-lists for most attributes, additional metadata may be defined by using attributes from pre-defined catalogues. These attribute catalogues may be customer- or country-specific.

6.4.2 Linking model objects to the data design of geographical information systems

This section discusses the CityGML data formats in relation to the data requirements for transport and traffic models. In the first sub section, the main concepts of the transport domain are modelled compliant to CityGML. The opportunities for improvements and extensions of CityGML model are investigated, in order to facilitate information that is required in the transport domain.

A mutual exchange of data between models and databases, requires objects within the transport and traffic model to comply with the format of the data standards. To put it in other words, every concept in the transport domain should have an equivalent class in CityGML. This is called *mapping of concepts*. In this section a mapping is done by comparing the main object types that are identified in the requirement analysis for transport and traffic models (see for example Tamminga, van den Brink et al. (2013)) to CityGML using the same approach as in van den Brink, Stoter et al. (2012). The results of our mapping for network and scene contexts are summarized in Table 6.1 respectively Table 6.2.

Table 6.1: Mapping between requirements for network objects and CityGML classes

Requirement		CityGML Class	Level of detail
Link based objects			
Link	*Lines establishing a linear network*	TransportationComplex	LOD0-4
		has LOD0 Network	
- Geometry	Centreline of a road (with driving direction)	differentiated into Road, Track, Railway and Square	
	Centreline along the median of a road (with driving direction) with vertices		
	Centreline along the median of a road (with driving direction) with vertices and curvature		
- Usage	Highway, on-ramp, cycle path etc.	TransportationComplex function or TrafficArea function (for segments)	LOD0-4 LOD2-4
- Functionality	Allowed vehicle types	TransportationComplex usage or TrafficArea usage (for segments)	LOD0-4 LOD2-4
- Attributes	Length	-	
	Maximum legal speed	-	
	Number of lanes	-	
	Capacity	-	
	Gradient (%)	-	
Cross Section	*Cross section representation of a road*		
- Element list	Complete list of Cross section elements (such as driving lane, barrier, cycle path)	TrafficArea + AuxiliaryTrafficArea	LOD2-4
	List of Drivable Cross section elements	TrafficArea	LOD2-4
Cross Section Element			
Drivable	*Allowed for driving such as driving lane(s) or cycle path*	TrafficArea	LOD2-4
- Attributes	Type (such as cycle path, driving lane)	TrafficArea function	LOD2-4
	Maximum legal speed	-	
	Allowed vehicle types	TrafficArea usage	LOD2-4
	Width	derivable from surface geometry	
	Max. height	-	
	Road marking along	TrafficArea function	LOD2-4

Table 6.1: Mapping between requirements for network objects and CityGML classes (continued)

Undrivable	*Not allowed to drive*	AuxiliaryTrafficArea	LOD2-4
- Attributes	Type (such as barrier, median)	AuxiliaryTrafficArea function	LOD2-4
	Width	derivable from surface geometry	
	Height	derivable from surface geometry	
Spot elements		CityFurniture class=traffic	LOD1-4
Detector	*Sensor measuring speed, vehicles etc.*	CityFurniture function=[missing]	LOD1-4
- Geometry	Polygon	CityFurniture lod[1-4]Geometry	LOD1-4
Traffic sign / display	*Provide information to road drivers*	CityFurniture function = communication fixture | traffic light | free-standing sign | free-standing warning sign	LOD1-4
- usage	Indicate behaviour	-	
- attribute	Mandatory	-	
	Priority	-	
	Direction for turning	-	
	Signpost		
	Warning	-	
	Information	-	
Stop line	*Position at lane where traffic halts before junction*	TrafficArea function = road_marking_stop	LOD2-4
Car park entry/exit		TrafficArea function=car_park	LOD2-4
- Geometry	width and height	surface geometry	
- Attributes	Handling time in case of barriers	-	
Junction area	*Area: outer borders and interior of an intersection*	Square (specialization of TransportationComplex) with surface geometry	LOD1-4
- Geometry	Polygon (intersecting roads / lanes)	Square lod1MultiSurface	LOD1-4
- Interior	lane connectors	-	
- Typology	Junction control info	-	

Table 6.1: Mapping between requirements for network objects and CityGML classes (continued)

Car Park objects			
Parking lots			
- Geometry	Polygon per lot	TrafficArea function=parking_lay_by	LOD2-4
	Polygon with number of lots by level	-	
	Polygon/link with number of lots	-	
Public Transport			
Bus/tram stop		CityFurniture function=bus stop	LOD1-4
- Geometry	Location (point)	lod[1-4]Geometry can contain any geometry (one per LOD)	
	Location (point and link to section element)		
	Location and detailed geometry		
- Attributes	Stops: Line number(s)	-	
Tram/bus	*Geometry of the rail infra / bus lane*	Railway Road usage=bus	LOD0-4
- Geometry	Infra defined as series of nodes	lod0Network geometry representation	
	Infra defined in cross section elements (drivable see row B 1-7)	-	
	Bus/tram line number(s) using links/cross section elements	-	

Table 6.2: Mapping between requirements for transport and traffic models based on Scene Space and CityGML classes

Object		CityGML Class	Levels of detail
Accessible space for pedestrians(open air)			
Square/plaza		Square (specialization of TransportationComplex)	LOD1-4
- geometry	Open area: unbounded polygon	multisurface geometry (polygon)	LOD1-4
- with objects inside	*Such as a bookstand, statue, flower bed, city furniture*	CityFurniture	LOD1-4
- geometry	3D (length/width or diameter, height)	multisurface geometry with 3D coordinates	LOD1-4
Pedestrian crossing		TrafficArea function= road_marking_crosswalk	LOD2-4
- geometry	Polygon	multisurface	
Gateway	*such as entrance of shops, offices, museums*	Door	
- geometry	Opening in open areas (width and position)	Building or CityFurniture	LOD0-4 LOD1-4
	Building with entrance(s) (2D or 3D)	Building with Door	LOD0-4 LOD3-4
- attributes	Type of entrance (normal door, turning door, opening)	-	
Accessible space for pedestrians in Transfer Locations			
Accessible space within Public Transport Transfer	*Hall, ramp*	Building CityFurniture function=ramp	LOD0-4 LOD1-4
- geometry	Accessible area bounded by walls	Building multisurface or solid Wall surface	LOD1-4 LOD2-4
- objects inside	*Such as a ticket machines, meeting point, shops, toilets*	BuildingFurniture (movable) and IntBuildingInstallation (nonmovable)	LOD4
- geometry	3D (length/width or diameter, height)	3D point, line, surface, or solid	LOD0-4 (depending on object class)

Table 6.2: Mapping between requirements for transport and traffic models based on Scene Space and CityGML classes (continued)

Accessible space- within Car Parks	*Parking level, entrance halls*	Building	LOD0-4
- geometry	Accessible area bounded by walls	Building multisurface or solid	LOD1-4
		Wall surface	LOD2-4
- objects inside	*Such as a ticket machines, parked vehicles*	BuildingFurniture (movable) and IntBuildingInstallation (nonmovable)	LOD4
Connecting elements between levels	*Escalator, steps*	Room function=escalator	LOD4
		IntBuildingInstallation or Room function=stairs	LOD4
Gateways connecting accessible areas	*Doors, corridors*	Door	LOD3-4
		Room	LOD4

From this mapping exercise, we can draw the conclusion that there is an extensive semantic match between the requirements of the transport domain and the CityGML semantic model. In fact, for the majority of objects, a match can be found in CityGML. Though the majority of concepts can be matched by CityGML objects, this does not account for every attribute and object.

We notice the following shortcomings:

- Attributes of links and lanes, such as legal speed limits and sensors, are lacking;
- The network representation only supports one LoD (LoD0) and this network is not related to the surface representation at different LoDs;
- CityGML does not specify how to model intersections and roundabouts.

This limits the usability of the current version of CityGML (2.0) within certain transportation applications and doesn't align with LoDs present in other CityGML modules.

6.5 Ingredients to improve the representation of transportation in CityGML

In this section we propose adjustments and extensions to CityGML in order to comply to the demands from transport and traffic models.

6.5.1 Extending the CityGML data standard for transport and traffic

There are two ways to add these properties to CityGML: by using generic attributes (OGC 2012: p.156) or by adding them to an ADE. Because the aim of our research is standardization of input data for transport and traffic models, a formal CityGML extension in the form of an ADE is preferred over the use of generic attributes.

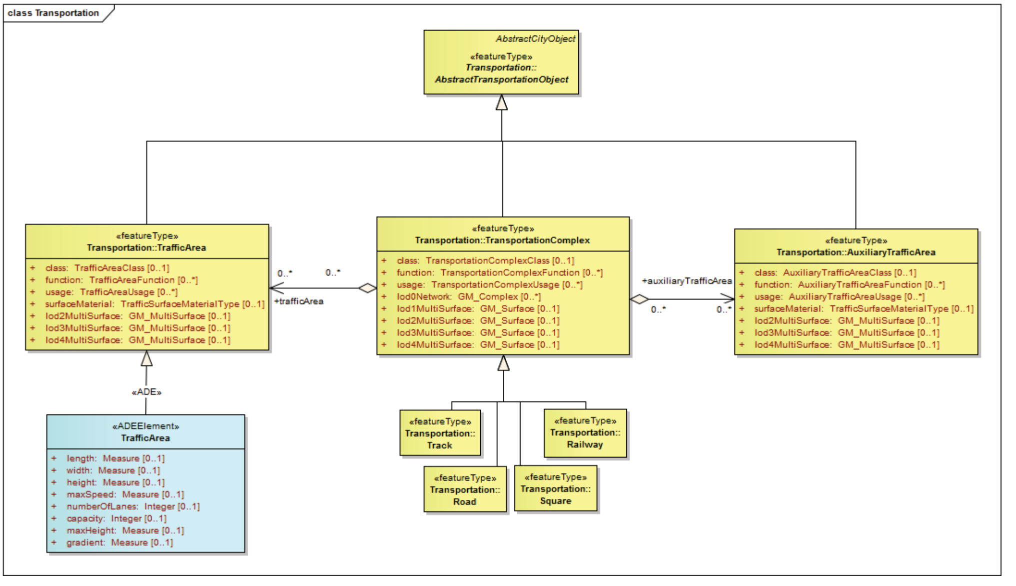


Figure 6.8: Example of CityGML extension

In this Figure, the CityGML classes are depicted in yellow and the extension class is depicted in blue. The extension class is modelled as a subclass of the CityGML class, and intentionally has the same name as the corresponding CityGML class. That, and the stereotype <ADEElement> are indicators that the extension class models the same concept as the CityGML class and only adds extra properties to it. This way of modelling is described in Van den Brink, Stoter et al. (2012), van den Brink, Stoter et al. (2013).

The extra properties are added to the TrafficArea class that provides the highest level of detail. This seems the best fit for the transport and traffic modelling domain, where these properties are required at the link or even lane level. If we were to add the properties to TransportationComplex, the properties would be added at the network level. All properties are optional ([0..1]: indicating they may appear zero or one time, because none of the properties are required for all use cases. Length, width, height and maxSpeed are of type Measure, allowing the unit of measure to be indicated with the value. In the same way, other classes could be extended with properties. For example, the CityGML class *Door* could be extended with a property for the type of door. Traffic signs can be represented by CityFurniture objects in combination with a custom code list for the classification of traffic signs (Mandatory, Priority, Direction, Warning, and Information). Alternatively, if further study of traffic sign indicates a requirement for more detailed modelling of traffic signs, CityGML could be extended with a new class TrafficSign as a subclass of CityFurniture.

The subsequent sections discuss the further requirements and opportunities to create a successful link between the CityGML data structures and transport and traffic model objects and focus on:

- Redefine the Levels of Detail for the transport infrastructure;
- Representation of lanes and carriageways as a graph;
- Defining (the interior) of junctions.

6.5.2 Improving the CityGML representation of roads at various levels of detail

As appeared from the previous section, and also discussed by others (for instance Beil (2017)), the current design of the CityGML *TransportationComplex* provides a basic framework for the representation of the infrastructure, but lacks a thorough specification of the road objects. A further clarification and elucidation of all LoD´s to attain a comprehensive set of objects and attributes that covers the majority of aspects of the transportation infrastructure, appears useful. Becker, Nagel et al. (2013) developed the concept and classes for *utility networks* as an ADE of CityGML, and addresses issues that seem also applicable for the design of the transportation infrastructures. They provide a structural description of the entire network by graph structures with a 2 and 3D embedding that is suitable to calculate the relative position between network objects of different modalities, their spatial relation to each other as well as their spatial relation to other objects of the city, such as buildings, waterways and city furniture (Becker, Nagel et al. (2011)).

Analogous to the reasoning for utility networks (Becker, Nagel et al. (2013)), an integrated 2 and 3D data model of transportation networks that enables the representation of various modes has to meet the following requirement:

1. **R-GIS.7**
   An integrated data-model for multi-modal networks, including the transfer options between modes, must support the simultaneous representation of these networks.

The main concept of such modelling approach is the dual representation of a network feature, according to which each network component can be represented both by its 3D topography and by means of a complementary graph structure. The graph structure represents its functional, structural and topological aspects. This concept differs substantially from previous data-modelling approaches which usually map the entire road network onto a single topological network graph derived from a classification of network components into point-like and line-like objects (Becker, Nagel et al. (2013)). Building on this idea, Beil (2017) proposes a further elaboration and specification for all of the LoD's, and a subdivision of LoD2-4 into:

- LoD2:
  This level resembles the LOD2-4 from CityGML version 2.0, but adds MultiSolids for the representation of solid bodies.
- LoD3:
  This level adds two elements: (1) a further refinement of the surface geometry, for instance to represent gully tops and potholes, and (2) a linear representation of the driving lanes.

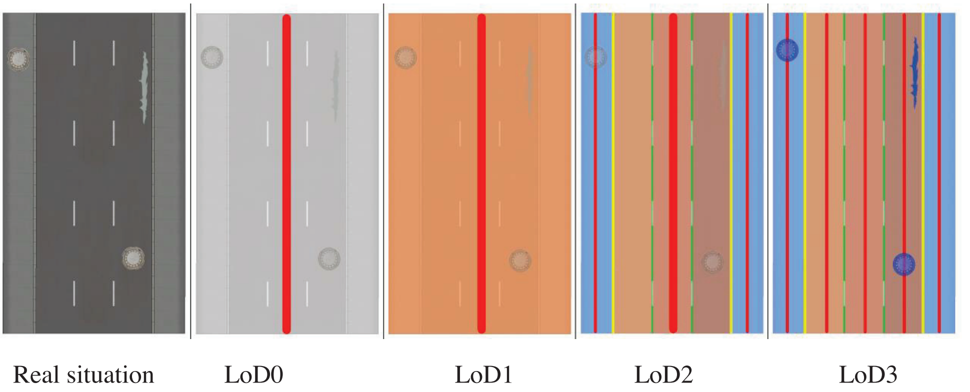


Figure 6.9: proposal for an improved LoD design (source: Beil (2017))

In addition to the line representation in LoD 2 and 3, Labetski, van Gerwen et al. (2018) proposes:

- Make the presence of nodes explicit for LoD1 - LoD3 network representation.
- Differentiate further between LoD2 and LoD3 by introducing carriageway representation for LoD2 and lane representation for LoD3.

In section 6.6 this proposal is further elaborated.

6.5.3 The representation of junctions

A major item that *CityGML* (version 2.0) does not address, is the representation of junctions. Other standards such as *GDF*, *OpenDrive* and the internationally standardised topology message *MAP* for intersections (SAE J2735, ISO TS 19091) do provide ingredients for a comprehensive junction representation. Their approaches are shortly introduced, and for every standard we will highlight the concepts and approaches that could be useful for both the extension of *CityGML*, and the design of the transport infrastructure for road based modelling.

GDF explicitly defines rules for defining junction areas. Figure 6.10 shows a schematic representation of the roads that enter a junction, with a contour of the enclosed traffic area.

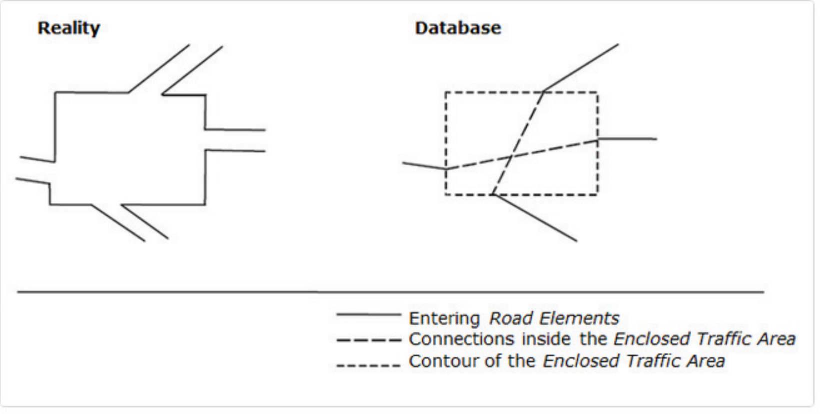


Figure 6.10: junction lay out in GDF

Within this polygon a further elaboration of the driving lines is desirable for a full description of the connections between the in- and outgoing arms of the junction area. To represent a junction, *OpenDRIVE* defines *paths* for the movements within a JunctionArea as shown in Figure 6.11. Roundabouts consist of a collection of T-shaped junctions, and are represented by a JunctionGroup. The signals at a junction, as well as other signs along a road¸ are defined relative to the longitudinal position along a road, and, optionally, their validity by lane and direction.

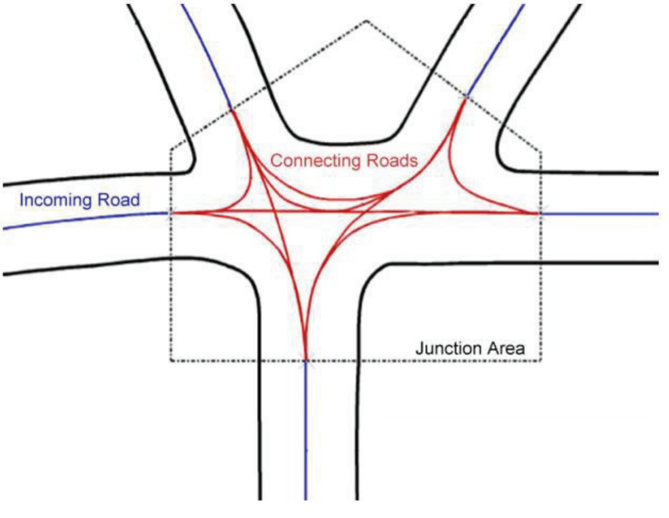


Figure 6.11: junction lay out in OpenDRIVE (source: Dupuis (2015))

A likewise approach is provided by the Intersection Topology Format (MAP/ITF). While GDF and OpenDrive provide a data standard for complete transportation networks, the MAP/ITF format focusses on the description of intersections (Vreeswijk, Claassens et al. (2016)). It is largely based on the internationally standardised topology message MAP (SAE J2735, ISO TS

19091). The configuration describes one or more intersections which are controlled by a single traffic light controller. It contains both the intersection geometry and its relations with the traffic light controller. Basically, it requires information about the approaching and egress lanes, its sensors/detectors, the vehicle types that are allowed to use specific lanes, and the lane connections at the junction (turning information). Figure 6.12 visualizes the lanes as part of the MAP/ITF geometry. The purple arrow represents an incoming lane, coming from the east. This lane is connected to all exiting arms, which is visualized by the yellow ochre connectors within the junction area.

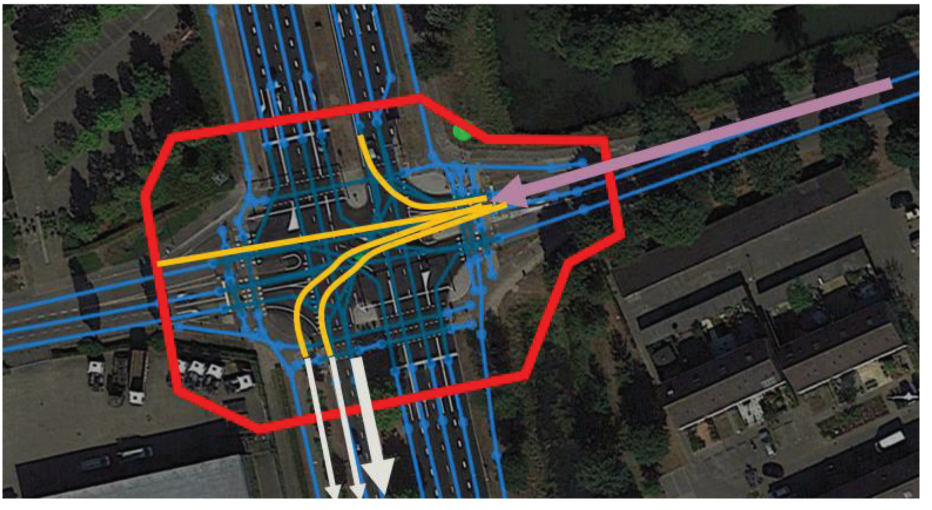


Figure 6.12: junction lay out in OpenDRIVE

Based on the ITF/MAP representation of junctions at the level of lanes and the idea to represent roads at three levels of detail (section 6.5.2), the next section proposes a representation of both roads and junctions at three levels of detail: road, carriageway and lane.

6.6 Proposal for a GIS representation of roads and junctions at three levels of detail

We propose a line based representation of roads and junctions at three levels of detail, and combine it with a 3D topography of the road. Again, the 3D representation can be divided into levels of detail. The representations then can be combined at every level. For the transport and traffic models the line representations are specifically relevant, as it provides the basis for the graph based representation of the networks. In addition, the representation of objects in 3D allows a (model) user to combine data from both sources.

Based on the previously presented elaborations, CROW[11] developed the "information model roads and transport" (IMWV) with a combined representation of lines and spaces (Figure 6.13). The blue line represents the road which contains two carriageways (red lines). The lanes of the carriageway are represented by the green dotted lines. The relationships between the objects is explicitly defined, and thus provides the connection between a specific speed limit and a carriageway or lane.

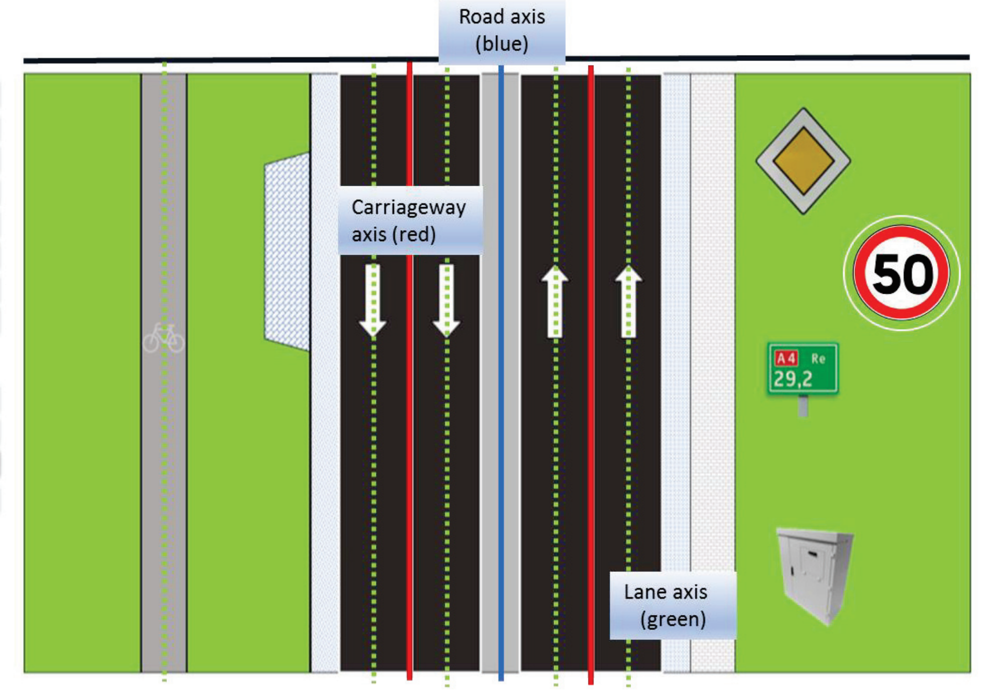
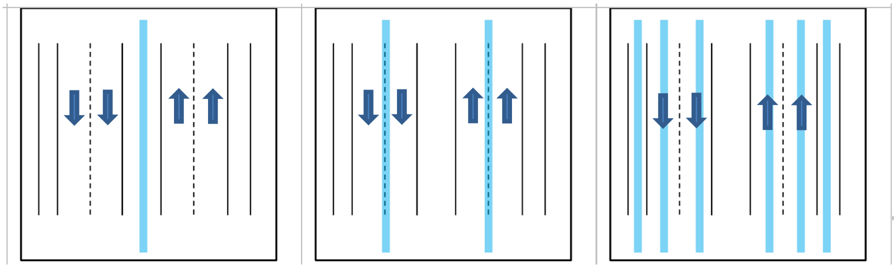


Figure 6.13: Combined representation of a road by lines and spaces (source: Marleen Hovens, CROW)

Most of the current transport and traffic models apply a line representation of the road network. In the remainder of this section we further elaborate on the line representation of the various LoD's. Figure 6.14 shows a motorway with 2 lanes and a shoulder for both driving directions. At LoD1, the road is represented by one line (painted in blue), defined as the *middle* of the two *inner* lanes. LoD2 represents the carriageways by two lines: for both driving directions a line, situated in the middle of the driving lanes. LoD3 represents the driving lanes and the shoulder by separate lines: 3 lines for the southbound direction and 3 lines for the opposite driving direction.

---

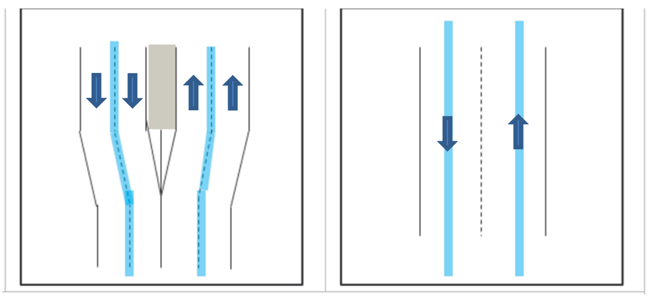
[11] CROW is the Ducth technology platform for transport, infrastructure and public space. The IMWV project has been managed by Marleen Hovens



a) road (LoD1) b) carriageway (LoD2) c) lane (LoD3)

*Figure 6.14: proposal for the representation of a motorway (2 * 2 lanes and shoulder) at three levels of detail (blue lines)*

Usually, a single carriageway is defined as a collection of lanes on which a vehicle is not restricted by any physical barriers or separation[12]. Thus, a carriageway can have two (opposite) driving directions. The LoD2 is not equivalent to the definition of a carriageway. As Figure 6.15 shows, the access road (b) represents a single carriageway, but the LoD2 representation distinguishes separate lines for each direction. The rationale is to enable a clear representation of both driving directions which is also useful for the representation of connections at junctions. Traffic simulation models that include the possibility for overtaking and temporary use the lane of the opposite driving direction, require the combination of both lanes. For this aspect, the combination of the line-based with the 3D representation provides the relationships to integrate both directions.



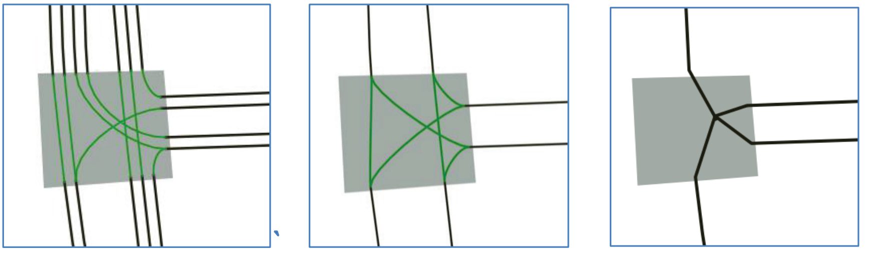
a) motorway (2*2 lanes) b) access road (2*1 lane)

Figure 6.15: proposal for representation at LoD2 (carriageways)

Junctions are also represented at three LoD's. Figure 6.16 shows the junction area (represented by a polygon) and visualizes the allowed turning movements from an entrance lane to an exit lane by the green coloured connectors. Likewise, LoD2 connects the carriageways by direction:

[12] *https://www.gov.uk/guidance/the-highway-code/general-rules-techniques-and-advice-for-all-drivers-and-riders-103-to-158*

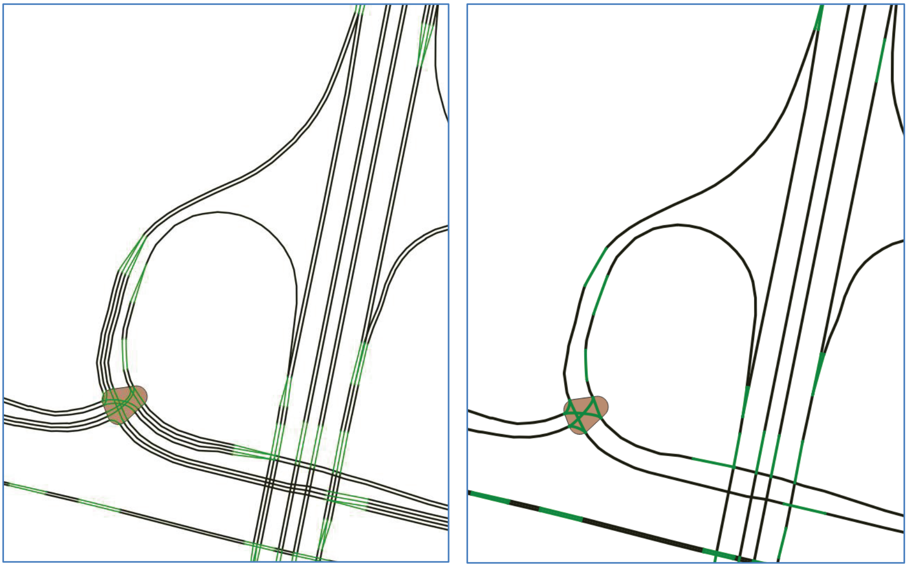
the green lines show the connections between the entering and exiting carriageways. LoD1 represents the roads and does not distinguish the turning movements at junctions.



a. LoD3 *b. LoD2* *c. LoD1*

Figure 6.16: proposal for the representation of junctions at three levels of detail[13]

An example of a network is shown in Figure 6.17 and illustrates the representation of roads and junctions for LoD 2 and LoD3. There is a one-to-one relationship between every group of parallel lanes and the corresponding carriageway section (by direction). The lanes at LoD3 thus represent the *cross section* of a carriageway link-object (LoD2).



a) at LoD3 *b) LoD2*

Figure 6.17: example of the representation of a small network by (a) lane and (b) carriageway (by direction)

[13] The figure is based on a project carried out by Sweco and commissioned by Stefan van Gerwen from the province of Noord-Brabant

In the next chapter, we explore if the proposed dual approach with a 3D representation of road objects in conjunction with a graph based description (at the level of detail of lanes and carriageway) matches the design of the road infrastructure for transport and traffic modelling.

6.7 Evaluation of requirements: a geospatial data standard for transport and traffic

The previous sections identified a list of requirements. The choice of CityGML appears a suitable choice, although it does not meet all of our requirements. To meet the resulting requirements, the proposed extensions (more objects) and further elaborations on the level of details with a linear representation of roads and lanes appear necessary.

- **R-GIS.1**
  Data objects in transport and traffic models should align to their counterpart in GIS databases.
  The analysis in section 6.4 shows that although CityGML (2.0) already covers a large share of the objects, an extension is required. The proposal in 6.6 shows an example of such an elaboration and presents the Dutch information model for roads and transport (IMWV) and combine a line and 3D representations that aligns to the requirements for transport and traffic models.

- **R-GIS.2**
  A geometrical, topological, and functional embedding of the infrastructures into the urban space must be done.
  CityGML already includes both the transportation infrastructure and the urban space with rich semantics at a high level of detail. The proposed dual approach with a 3D representation of road objects in conjunction with a graph based description, further supports this embedding of the road infrastructure.

- **R-GIS.3**
  The external database should provide the data for the modelled infrastructure at various levels of detail, in order to allow the exchange information at all levels of detail.
  Because CityGML (version 2.0) lacks an elaboration of the Levels of Detail for roads and junctions, we propose an extension of CityGML with three LoD´s that includes the representation of junctions. This proposal aligns to the requirements for transport and traffic models at various levels of detail.

- **R-GIS.4**
  There should be a correspondence between the links at various levels of detail, in order to exchange information between the levels of detail.
  The design of the levels of detail within CityGML ensures the correspondence between the levels of detail (section 6.4.1).

- **R-GIS.5**
  There should be a relation between the junction-description at various levels of detail. The proposal in section 6.6 includes an elaboration of the representation of junctions and the allowed movements and distinguishes the three levels of detail: turning lane, carriageways and road.

- **R-GIS.6**
  The exchange of data between models and external databases requires the availability of a mature and complete geospatial standard for transportation infrastructure
  As we conclude there is yet no geospatial data standard that fully covers the transportation infrastructure. While CityGML may be a good solution for addressing the shortcomings of the other standards and ensuring the representation of multi modal infrastructures in 3D city models, it requires a further elaboration for the transportation infrastructure. In section 6.5 and 6.6 we present proposals for further improvements.

- **R-GIS.7**
  An integrated data-model for multi-modal networks, including the transfers between modes, must support the simultaneous representation of these networks.
  CityGML defines classes and relations for most occurring topographic objects in cities and rural area (section 6.3.5), which supports the representation of multi modal infrastructures. By means of an Application Domain Extensions (ADE) it even provides information for pedestrian indoor navigation. As already mentioned, an extension for some additional objects and attributes still appears necessary.

6.8 Conclusions

To feed transport and traffic models with data and vice versa, a straightforward relation between the objects in transport and traffic models and external data sources is desirable. A sustainable exchange between external data sources and transport and traffic model objects, requires data definitions that align to internationally accepted geospatial data models. Within the existing (open) data standards, CityGML may be a good solution as it (1) covers the complete urban and rural environment which supports topological consistency, (2) distinguishes various levels of detail and (3) ensures the representation of multi modal infrastructures. Yet, the transportation domain of CityGML (version 2.0) shows some limitations. Table 6.3 provides an overview of the requirements for the geospatial data standard, and indicates if CityGML matches these requirements or requires improvements. While *CityGML* provides a full description of the urban and regional space, and describes it at various levels of detail, it lacks information on some specific aspects with regards to transportation. The proposed extensions result in a standard that appears to fulfil all the requirements.

Table 6.3: Overview of the requirements for the geospatial data standard

Requirements	**Meet the requirements?**	
	CityGML (version 2.0)	Proposed extension
Data objects in transport and traffic models align to their counterpart in GIS databases.	Only partly (i.e. signs are incomplete)	Yes
A geometrical, topological, and functional embedding of the infrastructures into the urban space must be done	☑	
The external database should provide the data for the modelled infrastructure at various levels of detail, in order to allow the exchange information at all levels of detail	No linear representation of all Levels of Detail	Yes
There should be a correspondence between the links at various levels of detail, in order to exchange information between the levels of detail.	The link description is not available at LoD2-3	Yes
There should be a relation between the junction-description at various levels of detail	No junction description	Yes
The exchange of data between models and external databases requires the availability of a mature and complete geospatial standard for transportation infrastructure	Not complete: some objects are lacking or not fully specified	Yes
An integrated data-model for multi-modal networks, including the transfers between modes, must support the simultaneous representation of these networks	Pedestrian infrastructure is part of CityGML, but the linear representation is lacking	Yes

An improvement of the transportation objects within CityGML by means of an application domain extension (ADE) thus appears necessary, in order to create a comprehensive description of the road and junction lay-out at various levels of detail, which is currently lacking.

Table 6.4 summarizes the proposed extensions. The dual approach with a 3D representation of road objects in conjunction with a graph based description at three levels of detail would increase the usability. It facilitates the exchange of data between transport and traffic models and external databases.

Table 6.4: Proposal for extending CityGML objects

Proposed addition
Linear representations of a network by road (LoD1), carriageway by direction (LoD2) and lane (LoD3)
Junction (polygon) with a linear representation of the inner connections (at LoD 1-3)
Traffic signs with a direct linkage to roads and intersections by LoD
Traffic sensors (to detect vehicles)
Functional systems such as a traffic light controller

The next chapter presents the development of the data model of *OpenTrafficSim* according to these scalability principles.

Chapter 7

A novel design of the transport infrastructure for traffic simulation models

While the previous chapter discussed the exchange of transport and traffic model objects with external standardized data sources, this chapter focusses on the design of network objects for transport and traffic models in order to represent the road infrastructure, the means of transport, and the connection with the activity locations. The design addresses the specific requirements for a road based transport infrastructure that have been identified in chapter 5. A novel aspect in this design is the ability to represent the objects at different levels of detail, which enables the modeller to build one transport infrastructure that is applicable at different levels of detail. In addition, the matching (or synchronization) of these model objects with internationally accepted data standards enables an efficient building of transport and traffic models. Once the data are gathered, the model input can be directly derived from the spatial databases. This increases both the efficiency of creating models, and diminishes the chances of flaws in the digitizing process of objects.

7.1 Introduction

Traffic and transport models describe the impact of people's activity choices on transport and traffic, and are used for studies that range from long term issues, such as land-use and road infrastructure planning, towards more operational issues such as traffic management and control. This requires transport and traffic models with different approaches, and varying levels of detail. The most detailed microscopic models describe the behaviour and movement of people at an individual level. This chapter presents the design of the objects that compose the transport infrastructure for road based traffic models at this high level of detail, but also satisfies the requirements for the modelling at lower levels of detail (mesoscopic and macroscopic models). Part of this design is already implemented in *OpenTrafficSim* (version 1.01). In the remainder of this chapter the implemented *OpenTrafficSim* class names of objects and attributes are used and printed in **bold**. The other objects and classes that are not implemented in *OpenTrafficSim* (version 1.01) are printed in *italics*.

The design of the infrastructure in OpenTrafficSim is based on the joint work of the OpenTrafficSim developers: Alexander Verbraeck, Peter Knoppers, Wouter Schakel, Guus Tamminga and Hans van Lint (Delft University of Technology). The actual implementation in the OpenTrafficSim code is done by Alexander Verbraeck, Peter Knoppers and Wouter Schakel.

Before we present the design of the infrastructure and their connections with the activity locations, section 7.2 introduces the Generic Travel Unit (GTU) as the basic entity for movement over the transportation infrastructure and elaborates the design of various types of vehicles.

Subsequently, we focus on the infrastructure and show how it enables a highly detailed representation for microsimulation approaches. The transportation infrastructure is divided into three specific categories:

- Road network (section 7.3)
  This describes the topology and geometry of the road lanes, and includes street furniture, such as signs and barriers.
- Functional systems (section 7.4)
  The functional systems describe these entities that have a specific role in the transportation infrastructure. They merely combine network components and devices. Traditional examples are controlled intersections, stations, bridges and car parks. ITS developments provide new systems that combine vehicle and road side systems by wireless communication.
- Devices (section 7.4.4)
  The road network is complemented by various types of devices that are used to monitor, inform, and manage traffic. They are divided in sub-classes: measurement (speed detection, count), control (traffic light) and information (route information panel).

After this overview, section 7.5 addresses specific issues with respect to the implementation of the road infrastructure in transport and traffic models, such as the connection between the transport infrastructure and the activity locations, the transfers between various modes, and design rules (for instance how to present a dual carriageway: as one bidirectional link or two one-way links).

After the design of the road network infrastructure that provides a high level of detail, section 0 describes the possibilities to switch to lower levels of detail.

To apply this design in practice, and provide opportunities to exchange data between transport and traffic models and external databases, an elaboration of open data standards for geo-spatial information is required (section 7.7). Section 7.8 explores the opportunities to generate new datasets that align to this new design, from available data-sources in the Netherlands. The chapter closes with the requirements analysis (7.9) and the conclusions (section 7.10).

7.2 Means of travel and transport: the Generic Travel Unit (GTU)

A proper design of the infrastructure provides a basis for the simulation of trips between the activity locations. To simulate these trips, the simulation model needs traffic units to move people between these locations. In this section we explore the design of these traffic units, with the aim to provide a framework that is capable to model multiple modes at various levels of detail.

The design of the traffic units is shown in Figure 7.1, and defines the Generic Travel Unit (**AbstractGTU**) as basic unit. The idea of the Generic Travel Unit (GTU) has been proposed in van Lint, Schakel et al. (2016) and describes the opportunities of this design to provide maximum flexibility for simulation purposes.

The ancestor of all further **AbstractGTU** extensions is an object that has a (directed) location, dimensions, and some properties such as the maximum speed that all GTU's share. The

AbstractGTU is not yet bound to any infrastructure. Its descendants add further attributes and behaviour. The first major schism in the class hierarchy for road based traffic is between a **Person** and a **Vehicle** type. The **Person** object is already introduced in section 4.2.1, and implements (personal) characteristics and behaviours (e.g. gender, age, reaction time, etc.). The **LaneBasedGTU** adds mechanical and physical characteristics and behaviours (e.g. engine capabilities, dimensions, etc.) and defines it as a vehicle such as a car, bicycle bus or truck. The proposed *Mover* class defines the role of a Person: being either a driver, walker of passenger. In addition to the driver, a **LaneBasedGTU** may have passenger(s), a series of Persons that is transported by a vehicle, who do not actively engage in driving operations. This framework makes a clear distinction between a vehicle and a person, and thus provides a sound architecture for the simulation of a work trip that starts by walking from home to the location of the car, a drive to the train station, park the car and walk to the platform, continue by train, and finally walk to the office.

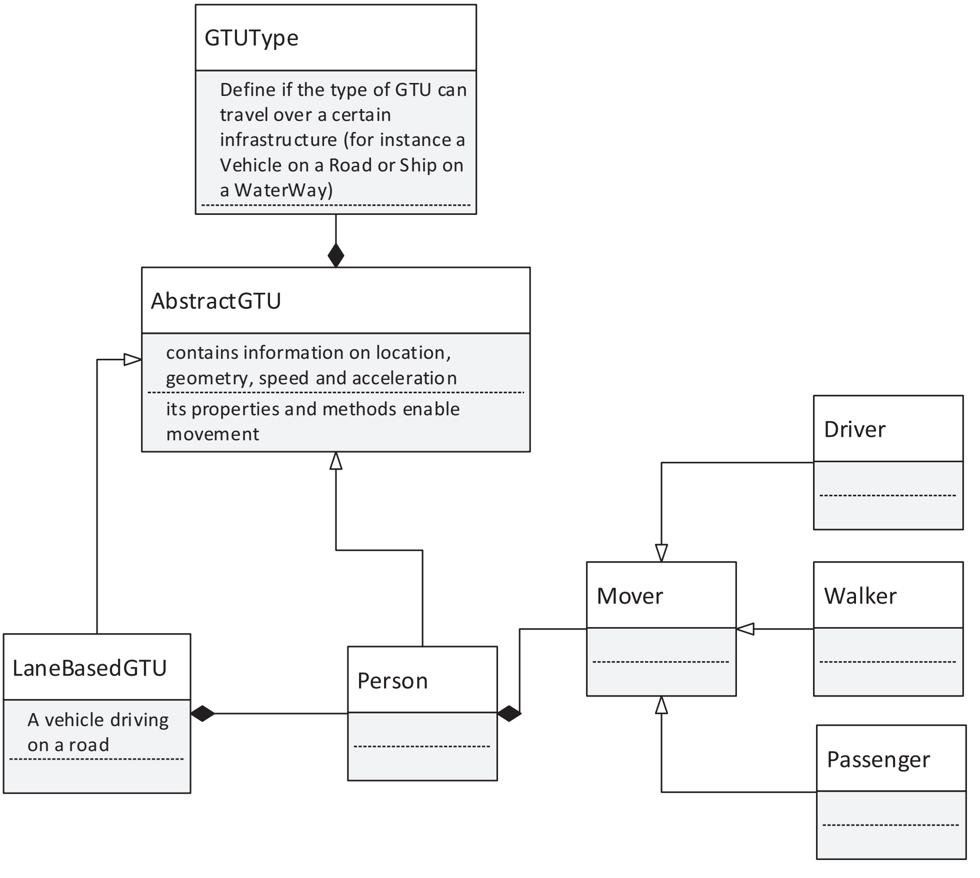


Figure 7.1: The generic travel unit as a basis for traffic units (based on van Lint, Schakel et al. (2016)

This generic structure allows the implementation of most microscopic traffic models (both 2-D pedestrian models, and longitudinal and lateral models for vehicles). Moreover, the ability to connect a person with a vehicle, enables the modeller to actually create a transport and traffic modelling environment that reflects the entire process from activity planning towards trip execution, and allows the implementation of recursive processes; for instance, reconsidering an activity pattern in case of heavy congestion. This design makes *OpenTrafficSim* typically appropriate for agent based simulations.

7.3 The road network

The transportation infrastructure defines the elements that describe the road network and its related equipment: (1) the street furniture and (2) the systems to manage, control, inform and warn traffic. The succeeding sections describe the objects (and their interrelationships) that are used to represent the road infrastructure in a transport and traffic model. As will be shown, this design covers both the requirements for microsimulation approaches, as well as the more aggregated approaches (meso- and macroscopic). In chapter 5 we already identified a series of requirements for creating the modelled infrastructure. Section 7.9 describes if the design of the modelled infrastructure satisfies those requirements. The requirements relate to (1) a valid topology and (2) a valid description of the road infrastructure for tactical and operational driving decisions.

1 Topology: links and nodes

In order to create valid routes, the road infrastructure (1) requires a graph that captures the topological relationships of the roads and its lanes that make up the network, and (2) provides a geometric representation to model the movement of vehicles and pedestrians (Wilkie, Sewall et al. (2015), Fellendorf and Vortisch (2010), Willemsen, Kearney et al. (2006)).

To create such a representation, a valid topology is a *first* requirement. Topology is the mathematical concept of spatial structure, sometimes defined as "characteristics of geometry that do not change when the coordinate space is deformed". It expresses explicit geometric relationships, such as "connects to, touches, adjacent to, or within". A topological model consists of topological primitives (nodes, edges, and faces) with references defining spatial relationships between them. As with scalability, a valid topology is essential to preserve the connectivity and adjacency relationships quickly and simple during the generalization process (Hardy, Hayles et al. (2003)).

2 Design of the infrastructure: the cross section

The second requirement is a design that enables a detailed representation of driving behaviour. Assuming a situation without any other traffic, driving behaviour is tightly related to the structure and geometry of roadways. This requires a solid representation of curves, turns and connections between road and lane segments (Wilkie, Sewall et al. (2015)). In addition, the traffic simulation model needs to reflect the relevant responses of drivers and vehicles to the signs along, above and on the road, as well as the impact of traffic systems such as traffic lights and dynamic route-information panels. While most micro-simulation models do not take account of the impact of the environment on driving (see for example Figure 7.2), this could be desirable for future applications. There should at least be the opportunity to add the objects that describe the environment to the modelled infrastructure.

Finally, the micro-simulation model requires the representation of both longitudinal and lateral interactions between vehicles.

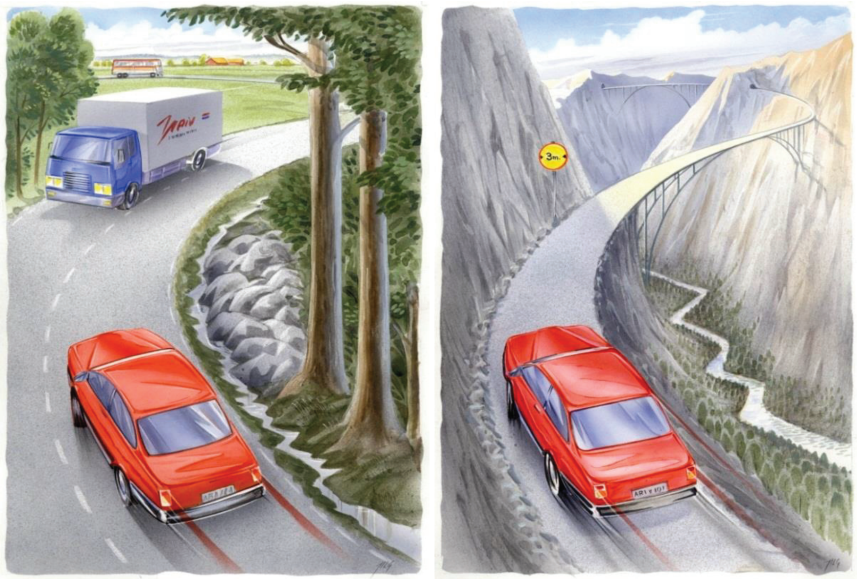


Figure 7.2: driving and the environment (source: https://www.slideshare.net/eab-themadagen/meer-veiligheid-meer-met-minder-maart-2011)

The distinction between the *link* and *cross section* will be further elaborated in the subsequent sections. As will be shown, the approach in this chapter resembles the proposal for the GIS representation of the road network at LoD2 (carriageways) and LoD3 (lanes) in section 6.6.

7.3.1 The basic topology for routing: a network with links and nodes

Route planning at a strategical level can be enabled by defining a network (**OTSNetwork**) as a collection of *link* objects (*edges*) that are directed and connected through *node* objects. In *OpenTrafficSim*, the **OTSLink** is the core link object. Its direction is defined by two nodes (**OTSNode**): a startNode *(A)* and an endNode (B). The geometry of the **OTSLink** is defined by the **OTSLine3D** and allows several implementations. The simplest is a **Straight** line that can be extended to a **Polyline** allowing the use of vertices. The **Arc** and **Bezier** options allow the implementation of curves. The **OTSLine3D** also allows the representation of the vertical slopes of a road**.**

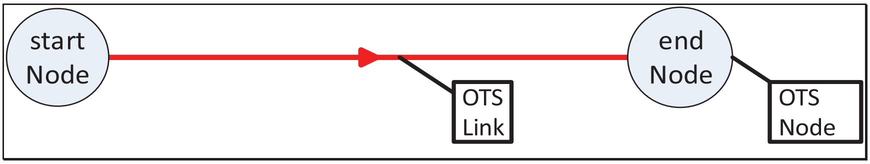


Figure 7.3: Link enclosed by nodes

For many mesoscopic and macroscopic models, the link and node objects are the basic elements for a transportation network. In addition, this requires link-attributes such as the driving direction, legal speed limits and allowed vehicle types. But for microsimulation approaches more detailed information about the road topology is required to mimic traffic and vehicle operations.

7.3.2 Requirements for microsimulation

This section introduces some of the main requirements for the design of the roadway infrastructure for a traffic microsimulation.

Flow direction: lateral and longitudinal
A microsimulation requires information about the number of lanes and their width. Moreover, vehicles in a simulation need to know whether it is permitted to switch lanes. A functional representation of lane markings at a road is therefore a compulsory requirement. In addition, the driving direction of a lane is an essential part of the description. For bi-directional roads without a physical barrier between the opposing lanes, a relevant aspect is whether traffic may temporarily drive on the opposite driving direction. This typically accounts for single carriageways that allow for overtaking, or roads with overtaking lanes where traffic is allowed in two directions. Other examples are a one-way road where bicycles are allowed to drive in the opposite direction too, or a bidirectional lane.

Modes of transport
In addition to the movements, the classification of the lane describes the modes of transport and vehicle types that are permitted to enter the lane. A lane can have mixed functions, for instance when a tramway shares the same infrastructure as the road traffic.

Dynamic road configurations
In addition to the fixed situations, *dynamic* road configurations, such as reversible lanes where the direction of a specific lane depends on the time of day or speed limits that vary by time of day, need to be part of the design too. Other examples that only incidentally occur, are contraflow lane reversal as a measure to support emergency evacuations, or temporary changes in the road layout as a part of road maintenance or reconstruction activities.

7.3.3 The CrossSection and its CrossSectionElements

Within *OpenTrafficSim* the **CrossSectionLink** extends the **OTSLink**, and captures the elements that are required for a traffic microsimulation model. The **CrossSectionLink** is composed of **CrossSectionElement** objects that describe the cross section of a link. These elements embed the typical elements of a street, such as traffic lanes, bicycle lanes, a hard shoulder, and a berm along the roadway. The *Sidewalk* is included too (but is not yet included in *OpenTrafficSim)* and allows the modelling of adjacency relations. This simplifies programming of interactions between vehicles and pedestrians walking or standing on the side of the road and, if allowed, permits pedestrians to cross the road (Willemsen, Kearney et al. (2006)).

Every **CrossSectionElement** has attributes to describe its geometry and functionality (or semantics). Figure 7.4 visualizes a simple one-way road with a *cross section* consisting of two lanes distinguished by a broken stripe and surrounded by edge-markings. In this configuration a description of these objects by their width and their lateral offset from the red-coloured heart-line is sufficient to represent the geometry.

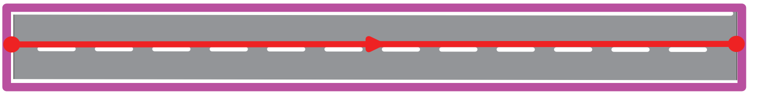


Figure 7.4: Link with a simple cross section

The ability to create a detailed geometry is provided by the **CrossSectionSlice**. At all points along a **CrossSectionElement** where either its lateral distance from the *heart line* of the parent **CrossSectionLink**, or its width changes, it provides an *offset* and *width* (as indicated by the blue arrows in Figure 7.5 indicating the offset of the lane marking along the lane). The **CrossSectionSlice** enables a detailed representation of the geometry of a **CrossSectionElement**, and represents the road and its lanes as a ribbon-like structure.

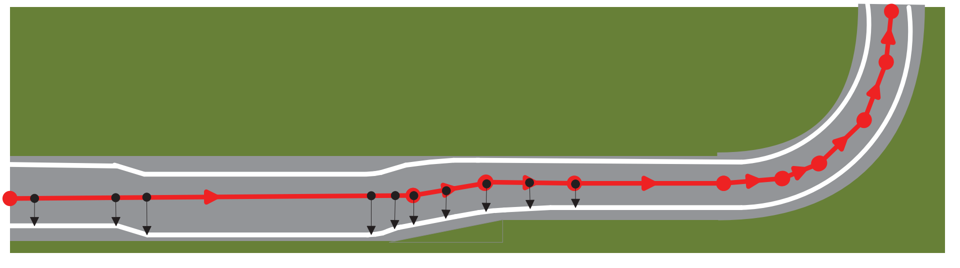


Figure 7.5: detailed geometry of the lane marker along, by cross section slices

The **CrossSectionElement** has various extensions to provide specific details about its function. For driving, the **Lane** class is essential in microsimulation models and provides information about the allowed driving direction and the vehicle types that are allowed to drive on the lane. A diagram that provides a summarized view of the structure of the Link object with its main attributes, is shown in Figure 7.6. This figure describes the Link as the central object, and shows its relationships with other objects (or *classes*) in an UML diagram. Every *Class* has a name (upper part), attributes (middle), and methods (lower). The relations between these classes are defined by different types of relationships. The figure shows an *aggregation* from the **CrossSectionElement** to the **CrossSectionLink**: every **CrossSectionElement** belongs to only one link, while a link can have 0 or more (0..*) **CrossSectionElements**. The **CrossSectionElement** is a basic blueprint for objects such as lanes and road markers. Such a relationship is called *inheritance*: the **Lane** object inherits the attributes and methods from the **CrossSectionElement**, and adds new attributes and/or methods. Every **Lane** has a **LaneType**.

The *association* between the **LaneType** and the **GTUType** encompasses a logical relationship or connection: in this case it shows which GTU types are allowed at a specific **LaneType**.

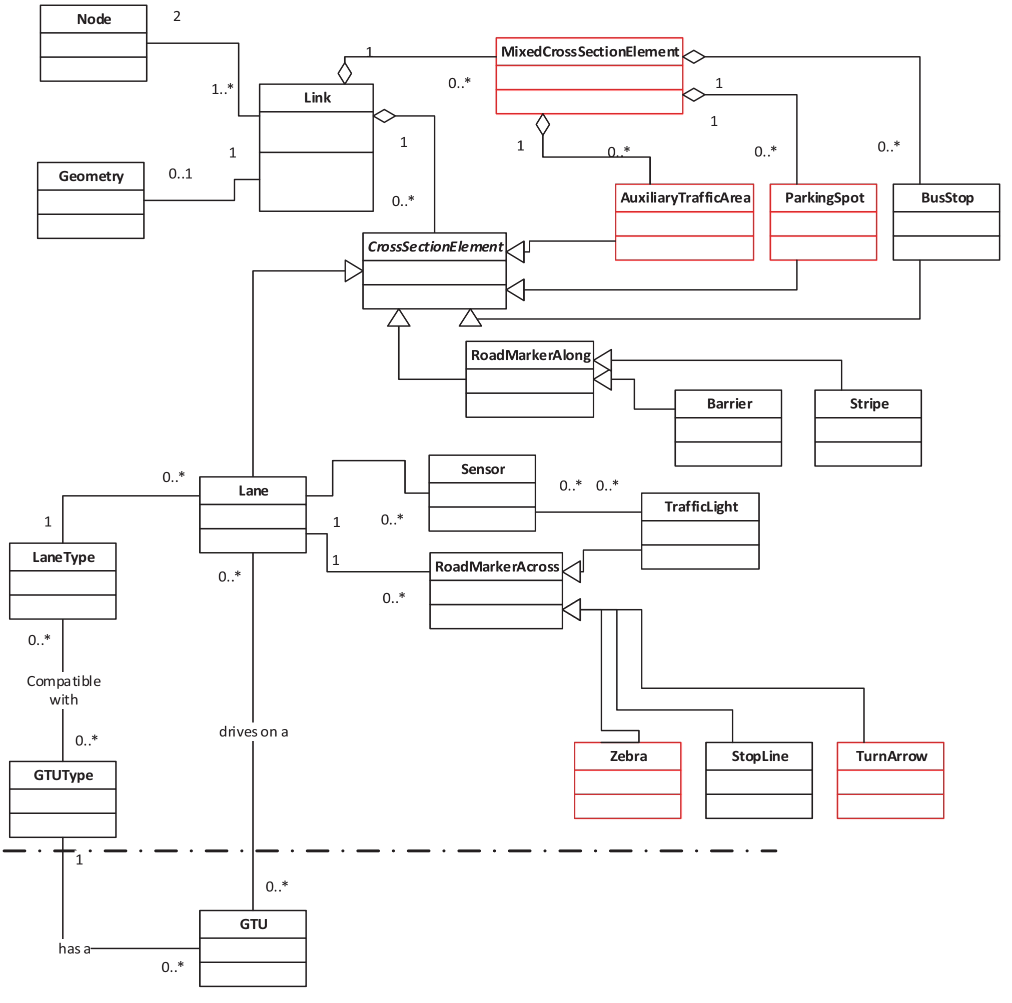
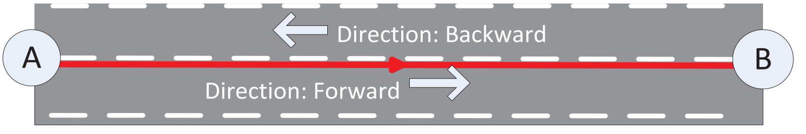


Figure 7.6: Pseudo XML schema of the Link design[14]

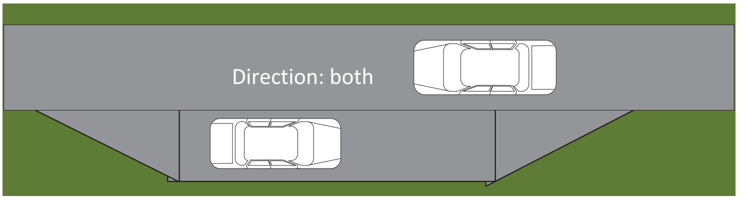
[14] Based on the discussions with the *OpenTrafficSim* team (red objects not implemented)

The **LongitudinalDirection** defines the direction of the separate *lanes* by **GTUType** and can be either "**Forward**", "**Backward**" *or* "**Both**", and relates to the direction of the *link*, defined by its "**startNode**" and "**endNode**".

> *The association of the GTUType and the LaneType allows the modelling of relationships between vehicle types and lanes of a road.*
>
> *Impact on: R-InfraDesign.1 and 2*



a) separate directions



b) bi-directional lane with passing

Figure 7.7: directionality of lanes

The abstract class **RoadMarkerAlong** provides information about the separation between lanes and other elements of a **CrossSectionLink**. These are used to derive lane changing rules and opportunities, as is concretized by the classes **Stripe** and **Barrier**. In addition to these objects, driving lanes can be surrounded by other **CrossSectionElements** such as a **ShoulderLane**. The combination of the geometry of the **CrossSectionElements** and the semantics from the **Stripes** and **Barriers** defines the ribbon relations: side-by-side, end-to-end, merging and splitting (Laurini (2012)).

> *The detailed road geometry is defined by CrossSectionElements that describe the lane and the road marking. The vertical slope is defined at the Link level.*
>
> *Impact on: R-InfraDesign.5*

Objects on or along the road are defined as classes that extend the **AbstractLaneBasedObject**. Examples of objects that are implemented in *OpenTrafficSim* are **Sensor**, **TrafficLight** and **BusStop**. All of them are positioned at a specific longitudinal position at a **Lane** and have a specific direction. The **BusStop** itself has information of the lines that stop at this location.

While the function of a **Lane** or **Barrier** is homogeneous in longitudinal direction, other elements such as parking spots and green areas often alternate with each other. Instead of moving/driving, the primary function is to park a vehicle, and either start an activity, change to another mode of traffic, or board and/or alight passengers. The objects *parking spot* and *green areas* are not yet implemented in *OpenTrafficSim*. We propose the abstract class *MixedCrossSectionElement* to allow such alternating functions. It extends the **CrossSectionElement** with a longitudinal start and end position along the **CrossSectionLink**. In this way a series of objects such as *GreenArea-ParkingSpot-ParkingSpot-GreenArea-BusStop* could be added as a cross section and allow for discontinuous configurations. The geometry of a *ParkingSpot* reflects a rectangle which is defined by its entrance-width, its length, and an offset angle. The *OffsetAngle* defines the entrance angle from a road to park a vehicle.

Figure 7.8 shows a one-way road with two lanes, a bicycle lane and finally some scattered parking spots and green areas along the road.

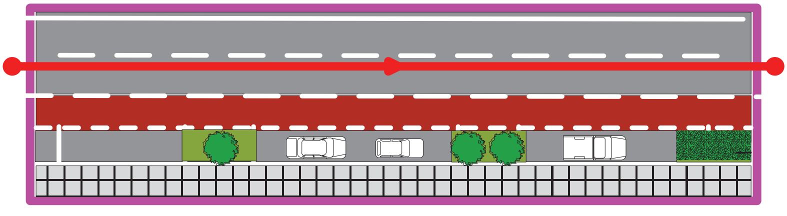


Figure 7.8: Street with a complex CrossSection

7.3.4 Mixed modes

Many roads allow mixed modes of traffic, such as vehicles in combination with two-wheelers (bicycles and mopeds). Figure 7.9 for instance, shows a driving lane where part of the lane is reserved for bicycles. Cars should use this cycle lane only, when there are no bicycles around.

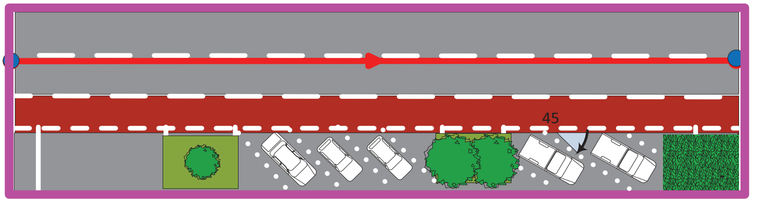


Figure 7.9: geometry of parking spots in a MixedCrossSectionElement

Another typical configuration regards the combined use of the road by tramways and other users of the road. The reason to integrally model these modes, is the impact of the combined use of the road on traffic flow capacity. The **Lane** attribute **GTUType** lists the allowed modes on a specific lane of the road. This could for instance be a combination car, truck, bicycle, and tram. The lay out of the **CrossSectionElement** then requires the addition of a *TramWay* (not yet implemented in *OpenTrafficSim*), describing the geometry of the rail track (as in Figure 7.10).

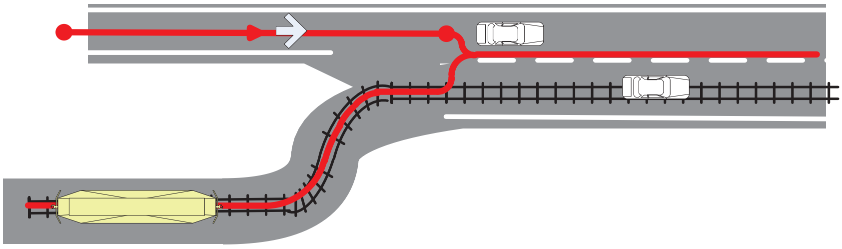


Figure 7.10: tramway at road

7.3.5 Lane changing rules

With respect to the infrastructure, the markers along a road provide information about the opportunity to change between adjacent lanes. The **Stripe** class provides information about the accessibility to parallel lanes. The attribute **StripeType** defines the actual lane changing rules, for instance by the value "**dashed**" (lane change from both lanes) or "**leftonly**" (only from the right to the left lane). Other objects such as barriers or kerbs, generate the same kind of behaviour (i.e. no lane change), but add a physical aspect that makes it difficult or impossible to overrule the intended behaviour. Ultimately, the transport or traffic model defines the behavioural rules with respect to the information.

> *The detailed road geometry is defined by CrossSectionElements that describe the lane and the road marking. The vertical slope is defined at the Link level.*
>
> *Impact on: R-InfraDesign.5*

In addition, the physical attributes that determine lane changing rules, the simulation requires the identification of nearby vehicles and objects on the road. There is a collection of classes, such as the **LaneBasedObjectIterable** class that, as the name indicates, identifies the downstream lane based objects. The same kind of methods, such as the **DownstreamNeighborsIterable** determine the neighbouring vehicles.

7.3.6 Dynamic road configurations

There exist various forms of time-dynamic road configurations. A first example is a dynamic turning lane at a junction, where the turning direction can be switched between a left and right turn by time of day. This is regulated by adding time dependent closures of the lane that connects the entrance lane with the egress lane. A list with timeslots by day indicates when a lane is closed. A reversible lane, as in Figure 7.11, can be modelled accordingly. In this case, the reversible lane is defined as bi-directional and has entrance and exit lanes at both ends of the lane. Again, time dependent closures define which direction is accessible. Within

> *The provision of giving the location of nearby objects is provided by various methods.*
>
> *Impact on: R-Topology.2*

OpenTrafficSim these aspects are not yet implemented. An example where time dependent behaviour is implemented is the **SpeedSign** with a startTimeOfDay and endTimeOfDay.

Figure 7.11: closed entrance of a reversible lane system (source: Google maps)

7.3.7 Traffic Signs

Besides the lane configuration, the impact of driving behaviour is strongly influenced by additional road side information and legal rules. Traffic signs encompass a broad collection of objects that impact driving behaviour. This section provides an exemplary description of the main types of traffic signs and rules, and how they impact on driving speed.

The actual free driving speed is largely based on the driver´s behaviour which is impacted by 1) legal speed limits, 2) priority rules and 3) the geometry of the road in terms of curvature, elevation and speed calming infrastructure[15]. The physical characteristics of the infrastructure, as well as priority rules at junctions, are already discussed as parts of the *CrossSectionLink* object. Figure 7.12 shows an overview of various traffic signs and infrastructural characteristics, and their expected impact on speed behaviour. The class **SpeedSign** that extends the **AbstractLaneBasedObject** defines the road sign with speed limits and contains the attribute Speed, the type of vehicle (**GTUType**) and specific *time slots*. In addition, other signs such as the road class with a maximum legal speed, or name place indicating the start of the built-up area, provide implicit speed information. These are not yet implemented in *OpenTrafficSim* (version 1.01). The same accounts for traffic signs that give notice of the end of a certain regime. This can be implemented by an *EndOfSign* class that contains a list of traffic sign types. An example is the type *EndOfLEgalSpeedLimit* that resets the value of the legal speed limit at the specific longitudinal position of a lane to the maximum legal speed of a specific road type. At the end, the information from these objects is transferred to the class *SpeedLimitSlices* that can be added to the **Lane** object. The class describes the speed characteristics from traffic signs, but also infrastructural characteristics such as curvature or speed bumps. A list records for every point along the lane to what level the speed limit changes. From this list, the leading speed limit can be selected by (often country specific) rules.

[15] The information in this section is largely based on the analysis by Wouter Schakel. See: http://simulation.tudelft.nl:8086/display/OP/Speed+limits

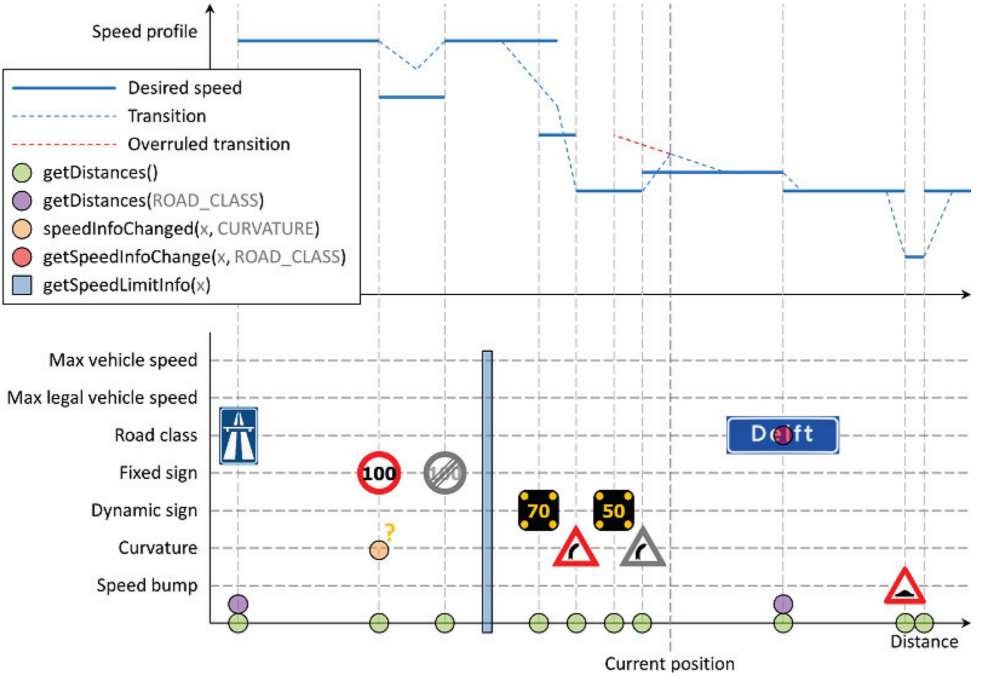


Figure 7.12: Speed profiles in relation to infrastructure (made by W. Schakel [15])

> *The implementation of LaneBasedObjects allows the definition of lane related objects that have an impact on driving behaviour. These are only partially implemented in OpenTrafficSim, and also not further elaborated in this chapter (but for speed limits).*
>
> *Impact on: R-InfraDesign.6*

7.3.8 Junctions

Lane markings at junctions often do not offer sufficient information to automatically derive the geometry of connections between entering and exiting lanes (see for instance Figure 7.13). The modelling of these lanes therefore requires an interpretation of the turning movements based on partial lane information and, when necessary, observation.

Figure 7.13: junction area with partial lane markings (source: Google)

The definition of an object that defines the boundaries of the junction area is not implemented in *OpenTrafficSim* (v1.01). We propose the class *JunctionArea* (see Figure 7.14) as a polygon that marks the interior of a junction.

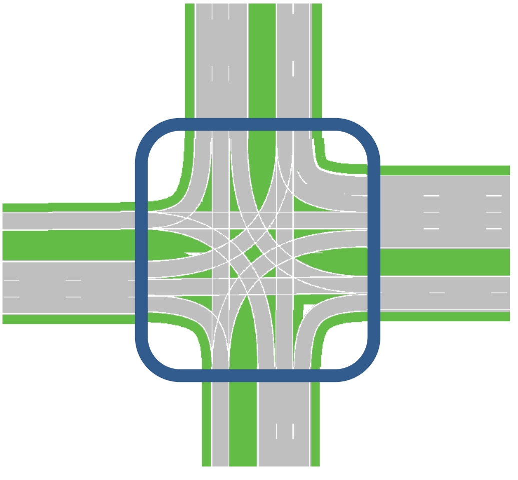


Figure 7.14: the JunctionArea describing the (virtual) lanes enclosed by the blue rectangle (based on work from W. Schakel)

While the junction object is not implemented in *OpenTrafficSim* (v1.01), the connections between entering and exiting lanes are already modelled by the link objects (**CrossSectionLink**). Moreover, *OpenTrafficSim* already generates the lane priority conflicts, as can be seen from Figure 7.15. There are two types, namely merge conflicts (the black rectangle) and cross conflicts (the dotted rectangle). The green (priority) and orange (give-way) lines represent the priority rules at these conflict areas.

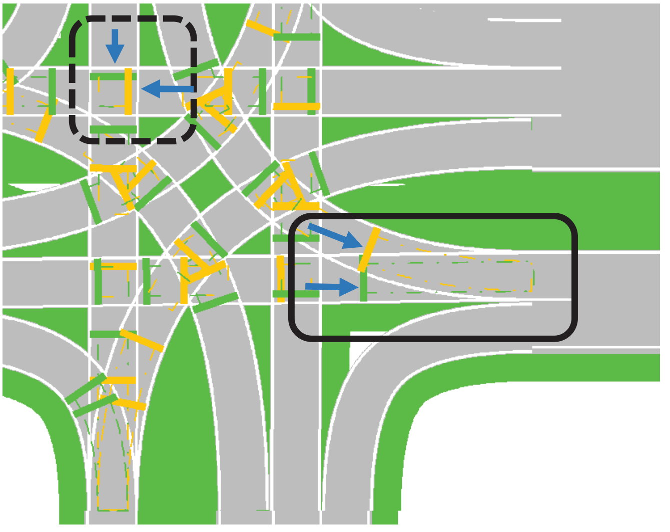


Figure 7.15: priority at conflicting lanes: merging and crossing paths (based on work from W. Schakel)

The design provides a detailed description of the junction topology
Impact on: R-Topology.3

7.4 Functional systems

The functional systems of the transportation infrastructure combine features of transport and traffic models, to simulate logical infrastructural systems such as signalized junctions, car parks or bridges. They contain references to entities from the road network and devices, such as a sensor or traffic light. The functional elements allow traffic flow models to define dedicated systems, and refer to all relevant units from the transport infrastructure. For road based traffic, the traffic light controller is the most common operational system within transport and traffic models. Other examples are a dynamic route information panel (using travel time measurements), on-ramp metering (using vehicle counts), a bridge (impact of openings), or a car park (with its characteristics such as parking barriers and tariff). These are not yet implemented in *OpenTrafficSim* (v1.01).

7.4.1 Traffic light controller

The main controlling system for road based traffic is the traffic light controller. In essence, it distributes the available green time over the various turning lanes. A vehicle actuated controller uses the measurements from sensors and/or other devices, to determine traffic demand from the various arms of a junction, optimizes the distribution of green time over the turning lanes, and returns the colours of the traffic lights. As has already has been described in the previous chapter, the Intersection Topology Format provides a data standard to describe the topology of an intersection (Vreeswijk, Claassens et al. (2016)). This standard provides a description of both the topology as well as the connection between the controller and the relevant road elements such as the sensors, stop lines and lanes.

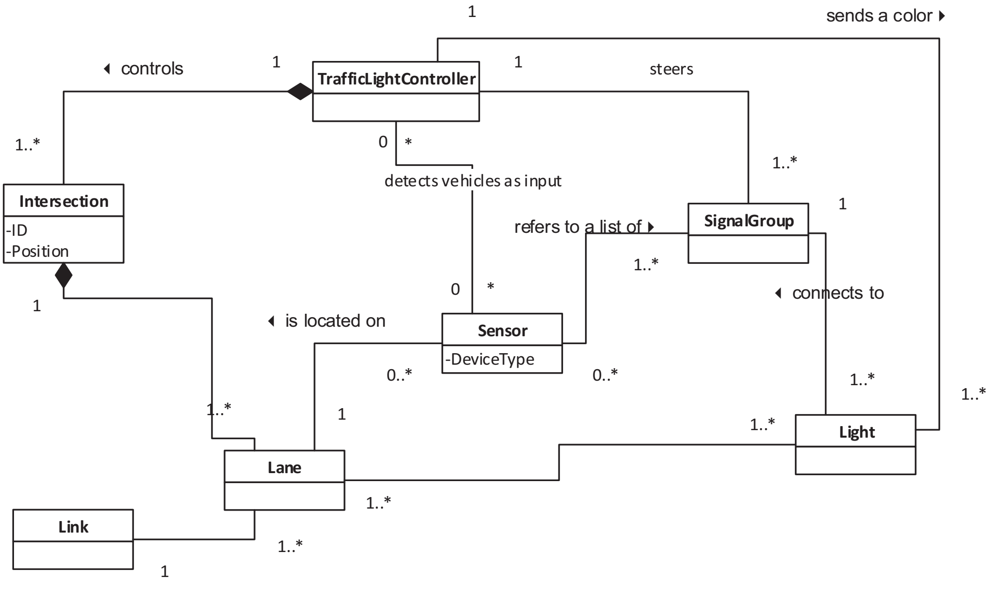


Figure 7.16: components diagram of a traffic light controller

The same concept can be used in the model. A traffic light controller has a unique ID and a descriptive name, and controls one or more intersections (Figure 7.16). These intersections describe all relevant lanes of the arms that enter and leave the junction, as well as the connecting lanes at the inner area. Every sensor has a location on a lane at a certain distance from the junction entrance, and sends vehicle pulses to the traffic light controller. A measurement by a specific sensor can relate to a list of lanes. If the detector is further away from a junction, it can be used to detect vehicles without knowing their intended turning direction at the junction. In that case the sensor relates to multiple (turning) lanes, as is shown in Figure 7.17 by the blue and yellow arrows. When the sensor is positioned just before the junction, for instance detector "D7-1", it relates to its own lane only. Based on the rules from the controlling program and the sensor measurements, the controller computes the state and changes of the traffic lights that it controls.

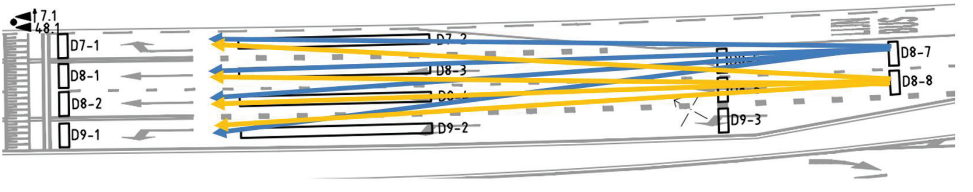


Figure 7.17: relation between sensor detection and driving lane (source: Vreeswijk, Claassens et al. (2016))

OpenTrafficSim already implements the traffic light controller. The **TrafficLightControllerFixedDuration** schedules the changes of the phases of the controller, and implements the **TrafficLightController** for defining the signal phases. The **SimpleTrafficLight** defines the signal heads by its lane and longitudinal position and returns the colour (**TrafficLightColor**). A more advanced demo presents an example with a vehicle actuated traffic light controller that uses the sensors to detect vehicles and based upon the detections and the control algorithm, controls the signals.

> *The main objects to define a traffic light system as described in this section are already included in OpenTrafficSim. This approach can be used for other crossings (next section)*
>
> *Impact on: R-InfraDesign.7 and 3*

7.4.2 Bridges and Level Crossings

Other functional elements are bridges and level crossings where the road infrastructure crosses another modal infrastructure. Analogous to the traffic light controllers, the system controls vehicle passings, but in these cases the decision to activate the barrier and the red stopping light is based on the bridge opening regimes. These can either be based on time-tables or induced by approaching boats. Likewise, level crossings (tram, rail) can be implemented as functional elements.

7.4.3 Car Parks and parking spots

The modelling of parking is not implemented in *OpenTrafficSim* (v1.01). In section 7.3.3 the *ParkingSpot* is presented as an ingredient of the *MixedCrossSectionElement* to allow the modelling of on-street parking. In many urban areas, parking is regulated by tariff systems and parking permits. The accessibility rules require a class that defines the drivers/vehicles that have a permission to park in accordance with the parking regime. These permissions may either be linked to a vehicle or a driver. As has been elaborated by van Lint (van Lint, Schakel et al. (2016)) this requires additional objects that can be linked to a person who drives the vehicle or the vehicle entity itself. Section 7.2 provides an overview about the modelling of vehicles and persons.

The functional element *ParkingZone* defines the area with a specific parking regime. In case of car-parks, the class *ParkingLot* approach allows the modelling of a collection of internal 'roads' within the car park that provide access to the parking spots of a car park. In this way it defines this internal structure of a *ParkingLot*, and combines it with the access and egress system by means of a combination of crossing barriers and traffic lights.

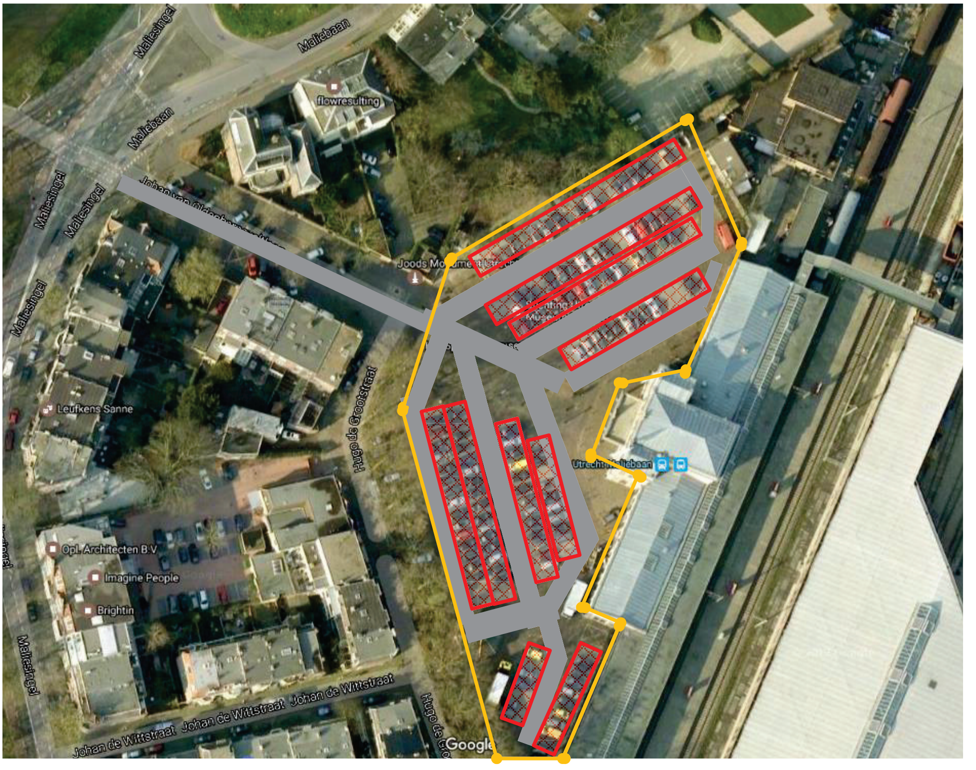


Figure 7.18: CarPark (ochre) with its entrance lane, and internal road structure and parking lots

> *The modelling of parking spots and car parks is not implemented in OpenTrafficSim. This section provides a global design.*
>
> *Impact on: R-InfraDesign.8*

7.4.4 Devices for measuring and control

Most (functional) traffic management systems are composed of network elements and devices. A device is a technical unit with various functions:

- Measurement
  - Inductive-loop traffic-detector
  - Speed camera
  - Laser radar sensor
  - Vehicle detection camera to measure travel times
  - Push button for pedestrians at a traffic light

- Control
    - Traffic light
    - Crossing barrier

The measurement devices are defined as extensions of the **AbstractSensor** object with a longitudinal position at a **Lane**. This allows a system to trigger the presence or passage of a vehicle. Sensors can provide additional information, such as the number of passages, the vehicle speed and the occupancy.

7.5 Modelling issues and the design of the transportation infrastructure

While the previous sections describe the design of the transport infrastructure, this section addresses specific transport and traffic modelling issues such as the connection between the road infrastructure and the activity locations, transfers with other infrastructures and, to start with, some design rules for the road infrastructure.

7.5.1 Representing links as bi-directional or separate one-way links?

The proposed design of a link at the highest level of detail contains the description of a road by means of **CrossSectionElements**. This enables a description of a road as one-way or bidirectional. This leaves a choice for representing roads: a dual carriageway for instance can either be represented by one bidirectional link or two one-way links.

As has been reported in section 6.6, we propose to define the driving directions (AB versus BA) on single carriageways as separate entities. Such a distinction in separate links per direction may appear appealing, because it instantly separates the road information by driving direction. However, for roads without a clear separation of driving lanes, such as overtaking lanes that can be driven in both directions and (usually narrow) roads without lane markings, the choice to define a single carriageways as *one* link would be advocated for traffic simulations. For a dual carriageway, a combined use is by definition impossible, as traffic in opposite direction is separated by a median (central reservation), leaving the choice open for defining them as one or two links. For these roads we propose to define them separately. This also complies with other data standards, such as the *GDF* data format and has also been suggested by Ruas, Gold et al. (2008), Beil (2017) and Hong, Jianping et al. (2008). This approach populates parallel lanes with lateral connectivity and the same driving direction into one object.

7.5.2 Segmentation of links

A further question regards the segmentation of roads. Partially based on rules used for navigation networks roads are split by a node in the following cases:

- Start and end of a road (dead-end);
- Change of road characteristics that are more or less fixed for a longer period, such as the lane configuration;
- Entry of a car park;
- Road crossing at grade;
- Crossing with a railway or bridge.

In case of changes by other attributes, such as a sign indicating a change of the legal speed limit, this will be indicated by a sign at a certain longitudinal position along a road or lane.

7.5.3 Transfers between multiple modes of traffic

The interaction with infrastructures of different modes of traffic has already been depicted for crossings, such as train and tram level crossings, and bridges. A second type of interaction regards the transfers from one mode to another, such as a parking place where people finish a car trip and change to other modes. The design of the road infrastructure should provide the means to model transfers between modes, and requires a connection between the infrastructures of the different modes. Figure 7.19 shows an example of such a transfer, with parking lots in between the driving lanes and the sidewalk for pedestrians. As the road object comprises both the pedestrian, parking and driving infra, their topological connection (touching) enables the modelling of relationships and transfers between these different modalities. Moreover, this simplifies programming of interactions between vehicles and pedestrians walking or standing on the side of the road and, if allowed, permits pedestrians to cross the road. The same reasoning accounts for bus stops where passengers egress, and continue their trip by walking to the next destination or mode.

Figure 7.19: intermodal transfer from road (blue) via parking lots (red) to pedestrian (ochre) infra (source Google).

The *PedestrianCrossing* is proposed as an extension of the **RoadMarkerAcross** and defines an object that overlaps multiple lanes. It can either be used for vehicle based simulations of a road, or as an opportunity for pedestrian models to connect the pedestrian infrastructure at both sides of a road. The opportunity of pedestrians to cross a street without using a crossing, partly depends on the road type (and the allowed vehicle types), and, additionally, the characteristics of the street. If, for instance, the road is surrounded by physical barriers, the opportunity and

This section describes the approach to create transfers between various modes of transport

Impact on: R-Topology.1

chance that pedestrians cross the street will be low, whereas more pedestrians will cross roads that are surrounded by pavements and are located in shopping areas. The actual occurrence of pedestrians crossing the street, thus is an outcome of characteristics of both the infrastructure and the build environment. A pedestrian model may use these data to predict pedestrian movements along and across the street.

7.5.4 Ferry for crossing waterways

The simulation of ferries as part of a vehicle based trip, is no common topic as appears from a literature survey. In this section we present an approach that discards the actual modelling of a ferry. Instead, the intermediate transfers are modelled by applying (1) a car park or road segment with the actual capacity of the ferry terminal, (2) a link that connects to the next car park (or link) and (3) a car park (or link) with a capacity that matches the maximum number of vehicles on a ferry. A traffic light controller at the exit of the first car park represents the entrance to the ferry. If it turns to green, cars are allowed drive to the entrance of the ferry, modelled by the second car park. At this stage, the exit of the second car park is blocked by a red traffic light. When all cars have entered the “ferry” the traffic light remains red until the time for crossing the waterway ferry has passed (a fixed time period). The cars leave the “ferry” and when all have left, the same process start for the opposite movement of the ferry.

Modelling intermediate transfers can be modelled with the proposed design.

Impact on: R-InfraDesign.4

7.5.5 Connecting trips with activity locations

As has been discussed, the parking location is the logical entity that represents the start or end of a car trip. Actually, the commercial microsimulation package Vissim already uses the parking lot for that purpose (Fellendorf and Vortisch (2010)), and links it to one of the traffic zones that cover the modelled area. For parking models this approach limits the choice for vehicles to park: the link between parking lot and a zone therefore appears less preferable. Ideally, the transport and traffic model should link the choice of the parking location with the next destination which can either be an activity location, or the starting point of the next trip leg (in case of multi modal trips). Yet, there are only few examples where parking (behaviour) is actually modelled. An example is *ParkAgent*, an agent based model that explicitly models parking choice and search in a city (Benenson, Martens et al. (2008), Martens, Benenson et al.

(2009)). It actually defines activity locations as destination, and models the search and choice for parking locations.

A development that may increase the relevancy for the modelling of parking behaviour, is the advent of the self-driving vehicle. When self-driving vehicles reduces car-ownership, and cars are able to self-park and drive to their next trip, the role of parking/halting will increase significantly.

Within the transport infrastructure the parking location is an intermediate destination for trips between activity locations. As Figure 7.20 shows, the car based trips start or finish at a *ParkingSpot*.

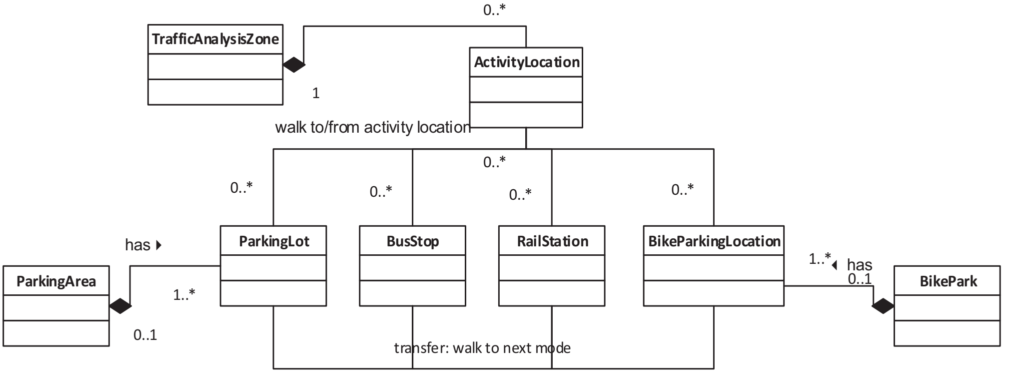


Figure 7.20: the Traffic Analysis Zone

> *This section describes the connectivity between the road network and activity locations*
>
> *Impact on: R-Topology.1*

7.6 Switching between levels of detail

While the design of the transport infrastructure is typically suited for microsimulation, many other approaches use coarser object descriptions. This section describes the general principles that hold, when scaling to a lower level of detail.

7.6.1 The road network

In the map making process, *generalization* is one of the main principles for switching to a lower level of detail. The scaling process distinguishes *model* and *cartographic* generalization (Ulugtekin, Dogru et al. (2004)). While the cartographic aspect is relevant for visualization of maps, the *model* generalization is of primary relevance for the functional aspects, in this case for transport and traffic models. This requires a description of the relations between different objects at one level – termed *relationships* – and the relations between the different representations of the same object at different levels – *connectivities*. To generate lower level of detail representations, an automated reasoning process is required. This means that the updates can be propagated from higher level representations by using the model generalization operators applied automatically in the modules to be generalized (Kilpelainen (1997)). The question is if and how the model generalization approach is also applicable for transport and traffic models.

To create representations of the network at coarser levels, we need a clear description of the changes of the model approach and its impact on the network representation. Basically, this requires an evaluation of the representation of the behaviours for model approaches. At a high level of detail, microsimulation models represent the following behaviours:

- Car following behaviour;
- Lane changing behaviour;
- Speed choice (under free driving circumstances);
- Route choice;
- Interferences of non-motorized traffic, specifically at junctions (Ben-Akiva, Gao et al.)).

Roads

In section 5.3 we have already shown that the typical behaviour of vehicles and their interactions in microsimulation models is not apparent in the meso- and macroscopic models. Thus, the detailed lane geometry of the microsimulation models is not included anymore. The lane changing and car following behaviour of individual vehicles is replaced by other methods to derive the road and lane capacity. This analytical derivation transforms the impact of discrete events into continuous formulae that use flow as an input. With only few parameters, the lane capacities are estimated from the traffic flows. These functions typically require additional link-attributes, such as the number of lanes, function of the road (link-type), capacity (i.e. maximum number of vehicles per time unit), maximum traffic density (vehicles per lane or link), and the legal (or estimated) maximum speed. In addition, section 5.3.3 presented specific approaches for speed-flow relationships at on-ramps, which required specific details such as the length of the changing area and the length of the on-ramp.

While not all objects and attributes can be directly generalized with *relationships* and *connectivities*, some attributes, such as the link length and its legal speed limits, can be directly derived from the link and lane attributes of the detailed simulation network. With respect to the speed limits, some microsimulation models use speed signs to *indicate* the location along the road where speed changes. Based on this information the average speed characteristics per link can be derived. For lane dependant attributes, the information needs to be averaged. Varying speed limits by lane for instance will often be averaged as the coarser models apply a link based speed limit. As mesoscopic and macroscopic models have their own data design, these kind of heuristics will be case specific.

OpenTrafficSim already has a design that allows a representation of the mesoscopic and macroscopic link representations. The **CapacityOTSLink** provides a basic template for such links, by discarding the cross section and instead adding the *capacity* as an attribute. A further extension this class allows all kinds of mesoscopic and macroscopic link representations. The case study presented in section 8.3 however, shows an example of using and extending the *OpenTrafficSim* framework, to determine the shortest path in a road network at a link level.

Junctions

With respect to the description of junctions, microsimulation models again provide a detailed description of the junction lay-out, to model the behaviour of vehicles choosing their turning lanes. Mesoscopic models again require less details. As the lane changing and car following behaviour is absent, the model only requires information about the number of lanes and their vehicle storing capacity. In addition, information about the allowed turns between the entering and exiting links is required (Florian, Mahut et al. (2008), Ben-Akiva, Gao et al.)). The information about junctions is already defined in our design (see section 7.3.8 and 7.4.1), and provides information about the type of junction, and the internal lane connections between access lanes and exiting lanes.

To model the mesoscopic and macroscopic approaches, the connections at a junction can be easily created within the *OpenTrafficSim* framework. Again, the **CapacityOTSLink** provides a basic template to do this.

When switching to lower levels of detail, a typical requirement is that network properties, such as the ability to generate valid graphs and routes, remain intact at the various levels of detail. This implies that after generalization, the spatial relations between objects must hold (Laurini (2012)).

> *The representation of the links and junctions at various levels of detail as proposed in this chapter allows an exchange of information between the various levels.*
>
> *Impact on: R-InfraDesign.9*

The *OpenTrafficSim* network design provides a structure that already contains the idea of a graph based representation of nodes and links, combined with a detailed description of the lanes, parking lots and other elements of the road. Section 8.5.1 presents a case study, based on an extension of *OpenTrafficSim*, where we create a graph from the links and nodes to find the shortest routes between origins and destinations.

7.6.2 Connecting the activity locations at various levels of detail

The zoning system in mesoscopic and macroscopic models is not standardized, but is usually based on the definition of Traffic Analysis Zones. These are areas that often correspond to administrative areas, that contain data about the population and activities. The connection of these traffic zones with the transportation network can be realized in the following ways:

- Define a centroid of the area, and connect it by means of feeding links;
- Define locations within the area (for instance parking lots) where traffic starts or ends a trip;
- Distribute the trip starts and ends over the links in an area.

In *OpenTrafficSim* the generation of trips is basically implemented by classes that allow the specification of links and lanes where vehicles are generated (such as the **GTUGeneratorIndividual**). In version 1.01 there is no object that defines which links are linked to specific activity locations. We propose a *TAZ* class with a polygon as its geometry that defines the activity location. This polygon should enable a connection with parking locations or (a collection of) road elements. As the size of these areas is flexible, it can be defined in line with prevailing areal classifications.

7.7 The design of the transport infrastructure in relation to data standards from GIS

The design of the transport infrastructure as described in the previous sections, provides a basis for usage in transport and traffic models. The counterpart of this design, the geospatial data standard, has been elaborated in chapter 6. This sections compares the data design of the road network with the proposal for the geospatial data standard.

7.7.1 Links

At the highest level of detail, the road is defined by a link that represents a carriageway and its cross section that describes the lanes, the lane markings along and the shoulders. Basically, lanes are represented by heart-lines and ribbon-like spaces around these heart lines. Further details of the road configuration such as speed bumps, detectors or lane markings can be added by providing the lane objects at a certain longitudinal position along a lane. This level of detail is generally used by microsimulation models such as Vissim, Aimsun and Paramics. This level of detail resembles the proposal for the geospatial data standard for road networks at LoD3, by extending the *CityGML* data standard as presented in section 6.6.

The design of the road infrastructure from this chapter also allows a coarser description by discarding the detailed cross section description. Instead, the lane based information is transferred towards the carriageway-level and requires additional *link* attributes. As has been described section 5.3 this can only partly be automated. For instance, the *capacity* of links, is a result of vehicle interactions within microsimulation models, and thus cannot be derived from the objects that describe the detailed road infrastructure. Therefore, other approaches are required to assert the capacity in macroscopic traffic flow models. While this representation fits within the *OpenTrafficSim*, it is not yet implemented in the current design. This level of detail resembles the LoD2 proposal for road networks in geospatial databases: a description of carriageways by driving direction.

7.7.2 Junctions

The connections between entering and exiting arms of a junction in *OpenTrafficSim* are defined as link objects with their associated cross section definition. Again, there is a strong resemblance with the proposed GIS design. The connectors in GIS at LoD3 connect the entering and exiting lanes in the same way as *OpenTrafficSim*. The same accounts for the coarser LoD2 level, where the entering and exiting carriageways in GIS are connected with connectors. Again, by discarding the detailed cross section description, the junction connections can be created in *OpenTrafficSim* in a similar way.

7.7.3 Evaluation

To apply this design in practice further elaborations are required, both from the side of data standards from GIS and the commercial transport and traffic software packages. A first essential step is the further elaboration of GIS data standards. From the previous chapters we have concluded that *CityGML* provides a good basis for modelling the transportation environment in transport and traffic models, as it (1) distinguishes various levels of details and (2) provides a description of both the transportation infrastructure and the activity locations (terrain and buildings). On the other hand, as has been mentioned in chapter 6.4, the description of the Transportation part in *CityGML* is yet insufficient and requires both an extended design of objects such as traffic lights and information panels, as well as a redesign of the levels of detail, as is proposed in section 6.6. In addition to the proposed design, a proper exchange of information between the geospatial databases and the transport and traffic models requires additional efforts. This regards for instance to the governance of the geospatial data collection. As there are many road authorities within a country, an appointment system that aligns the data collection and administration of the digital road map is an important factor.

7.8 Generating new datasets from current data-sources

In addition to the further elaboration of the CityGML object definitions, as described in the previous section, this section discusses the *opportunities to feed* future CityGML datasets for transportation with data from currently available sources. The essential question is whether and how these data can be used and transformed to meet the proposed design. This section explores the opportunities to combine open data sources that are available for the Netherlands, to create new datasets that align to the requirements of a combined description of roads by lines and spaces at LOD3.

Within the Netherlands, the "National Road File" (NWB) provides a digital description of the roads, at a comparable level of detail as the OpenStreetMap (OSM) network (see section 8.2). Specifically, for the highways, functional information about the number of lanes and legal speed limits is added. The geometry of the roads is represented by polylines. In addition to the NWB, all roads within the Netherlands are digitized as spaces within the "Registration Large Scale Topography" (BGT). The required quality of the BGT for road segments is defined in terms of geometrical accuracy (30 cm) and actuality (6 months). Combining these two data sources would be an option to create a network that combines line elements and spaces.

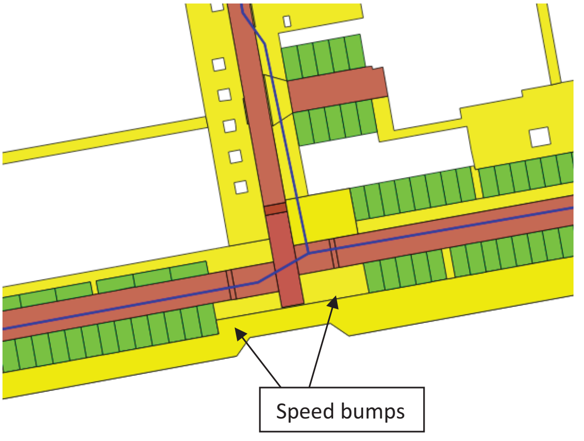


Figure 7.21: Example of the NWB roads (blue line) and the BGT spaces (roads (brown), pedestrian areas (yellow) and parking lots (green))

As Figure 7.21 illustrates, the BGT objects already represent road segments, pedestrian areas, parking lots and specific functions of road elements such as the speed bumps. The NWB road elements are indicated by the blue lines. As can be seen, the geometrical quality of NWB is not sufficient for all road segments: the road (blue line) partly coincides with the pedestrian area. Procedures to improve the quality would therefore be a first step. A second condition for a proper connection between area (BGT) and line (NWB) representations, is that one line feature coincides with one or more areas (De Boer and F. Penninga (2017)). Figure 7.22 however, shows the opposite. The selected road segment from BGT (space) encompasses multiple road segments (lines) from the NWB network. Again, procedures to create spaces that align to this condition would improve the usability for modelling transport and traffic. Figure 7.23 shows an example of such an elaboration.

As has been noted in section 7.3.8, the representation of junctions is an important aspect to consider. This accounts both for the lines and spaces. Chapter 7 further elaborates on this aspect, by exploring the possibility to convert an OpenStreetMap network into a detailed micro-simulation network (8.2).

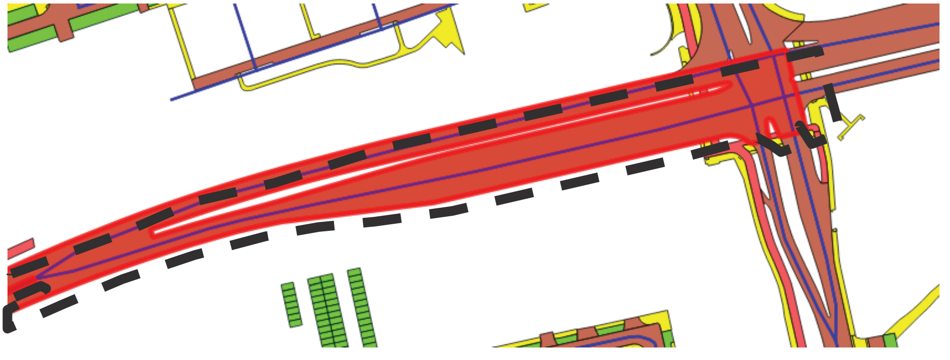


Figure 7.22: selected road segment from BGT (surrounded by black dotted line) overlapping multiple NWB road features (blue lines)

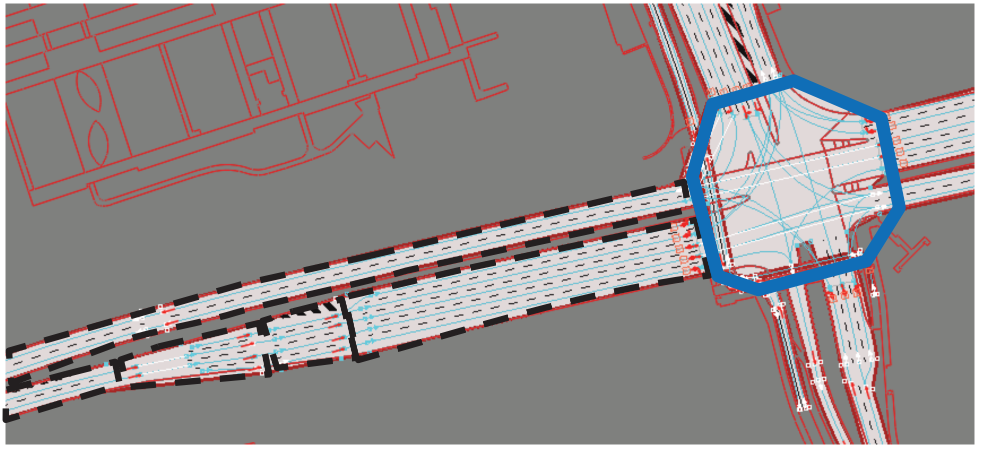


Figure 7.23: Partitioning the road area (see Figure 7.22) into separate spaces that better align with the road network characteristics.

7.9 Evaluation of requirements for objects of the transport and traffic models

- **R-Topology.1**
  The ability to create routes between activity points, along roads and lanes, including multi-modal trip legs and transfers between modes
  The basic design links and nodes with their cross sections has already been implemented in *OpenTrafficSim*. The simulations utilize shortest path algorithms to create routes between origins and destinations. There is yet no implementation of multi-modal trips with transfers between modes. While *OpenTrafficSim* already simulates the behaviour of busses at bus stops, there is no pedestrian simulation. The design however, includes the possibility to add a sidewalk as a cross section element next to a bus stop. Such a side-to-side relation allows an identification of connectivity between both elements, and thus enables a connection (i.e. by means of walking to other halting places or parking areas).

- **R-Topology.2**
  Provide occupancy information by giving the locations of nearby objects on the road infrastructure, and allow for geometric computations such as calculating the distance between cars
  The simulation environment of *OpenTrafficSim* enables the identification of nearby objects on the road. There is a collection of classes, such as the **DownstreamNeighborsIterable** class that, as the name indicates, identifies the downstream vehicles and its positions.

- **R-Topology.3**
  The design provides topological information of junctions, with internal lanes that connect the entrances and exits of a junction.
  These connections at junctions are defined by the **CrossSectionLinks**. The examples in this chapter already show visual examples of the junction topology and topography.

Further requirements to provide detailed information for vehicle movements:

- **R-InfraDesign.1**
  Ability to distinguish various vehicle types, and relate this vehicle type to the attributes of roads and lanes, such as speed limit and accessibility
  OpenTrafficSim uses two classes to connect specific vehicle types to a lane. The **Lane** has a **LaneType** that defines the allowed **VehicleTypes** and the associated driving direction. The same approach connects a **VehicleType** to lane objects such as a **SpeedSign,** and allows to define specific timeslots.

- **R-InfraDesign.2**
  Allow an overlap of mixed use of the infrastructure such as driving lanes and a rail track for trams
  Again, the linkage of the **VehicleType** with the **LaneType** allows a definition of mixed use of specific lanes.

- **R-InfraDesign.3**
  Implement the infrastructure of other modes, such as level crossings, when it interferes with road traffic
  OpenTrafficSim distinguishes various projects, among which the still empty projects *ots-water* and *ots-rail*. Basically, these could be developed and used to implement waterways and rail as separate networks. The interaction could then be modelled where the networks cross each other. In the current *OpenTrafficSim* framework, level crossings can already be defined by traffic light controllers, where a railway link is defined as an artificial road element that crosses a normal road. Another solution is the implementation of a traffic light controller where the closure of the gates is simulated by a traffic light that is turning to red in line according to the scheme of the train passages.

- **R-InfraDesign.4**
  Provide intermediate transfers that are part of a road based trip, such as transfers over waterways.
  There are various ways to simulate a ferry. Section 7.5.4 presents an example that can be implemented by using the objects from the proposed design.

- **R-InfraDesign.5**
  Represent the geometrical design of a road as characterized by the horizontal curves and vertical slopes, and the cross section of the road, including the road marking (stripes) to distinguish lanes and the shoulder

As has been reported, the **Lane** object distinguishes its adjacency relations with other lanes by the **rightNeighbours** and the **leftNeighbours**. The **Stripe** object and the **StripeType** support the definition of adjacency rules by **VehicleType**. The vertical slope can be defined by the Z-coordinates of the start and end nodes of a link object.

- **R-InfraDesign.6**
  Define road and lane related objects that have an impact on behaviour driving
  The AbstractLaneObject is the parent of all types of objects that can be defined as road and lane related objects. Section 7.3.3 presents some exemplary classes such as a sensor.

- **R-InfraDesign.7**
  Allow for the inclusion of traffic management and monitoring systems when they have an impact on traffic behaviour.
  Section 7.4 describes how traffic management and monitoring systems can be defined or implemented in *OpenTrafficSim,* and presents an example of a simple traffic light controller.

- **R-InfraDesign.8**
  Include parking spots or parking areas for multimodal trips and the relation with activity locations
  As is reported in section 7.4.3 the parking spot and parking zone are objects that are part of the infrastructure. In addition, section 7.5.5 describes the proposed connection of parking spots with the activity locations,

For the various levels of detail:

- **R-InfraDesign.9**
  There should be a correspondence between the links and junctions at various levels of detail, in order to exchange information between the levels of detail.
  Section 7.6.1 shows the design of links and junctions, and elaborates on the possibilities to exchange information between the levels of detail. The design allows the extension of the link and node objects, and enables their use for both microscopic, mesoscopic and macroscopic models.

7.10 Overview and conclusions

This chapter presents a design of the model objects that enable the composition of the road infrastructure. This design meets the requirements from chapter 5 and 6, as shown in the previous section. The basic objects are the *Network* that uses *Links* and *Nodes* to represent the connectivity between roads and junctions as a graph. For further details, the associated *CrossSectionElements* describe the configuration of a *Link*, and includes both the area where traffic moves or parks, and the parts that describe the further characteristics of the road such as a parking spot, barrier and berm. This integral approach provides the basic opportunities to represent the movements of vehicles and the exchange with other modes of traffic at transfer locations.

In addition to the link segments, we present an approach to define junctions that aligns with the proposal for the geospatial data standard. A junction is defined by a polygon and represents all internal 'links' that connect the incoming and egress lanes, and thus explicitly defines the allowed turning movements. In addition, functional elements such as traffic light controllers and car parks are connected with the road network. These functional units use devices, such as sensors or traffic lights, to inform, measure and control traffic.

The combination of a graph by links and nodes, with the details of the road configuration, presents a dual system that resembles both the requirements for microsimulation (detailed lane description) and the meso- and macroscopic (link-based graph) network representations. In order to connect trips with the network, the traffic zone is defined as an area that encompasses a collection of activity locations. At the highest level of detail, every activity location can be connected with the 'boarding' location for the first mode (car park, bus-stop, etc.) by means of a walk link. By means of generalization procedures, this design of the transport infrastructure allows a representation of the road network at various levels of detail. As already has been presented in the previous chapters, not all attributes can directly be derived from the highest level of detail. for these attributes (such as the capacity), additional procedures are required.

Table 7.1 summarizes the requirements that have been evaluated in section 7.9. As the table shows, the design meets most of the stated requirements, but for practical use requires further elaborations to include all objects an attributes.

Table 7.1: Overview of the requirements for objects of the transport and traffic models

Requirements	Yes-no	Comments
Ability to create routes between activity points, along roads and lanes, including multi-modal trip legs and transfers between modes	☑	
Provide occupancy information by giving the locations of nearby objects on the road infrastructure, and allow for geometric computations	☑	
Provide topological information of junctions, with internal lanes that connect their entrances and exits	☑	
Ability to distinguish various vehicle types, and relate this vehicle type to the attributes of roads and lanes	☑	
Allow an overlap of mixed use of the infrastructure such as driving lanes and a rail track for trams	☑	
Implement the infrastructure of other modes, such as level crossings, when it interferes with road traffic	☑	*Only road related infra is modelled*
Provide intermediate transfers that are part of a road based trip, such as transfers over waterways.	☑	*No explicit representation of ferry movement*
Represent the geometrical design of a road as characterized by the horizontal curves and vertical slopes, and the cross section of the road, including the road marking (stripes) to distinguish lanes and the shoulder	☑	
Define road and lane related objects that have an impact on behaviour driving	☑	*Not all objects defined, only exemplary*
Allow for the inclusion of traffic management and monitoring systems when they have an impact on traffic behaviour.	☑	*Elaborated for traffic light controller*
Include parking spots or parking areas for multimodal trips and the relation with activity locations	☑	
A correspondence between the links and junctions at various levels of detail, in order to exchange information between the levels of detail	☑	
Data objects of the model align to their counterpart in proposed (extended) CityGML standard	☑	

This design of the road infrastructure is not unique. Other traffic simulation model packages are able to represent complex infrastructures as well. Table 7.1 provides an overview of some complex infrastructural entities and shows if they can be modelled by two commercial packages (Vissim and Paramics Discovery) and the proposed design of *OpenTrafficSim* (including the unimplemented class proposals). It should be noted that *OpenTrafficSim* enables the modelling of most of these entities, but does not always enable the actual simulation of those entities. For instance, the joint simulation of cars and bicycles on a shared lane is not yet implemented. As the table shows, most of the infrastructural road and junction configurations can be represented by all packages. Both Vissim and *OpenTrafficSim* allow the representation of parking and the interaction with pedestrian infrastructure.

The main differences regard two other aspects. The *first* aspect that is only covered by the *OpenTrafficSim* design, is the distinction between Persons and Vehicles. This allows the modelling of agent based simulation, where behaviour depends on the combination of the driver and car characteristics. The *second* aspect concerns the alignment of the model object design with the geospatial data standards. As has been shown, the design of *OpenTrafficSim* aligns with the LoD2 and 3 representation. Both Vissim and Paramics provide another approach. As we will see at the next pages, the connections between the entrance and exit lanes are modelled differently.

Table 7.2: Possibilities to model the infrastructure of specific entities by some exemplary simulation packages

Aspects:	Vissim	Paramics-Discovery	OpenTrafficSim + additions
Infrastructural entities:			
• Multilane motorway with lane specific dynamic speed limit and overtaking prohibition for trucks	☑	☑	☑
• Multilane motorways with entries and exits including acceleration and deceleration lanes and/or weaving sections	☑	☑	☑
• Signalized intersection for various modes; green phases according to movement and vehicle type	☑	☑	☑
• Signalized intersection where bicycles are free to turn right	☑	☒	☑
• Intersection with turning prohibition (for example no right turn for cars but allowed for busses or bicycles)	☑	☑*	☑
• Signalized intersection with turning lane where turn movement varies by vehicle type (i.e. right turn, but busses go straight)	☑	☑	☑
• Car parks / parking lots	☑	☒	☑
• Exchange with pedestrian infrastructure	☑	☒	☑
Others:			
• Aligns with proposal GIS data standard	☒	☒	☑
• Allows agent-based modelling: distinguish persons and vehicles	☒	☒	☑

**Only if there is a separate right turning lane*

In Vissim the connections between two crossing links are defined by means of the attribute *isConn*. As Figure 7.24 shows, the (yellow) link completely runs through the junction. The connections at the junction are constructed by connectors (red). There is no resemblance with our proposed approach for junctions, where the approaching links stop before the junction area, and connectors represent the allowed movements between entering and exiting lanes.

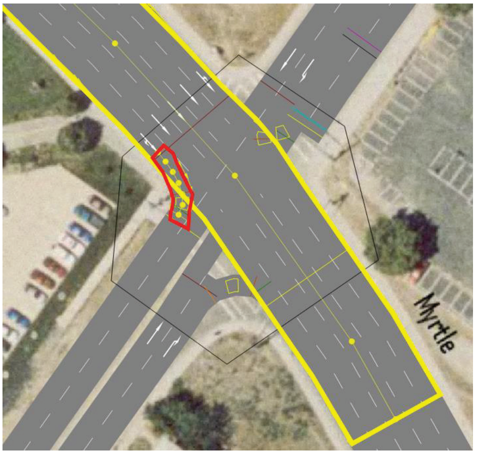


Figure 7.24: Vissim: link (yellow) and connector (red: right turning movement) to another link

Paramics applies a design with links and nodes, as is shown in Figure 7.25. The links towards the junction all join at one node. The connections between the entrance and exit of the junction are defined by *Next Lanes*. There is no possibility to define the vehicle types that are allowed to use the *Next Lanes*. Paramics has no readable network files and only a limited export to the *esri* shape format (GIS), which does not contain the next lane information. There is no possibility to exchange this information with other platforms.

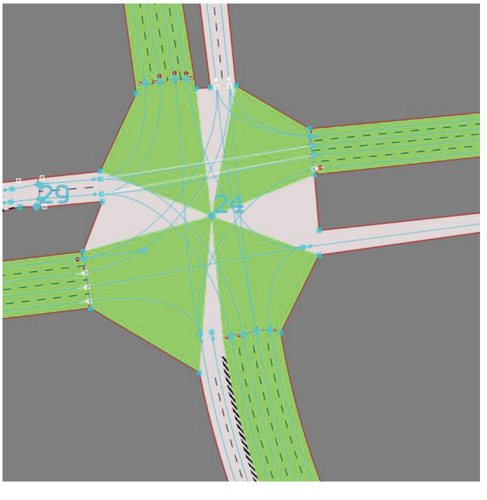


Figure 7.25: Paramics Discovery: entrance links (green) joining at node 24

To conclude, the main motivation for developing this design of the infrastructure regards the laborious exchange of data between models and with external databases. This mutual exchange of data is hampered by a lack of data-standardization and frustrates the portability of data (inputs, networks, ODs, etc.) between transport and traffic model packages As we have seen this has two reasons:

- The lack of a geospatial data standard for the transportation infrastructure;
- Every model package has its own design of model objects which impedes the exchange of data.

To shape this process efficiently, a reconciliation of the data design for the transport and traffic models, in combination with a mature (open) data standard for the transportation domain appears essential. Returning to the commercial model packages, we observe that while they provide a good modelling framework, they lack a design that allows an efficient data-exchange. A better adjustment between model packages and geospatial data sources would be beneficial as it reduces the efforts to build road networks for models. The availability of a mature geospatial data standard is a first and essential requirement. This could at least stimulate the providers of transport and traffic model packages to attune their design with this standard and provide the possibility of importing and exporting.

The proposed extension of the *CityGML* data standard for transportation that uses three levels of detail, and combines lines and spaces (as described in section 6.6), provides a suitable framework to enable the required exchange of data for transport and traffic models. The line structures resemble the idea of the transport infrastructure with *Links* and *CrossSectionElements*. The space based objects of the *CityGML* proposal, such as the position of signs, detectors and stop lines, can additionally be used to further define lane and road details.

Chapter 8

Case studies and evaluation of *OpenTrafficSim*

This chapter presents several case studies that relate to (1) the design of the infrastructure as described in the previous chapter and (2) the development and application of OpenTrafficSim. The chapter finishes with an evaluation of the OpenTrafficSim project.

8.1 Introduction

The case studies in this chapter further elaborate on the data design of the transportation structure and present two applications with the *OpenTrafficSim* software. Based on these case studies and further analysis, we present an evaluation of the *OpenTrafficSim* project with respect to requirements listed in section 3.7.

The *first* case presents an attempt to process the geospatial data from OpenStreetMap into a network that can directly be used as a network for *OpenTrafficSim*. On the one side, this case shows how data can be transformed to generate networks with more details about junctions connections. On the other hand, it illustrates the need for more detailed geospatial maps for transport and traffic, as has been proposed in chapter 6.

The *second* case (section 8.3) presents the modelling of the Network Transmission Model for The Hague. As part of this case, we show how information of modelled elements at a macroscopic (link) level can be propagated to the NTM model that has areas as its basic entities. By means of procedures and a hierarchical zoning systems, this transformation has been fully automated and provides an efficient way of using the existing network data. This case is based on a study by V.L. Knoop (Knoop, Tamminga et al. (2016)) who developed the concept and design of the NTM model. The implementation and calibration of the model for the city of The Hague has been conducted in close collaboration between Victor Knoop, Alexander Verbraeck, who wrote parts of the NTM-code, and this thesis author.

The third case describes a simulation project where we extend the functionality of *OpenTrafficSim* in order to fuse information from traffic sensors and traffic lights into a real time traffic simulation model.

On the basis of this case study and additional analysis, the chapter finishes with an evaluation of OpenTrafficSim (v1.01), by checking if and to what extend the project matches the requirements as listed in section 3.7.

8.2 Applying the network design in an urban environment

The import process of a transportation network from OpenStreetMap (OSM) illustrates the specific objects, topology and the attributes that are required to create detailed road simulation networks. The OSM network is characterized by so called *way* elements that provide information of a road by its direction, Every *way* may contain information about lane configuration and turning movements, but information about these attributes is optional. In

OSM elements are coded by means of tags in an open tagging system. Guidance is supported by providing best practice and typical coding schemes (Hochmair, Zielstra et al. (2015)). Each tag consists of a key and a value, for instance "maxspeed=>50" and "lanes=>3".

As OSM is a free, community driven map (Goebel, Skuballa et al. (2016)), its quality and completeness depend on voluntary contributions. Comparisons between OSM map data and real world observations, show that not all road details are fully provided. While this depends on the country and region, a quick scan for the city of The Hague reveals that a significant part of the roads does not contain information on all aspects that are relevant for the infrastructure description. Particularly relevant for microsimulations is the configuration of lanes and junctions. While OSM provides the number of lanes as an attribute of the road element, it lacks details about the lane topology. A further issue regards the description of junctions. OSM provides information about the approaching roads by the "turn:lanes" tag that defines the turning lanes. For instance, a three lane approach with two straight and one combined movement is defined as "turn:lanes => through|through|through;right". A quick comparison of the OSM map with for instance the map-images from Google show that the turn information is not provided for every junction approach. While this incomplete provision of lane and turn attributes can be overcome in due time, a more structural problem regards the OSM object design that lacks a complete description of the road infrastructure as input for a fully-fledged microsimulation model.

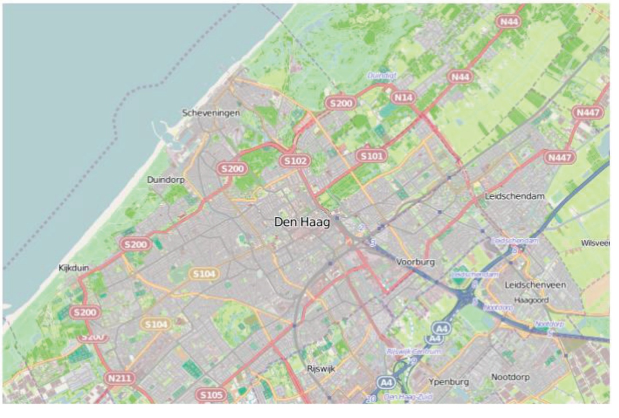


Figure 8.1: Representation of the OpenstreetMap data for the city of The Hague

The process of importing and transforming the road network data requires the following steps:

- Define the area of study and import OSM data
- Select and convert the relevant data layers into a transportation network format
- Transform the data into a suitable format for simulation
- Validate network elements and attributes
- Create additional simulation data such as connecting lanes at junctions

Figure 8.2 shows the sub area cordon (green) that is used to create a mall network of all roads within this area. It comprises a combination of highways and a mixture of urban road types.

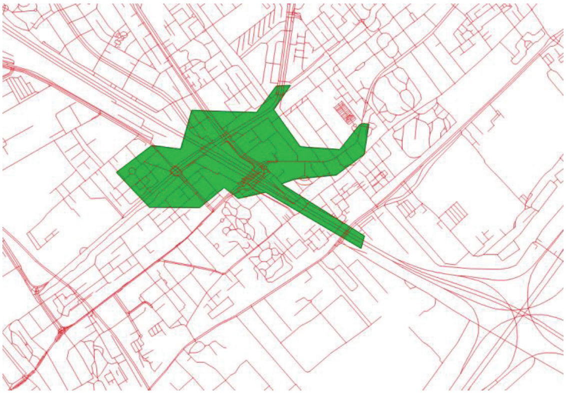


Figure 8.2: Sub area for a part of the OSM road network from The Hague

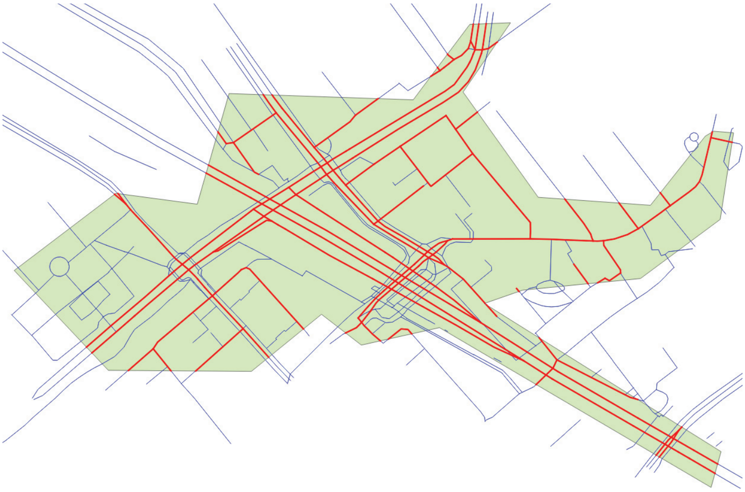


Figure 8.3: network for all modes (blue) and selection of roads (in red) within the subarea

From all roads, pedestrian paths and cycleways, the roads that allow vehicles (shown in red) are selected.

The OSM network requires various preparations before it can be imported in OpenTrafficSim.

Step 1: creating nodes and edges

The first step converts the road network into a collection of nodes and edges, that allows the creation of a network with valid routes. The red marked road in Figure 8.4 crosses several junctions, and needs to be split into separate edges for creating routes. This splitting process is not necessary for grade separated crossings: the property "layer" indicates roads at other levels (as shown in Figure 8.5).

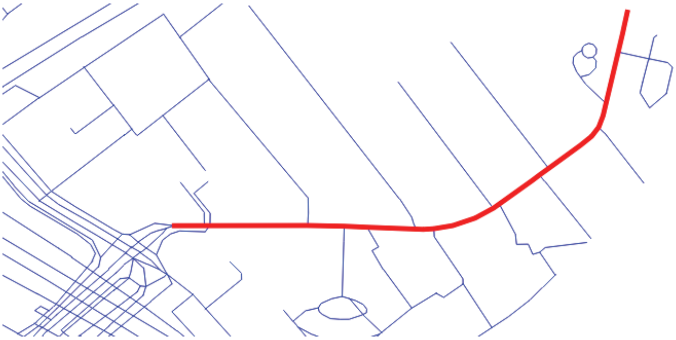


Figure 8.4: OSM network before splitting: the red coloured road overpasses several junctions

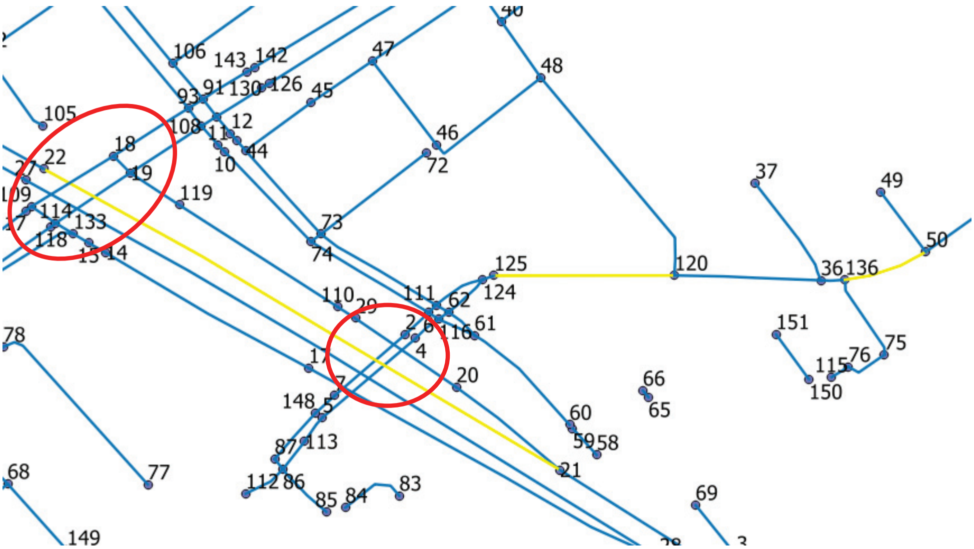


Figure 8.5: OSM network after splitting the roads in nodes and edges, showing the grade separated crossings of the highway

Step 2: Creating lanes

Subsequently, the nodes and edges are imported into OpenTrafficSim, and lanes per edge are created using the OSM information about the number of lanes. As Figure 8.6 shows, this step creates a network that lacks information of junction movements. Moreover, the network contains some illogical situations with respect to the lane configurations, as indicated by the blue circles, showing (1) a two way road at the off ramp and (2) a junction where the number egress lanes is not in line with the approaching lanes.

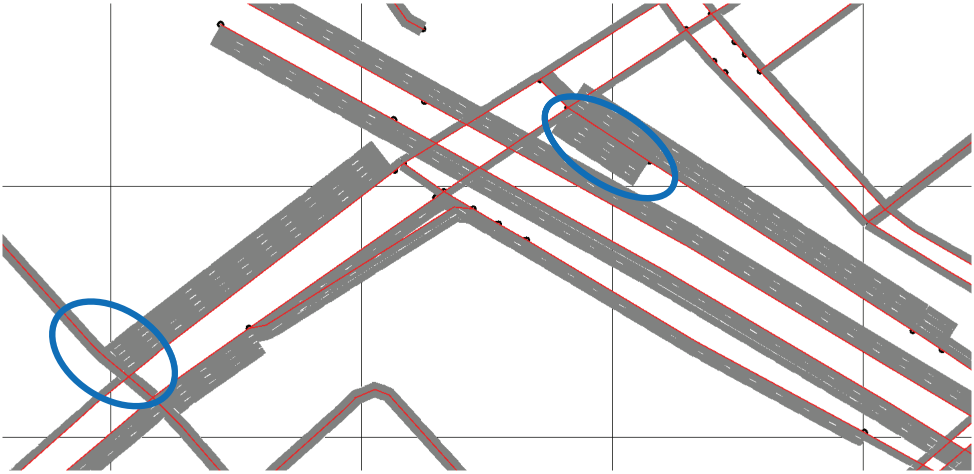


Figure 8.6: result after first step of importing the OSM roads

By adding validation procedures, the most obvious errors, or incomplete data, are filtered and adjusted. In this case, the assumption is that for situation (1) an off ramp from a motorway, defined as a one-way road, should by definition be a one-way road, and situation (2) the number of egress lanes minimally equals the number of approaching lanes.

Step 3: expanding the junction area

The OSM network lacks information about the junction topology. More specifically, only the turning lanes can be defined, but the connections between the incoming lanes and the egress lanes, as well as the exact topology inside the junction area, is lacking. The import procedure of the OSM network, therefor uses the provided information to create an initial junction area that is consistent with the input data. The process starts by analysing all nodes, and determine whether a node:

- Reflects a junction,
- Indicates a merge or split,
- Connects two roads or demarcates the end of a dead-end street.

This section describes the heuristics to create a plausible junction design.

To determine the function of a node, all links that enter or leave the node are identified. The first step is to detect the arms of the junction. Incoming and egress links are assumed to be part of the same arm, if one of the following rules holds:

1. The start node of the incoming link equals the end node of the egress link, and vice versa.
2. The street names are equal, the angle between the incoming and egress link is less than any other combination of links, and this angle is less than 90 degrees.

Figure 8.7 illustrates these rules, and also shows that the line geometry of the roads does not directly clarify if the node represents a junction or a continuation of the road. In this case, the street names fulfils the other requirements of rule 2).

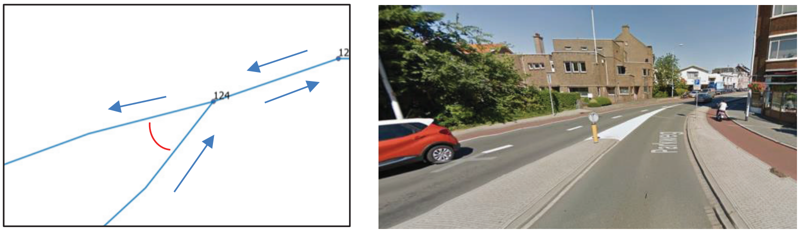


Figure 8.7: determination of the arms of a junction or node (source: Google maps and OSM)

After identifying the arms, the links are ordered in anti-clockwise order, which requires two steps. The initial ranking of the links is based on their angle, measured as the direction between the last vertex of a link and the junction node. Subsequently, if two successive links are part of the same arm, the egress link is ranked before the incoming link.

Based on this node and arms description, heuristics determine the role of the node. In case of two arms, the node represents a continuation of a road, but may be used to reflect changes of road characteristics. In case of three and more arms, the node will represent a merge, split or junction[16].

A further complicating factor arises because junctions in OSM are often represented by multiple nodes. Heuristics to determine a junction that is described by more than one node, are less obvious. Figure 8.8 for instance, shows a junction that contains three nodes (109, 114 and 133). Instead of complex heuristics, the addition of junction object that defines its nodes, would be preferable.

[16] In this case only some simple heuristics are used, as the objective is to illustrate which information is required to create microsimulation networks, and as a result, what additional information would be required to make OSM and likewise digital maps suited as a data source for microsimulation models

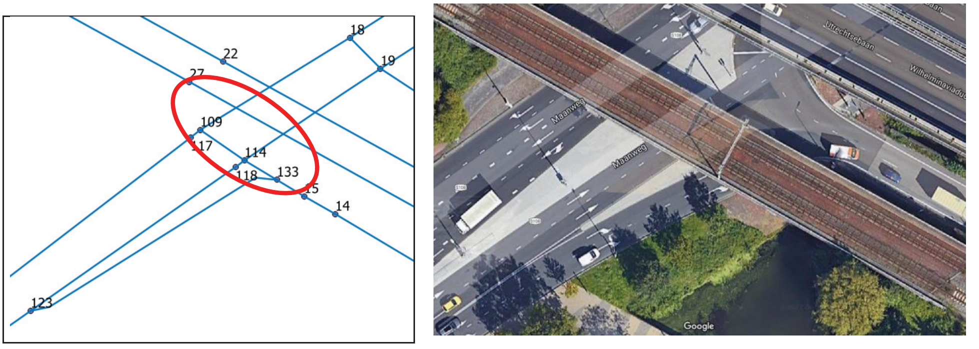


Figure 8.8: complex junction in OSM and real life (Google)

This case study does not provide heuristics for these complex situations, and creates separate junctions for single nodes.

For all nodes that represent a junction (or merge/split), an expansion procedure is implemented aiming to create a junction area that can be used for traversing the junction. The result is shown in Figure 8.9 with the blue marked nodes as the original input from OSM, and the black nodes that are created in the expansion process.

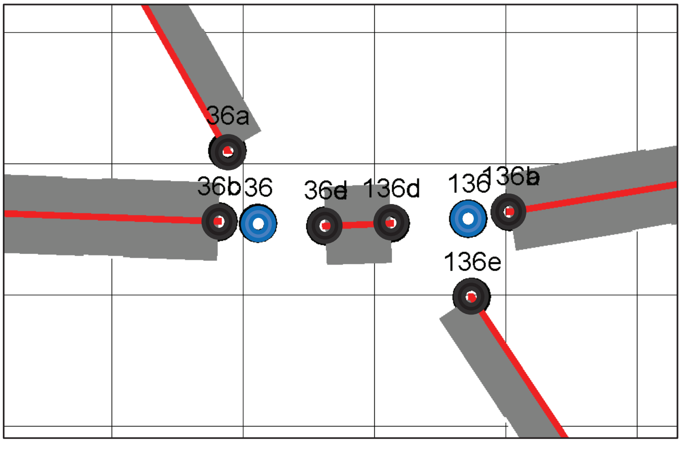


Figure 8.9: expanding the junctions

Step 4: creating the turning lane movements

After creating the junction area, the turning movements are determined. This procedure depends on the information that is provided by the OSM network. The simplest expansion is shown in Figure 8.10 where:

- none of the roads have any information on both the number of lanes of the road, in case we assume there is only one lane, and
- no turn lane information, in case we assume that all logical turning movements are allowed.

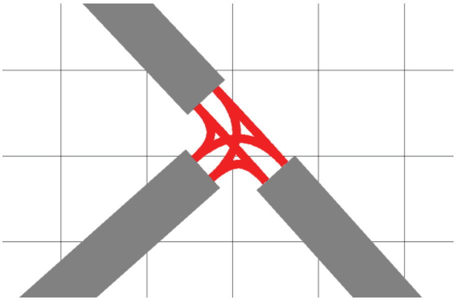


Figure 8.10: creating turning movements

In case the “turn:lanes” tag is provided we can define the turning lane movements from the incoming roads, If available, the “lanes” tag is used to determine the number of lanes at the egress link. If the “lanes” tag is not provided, the default assumption is one egress lane. But in case the number of incoming is higher than the egress lanes, the number of egress lanes is increased to the number of turning lanes. Figure 8.11 shows the outcome of this procedure that increases the number of egress lanes (blue circle). The green circle shows a situation where the number of available egress lanes is larger than the turning lanes. By default, the turn lanes are connected with the two rightmost lanes, but this choice is arbitrary. In both cases there is insufficient information, which necessitates a comparison with the real world situation.

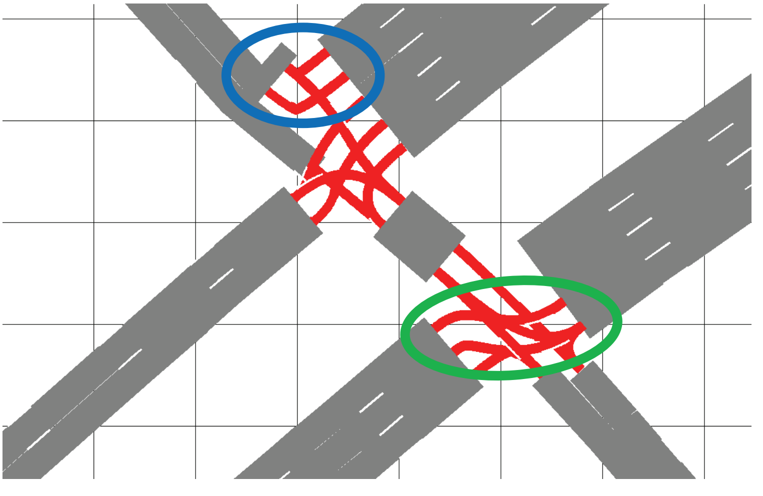


Figure 8.11: expansion of a complex junction

Concluding remarks:

The OSM network typically resembles the level of detailedness of many digital road maps. While it offers a significant amount of information about the road, the junction details to create a map with turning information that is required for microsimulation are lacking. A further description of junctions is required to improve it.

8.3 The Network Transmission Model for The Hague

In the remainder of this chapter we focus on applications based on *OpenTrafficSim*. The first case shows an application of the Network Transmission Model. The study illustrates the ability to apply the *OpenTrafficSim* network design not only for microsimulation, but also for coarser levels of detail.

This case is based on work by V.L. Knoop (see for instance Knoop and Hoogendoorn (2013), Knoop, De Jong et al. (2014), Knoop and Hoogendoorn (2015), Knoop, van Lint et al. (2015)) who developed the concepts and design of the NTM model. The implementation and calibration of the model for the city of The Hague that is presented here, has been conducted in close collaboration with Victor Knoop and Alexander Verbraeck from Delft University of Technology.

We describe the building process of a Network Transmission Model that uses information from a road network, to establish a coarser model that is based upon the interaction between homogeneous traffic zones. It illustrates the aggregation procedures to create the attributes of the traffic zones, and shows that it is not possible derive all of the attributes in a straightforward manner. As will be shown, this partly depends on the physical characteristics of the infrastructure, but is also influenced by many behavioural aspects that can vary by time.

8.3.1 Exploring the Network Transmission Model as a tool for dynamic traffic management

While the application of traffic flow models traditionally supports ex-ante studies, the application in on-line decision support is gaining more and more interest. To control traffic optimally in both normal situations and in case of specific circumstances such as incidents or events, information on the actual traffic situation and the ability to produce predictions for the short time, i.e., ten to twenty minutes, is essential. This requires a dynamic traffic model. On-line traffic management in a traffic control centre requires that information is provided almost instantly. This limits the possibilities to implement traditional dynamic assignment or micro-simulation models, due to the fact that the running times of these models are yet far too large to provide instant information about short term traffic predictions. An operational manager in a traffic control centre requires real time information, with a time lack of some minutes at most, to decide which operations need to be taken (Knoop, Tamminga et al. (2016)).

In a case study for the city of The Hague, the Network Transmission Model (NTM) has been tested as one of the components to support real time traffic control and the planning of road works. This is a dynamic model, which is able to describe changing traffic patterns in time and predicts the spillback of traffic to upstream roads or zones. The basis of the NTM-model are zones (rather than vehicles or roads, see also section 5.3), so the whole city can be described by a low number of interacting elements. This makes the NTM-model very fast to run.

The remaining sections focus on the implementation of the NTM, which is based on Knoop, Tamminga et al. (2016). Parts of the article are directly cited in this section, and can be identified by the use of quotes at the start and end of a section.

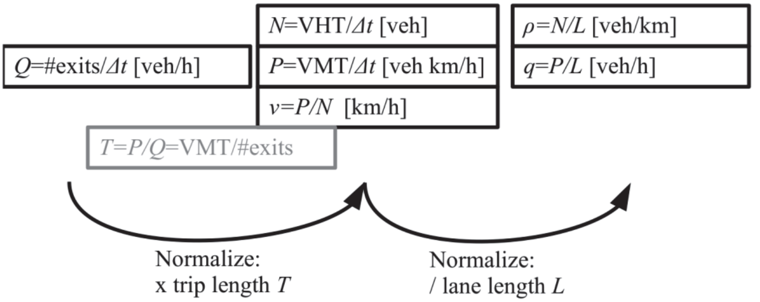


Figure 8.12: Relationships between the different variables (source Knoop, Tamminga et al. (2016))

Table 8.1: The symbols used (source Knoop, Tamminga et al. (2016))

Symbol	Meaning
N	Accumulation: the number of vehicles in the zone
P	Production: the amount of flow in the zone
k	Density: number of vehicles per lane kilometer
q	Flow: vehicular flow per lane kilometer
v	Speed: speed of the vehicles in the zone
L	Lane length: total length of the lanes in the zone
Q	Performance: the outflow out of the zone per time interval
u	Speed at a link
$\mathcal{L}$	Links in a zone
S	Length of a link
C	Capacity of a link
p	Nr of lanes of a link
T	Trip length: average length of a trip in the zone

"Already in the 1960's, traffic engineers envisaged that for traffic in an area there would be a relationship between the number of cars in an area and the speeds. The relationship between number of vehicles and speed was shown using simulation. The idea of an area-based relationship gained momentum when Daganzo showed the actual arrival rate could decrease for too many vehicles. A major step then was paper by Geroliminis and Daganzo (Geroliminis and Daganzo (2007)), showing that this relationship holds in practice. This relationship is called the macroscopic fundamental diagram (MFD), or network fundamental diagram, which exists in several, slightly different versions, relating slightly different variables. Figure 8.12 shows the relationships between the various variables. The base variables are the accumulation and the production. Accumulation is determined by a generalisation of Edie's definition (Edie (1963)), being the vehicle hours travelled divided by the aggregation time. Likewise, the production is determined by the vehicle miles travelled divided by the aggregation time. The more common equivalents of accumulation and production are density and flow. These are obtained by dividing the accumulation and the total lane length in a zone. Performance, or outflow rate, is obtained by dividing the production by the average trip length."

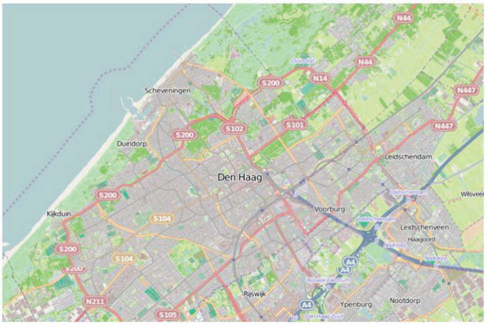


Figure 8.13: The city of the Hague

For this study, the NTM-model works as follows. "An urban area is split into different zones. In each of the zones an MFD is assumed to hold. Traffic is modelled destination specific, all with a route from the zone to the destination. Demand and supply functions are established, showing the demand of vehicles out of the cell and the supply of vehicles in the receiving cell. This gives the number of vehicles going from one cell to the other. The demand function is equal to the MFD (in accumulation-performance from) of the sending cell. The supply function is equal to decreasing part of the MFD of the receiving cell and for accumulations lower than the critical accumulation it is equal to the capacity. Moreover, the demand over an edge between two cells is limited to the (road) capacity which crosses that edge; for this moment, this is a parameter in the model, for which later we will find a value based on the road capacities. In case the demand to a cell is larger than the supply, supply will be allocated proportional to demand (over different sending cells, as well as traffic with a certain destination). For the full description of the model, we refer to (Knoop and Hoogendoorn (2015))."

8.3.2 NTM-model set-up

"For the NTM to work, the following elements are required:

1. a network defined as zones (areas)
2. a dynamic OD matrix
3. capacities of the edges
4. an MFD per zone, indicating the exit rate as function of the accumulation
5. a routing strategy

This section describes how the first two are obtained. These two do not change in the calibration phase. In the calibration or testing phase (described in section 8.3.3 and 8.3.4), the right values for third to fifth elements are found.

To test the quality of the model we use the test case of the city of The Hague, see fig. 4. This city has over 500,000 inhabitants, an area of approximately 10x5 km. It is noteworthy that the city has no freeway ring road. There is a ring road (S200/N14/A4/N211), but at places the speed limit is 50 km/h. At one side it lies at the sea, which influences the origin-destination (OD) pattern, as well as the network. At the east side, a freeway passes north-south direction. Another freeway goes into the city centre (A12)."

Zones

"The process followed is as follows. The first element used to build the model is the static model of the The Hague municipality. In this model the network is divided into zones, and for an afternoon peak hour the total traffic demand is given for each OD pair (zones). Origins or destinations outside the modelled zone are aggregated in boundary zones at the side of their true origin and destination. The division in zones is shown in in Figure 8.14a (the thin blue lines are the roads and the bold black lines are the zone separations). The zones are too small for a good Macroscopic Fundamental Diagram (MFD): no homogeneous conditions can be found if just several roads (with possibly some queues) are in the network. Moreover, internal congestion within the zones is unlikely, hence an MFD will not hold."

(a) All zones. The dark lines are the zone boundaries, the blue lines are the major roads

(b) Aggregated zones. The black arrow in the left hand side indicates the movement C80-C16-C96

Figure 8.14: Zones in the traffic models

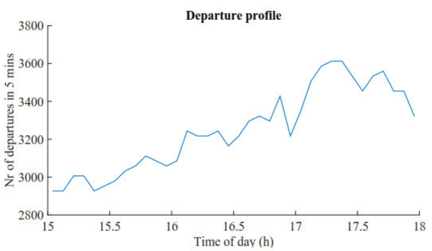


Figure 8.15: The profile of all departures as function of time

"New zones are constructed, by which neighbouring zones with similar characteristics are combined into larger zones. This was done based on expert opinion of the authors that are familiar with the city. They tried to combine them into zones with similar size and to separate residential zones from working zones or shopping zones. The areas have similar size, but separation of type was not possible for all zones. The resulting 29 zones are shown in Figure 8.14b. The OD matrix is adapted to the new zoning by aggregating the trips with origin and destination in the small zones into the larger zones."

Origin Destination matrix

"The simulated time for the NTM is from 15.00 to 18.00, which is 3 hours, of which the first 30 minutes are considered a warm up period. This time is being based on the longest trip in the network. The static OD matrix has 78 thousand trips in 2 hours, which needs to be changed into a time dependent, dynamic matrix for 3 hours. For the transition to a three hour period, we multiplied the original OD matrix by 1.5 (3 hours/2 hours), to get a representative for the total number of trips. Within this period, we require a peak, which is derived from counts on a non-congested part of the network. The total number of trips is now scaled based on that profile. Figure 8.15 shows the total number of trips departing in 5 minutes. The same scaling has been done for each OD pair."

8.3.3 Tunable parameters

"The dynamics of the network transmission model are governed by the model equations and the parameters. The tunable parameters are:

- the total capacity between two neighbouring zones
- the total traffic production of a traffic zone
- route choice of traffic through the network

The succeeding subsections discuss these items and shows how we quantify the relevant equations and parameters. For these parameters an initial choice is made based on reasoning. Afterwards, the model is run. The dynamic traffic states predicted by the model are assessed by practitioners from the The Hague municipality, which know the traffic states occurring in real life. Based on the comparison of model and real life, the parameters are adapted. The subsections consist of the meaning of the parameter, as well as the way they influence the traffic dynamics."

Capacities of the edges

"The model is implemented in the *OpenTrafficSim* simulation platform. To find the neighbouring zones, the platform first detects which zones have a common edge. Subsequently, for all these pairs of touching zones, the roads that cross their common edge are identified. If one or more of these roads have a junction in both zones, we assume that these zones are connected by this link and thus are neighbours in the sense that there are traffic flows possible between the links. The capacity between the neighbouring zones is calculated by the sum of the capacities of the roads that connect these zones. These total capacities are stored for every pair of neighbours. In cases of incidents this capacity can be lowered to represent the loss of road capacity. Alternatively, in case of events or evacuations specific measures could be taken to increase capacity, in which case the parameter value would increase to values above one.

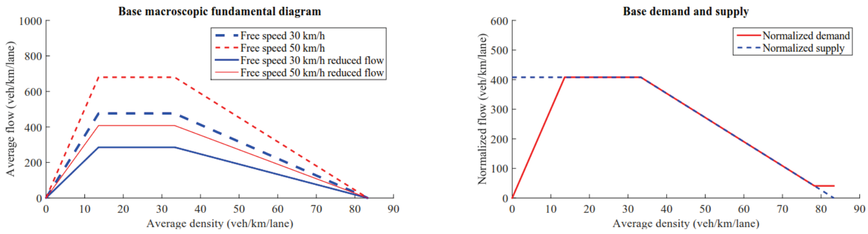
For the case at hand, this base estimation did provide reasonable results for most zones. At the edge where the traffic has to merge into the motorway, notorious problems are known with the traffic lights and internal spillbacks on the junction, limiting the inflow into the motorway. For this reason, the capacity for the edge from the city zone to the zone with the motorway has been further reduced, to a value where the congestion matched the real life value."

Characteristics of the MFD per zone

"Based on threshold values (1200 veh/h/lane) the number of lanes (p) for each of the roads l in the static model is found ($P_l = floor\left(\frac{C_l}{1200}\right) + 1$). Multiplying this by the length of the link (*S*) and summing over all links in the zone ($\mathcal{L}$) gives the total amount of lane length in the zone ($L = \sum_{l\in\mathcal{L}} p_l S_l$). Also, the speed limit for each of the roads is known, which leads to a free flow speed, calculated by an average of speeds weighted by the capacity of the links:

$$\left(\frac{dP}{dN}\right)_{N=0} = \frac{\sum_{l\in\mathcal{L}} u_{free,l} S_l p_l}{L} \qquad (1)$$

The NTM requires a diagram which relates the performance (outflow in veh/h) as function of the accumulation. This is based on the characteristics of the roads as well as theoretical considerations; a full theoretical approach for can be found in Laval and Castrillón (2015). In this case, we chose a more practical approach. For each zone, we use a base MFD as shown in Figure 8.16a. The figure uses fixed values for k1, k2 and k_{jam} as we will initially use (see sequel of this section). Equation 1 is used for the slope in under-critical conditions. This assumption means that the capacity varies with the free flow speed. Indeed, this seems reasonable since also at a road level lower speeds (in urban areas) lead to lower capacity; moreover, for the MFD a low speed could indicate high interference with traffic lights, which also decrease capacity.



(a) The base fundamental diagram for all zones (b) Base demand and supply

Figure 8.16: Characteristics of the zones. Note that the units have to be scaled by the lane

The slope of the production (equation 1) and the critical density would yield a capacity. However, that capacity would only hold in case of an unrestricted flow for all roads, including no restrictions at any intersection, which is not the case. Therefore, a multiplicative factor is applied to the capacity. This faction should account for intersections. One can think of the ratio of green time over the cycle time for an intersection (which should be on average 0.5). A better interpretation is the weighted average green time, i.e. the average green time weighted for the capacity of the incoming links. On intersections the main road is likely to get a larger share of green time, and has a higher capacity. Therefore, average availability of capacity exceeds 0.5; in our case 0.6 was chosen.

From this MFD, the demand and supply can be derived. Compared to the original NTM formulation (1), a change is made in the demand function to avoid complete and insolvable gridlocks. In fact, the demand in the congested branch does not return to zero for jam density, but remains at 10% of the capacity, see Figure 8.16b.

Note that for the base MFD in Figure 8.16a, the horizontal axis shows the vehicles per km lane length, and is basically an average flow - average density diagram for the zone (right column of Figure 8.12). To come to the MFD we multiply by the total length of all lanes in the zone (from the map and the number of lanes), and the relationship between production and accumulation is known. Finally, a constant trip length of 1 km is assumed (all zones are similar in size). We divided the production by this trip length, which gives a relationship between exit rate and the accumulation. We are aware the trip length can have an effect (see also Leclercq, Parzani et al. (2015)); in this first-order estimate we assume a constant trip length.

Varying the values for the different densities sorted various effects. The most important value was the density from which the production would decrease (k_2). Lowering that value causes outflow to decrease and the thereby the accumulation to increase, causing an even lower outflow. This is the most critical value for a zone. The density at which the speed first starts to reduce (k_1) is relevant for getting the right speeds in slight congestion. However, it will not influence the propagation of congested patterns through the network, hence its influence is more local. The jam density (k_{jam}) indicates how much extra storage capacity there is in overcritical conditions. Combined with the capacity and the value of the start of the decreasing branch (k_2) k_{jam} indicates the speed at which congestion spills backwards. By experimenting with different values for the densities the following values were found to give speed patterns over time which matched the experience of the practitioners at the Traffic Management Division.

Using experience from earlier research (e.g., Laval and Castrillón (2015), Knoop and Hoogendoorn (2013)) and trying some values (equal for all zones), we fixed k_1 to 14 veh/km/lane, the k_2 to 33 veh/km/lane, and k_{jam} to 83 veh/km/lane. These values are low, because a homogeneous state with all roads and all lanes in jam density (which would result in a k_{jam} of 125 veh./km or higher) cannot be reached. In fact, traffic will stop in an inhomogeneous network at lower densities (Knoop, van Lint et al. (2015)).

The MFD is tuned for normal circumstances. In case of a serious accident, the performance of an area might decrease significantly. This can be quantified by tuning the area-specific MFD diagrams, for instance by decreasing the average critical density and maximum capacity by a certain factor.

In the calibration process with the traffic operators especially the accumulation where the production decreases (k_2) has received attention, since this turned out to be the critical parameter. Lowering this parameter would have the most serious impact on traffic flow, since it starts congestion, which due to the increasing accumulation then worsens."

8.3.4 Routing strategy

"For routing we follow the description in (1). For each zone the routing is to a destination is coded as a split fraction: which part of the traffic to destination *D* goes to which neighbour n. This is indicated by $\Psi i,D$, which is a vector of which the length is the number of neighbours of zone *i*. This is updated every time interval of length τ . At that moment in time, the fastest routes from a zone towards a destination is determined, given the speeds in all the zones. Travel times are derived from the speeds and they are disturbed using an error with a normal distribution (mean 0, 10% st. dev). This is repeated several times, leading to different routes. For these routes, it is considered which part goes to neighbouring zone n. This gives a routing vector for the routes at that time step: $\psi i,D$. The new routing vector now is a weighted average of the old routing vector Ψ and the new one ψ:

$$\Psi_{i,D}^{new} = (1 - w)\Psi_{i,D}^{old} + w\psi_{i,D} \qquad (2)$$

The weight w indicates which fraction "reconsiders its route" during the time interval τ between two route updates. Note that this is an interpretation: for instance, not the same travellers are updating their route all time intervals, and moreover, no routes are assigned, but rather fractions are updated. Due to this scheme of creating routes which also start differing from the next zone, the number of draws on travel time does not need to be very high. Also considering computational load, we choose 6 routes for each probit round

In the testing phase, the route update times τ as well as the fraction of drivers changing their route is changed. If the update frequency was low (less than once per 10 minutes), congestion had time to grow (due to the fact that no equilibrium assignment was used) and then with a new route update it would suddenly reduce and another congestion area would grow. This was not in line with the experience of the practitioners of the Traffic Management Division. A fast update with a large share would lead to traffic homogeneously spread over the network, which was also not according to their experience.

Based on the traffic patterns and the experience from the traffic management centre the route choice was set to a re-routing fraction of 5% every 5 minutes (w in eq. 2 equals 0.05) – similar to a method of successive averages. With a longer time, the traffic states were too much fluctuating. A shorter re-routing time kept the traffic states stable. The amount of re-routing relatively low, suggesting indeed that most people do not adapt their route based on the changing situation en route. This was according to the traffic management centre the most realistic situation."

8.3.5 Results

"This section describes the results of the simulations. First the base results for a recurrent peak, and the adjustment by practitioners of the Traffic Management Division are presented.

Figure 8.18 shows the simulation results graphically. When using only the base MFDs as explained above, the road network experiences some congestion (see Figure 8.18a and b), just like in reality. The first 10 iterations of adaptation were used to find the abovementioned overall parameters (MFD parameters and routing update parameters). Now we focus on adapting the parameters of the individual zones and MFD boundaries. In this paper we only present time snapshots of the traffic states, but in during calibration, movies for the whole peak period have been used, showing the dynamics of the congestion patterns.

With the initial parameters not all congestion as represented as in reality. As explained in section 8.3.3, the local traffic operators explained that several traffic lights upstream of the entrance to the freeway A12 cause congestion. The calibration step taken is to reduce the edge capacity for the flow from the east towards the city centre.

By consequence the traffic is more congested, and spilling back in the way the traffic operators expect, and is seen in real life (see Figure 8.17 for a typical situation of the evening peak). The congestion starts in the right area and spills – at approximately the right pace – back to the more upstream areas.

The amount of congestion is adapted by adjusting the capacity at the edge: less capacity will mean more congestion. The adaptation was done together with the traffic operators, which

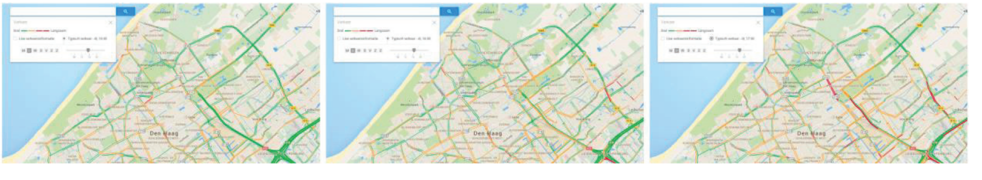
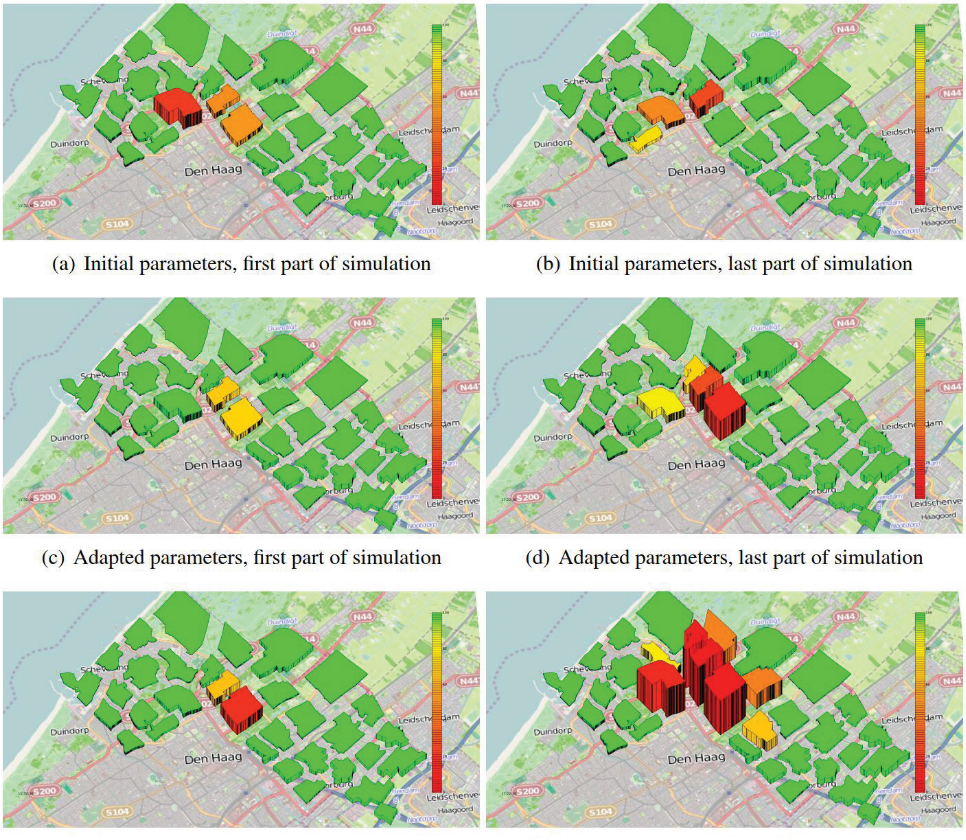


Figure 8.17: The typical traffic state in Google Maps



(a) Initial parameters, first part of simulation (b) Initial parameters, last part of simulation

(c) Adapted parameters, first part of simulation (d) Adapted parameters, last part of simulation

(e) Adapted parameters, accident, first part of simulation (f) Adapted parameters, accident, last part of simulation

Figure 8.18: Results of the simulation

shows the quality of the model in the second requirement, i.e. "The tuning of the model should be intuitive by operators of the traffic management centre".

The tuning parameters turned out to work as described in section 8.3.3. After finding the overall parameters in 10 iterations (section 8.3.3), 14 more iterations were needed to get a satisfactory result – a few hours work. Adaptations carried out were changing some boundary capacities, changing k_1 for two zones (one with a change of speed, the other with a change of capacity), and change of k_1 and k_{jam} for one zone: k_{jam} was increased in order to avoid further spillback of the traffic jam.

After the capacity at the edges was adapted, congestion occurred and propagated further upstream. The model works with zones which can get congested (for traffic to all directions), whereas in the ideas of an "evening peak" the traffic engineers were only concerned with traffic leaving the city. The model showed situations that in case of high congestion inbound traffic was also hindered, but the congestion mainly spread to the upstream locations for the main traffic direction, i.e. the outbound direction. The resulting combination of congestion propagation was well in line with the experience of the Traffic Management Division.

Therefore, the most important conclusion is that after the calibration the model was well able to predict congestion at the right location at the right time.

The model is very fast: simulating a 3 hour period on a (one core of a) laptop (Dell Latitude, Intel core i7-2530M CPU 2.8 GHz, RAM 4 GB) costs 15 seconds. The code is not optimized for computation time, and the programmers expect that optimization of the code can result in approximately 10 times faster computation times. If even further reduction in computation times for many scenarios is required, improvement of the hardware is possible, as well as making the computations for the different scenarios on different CPUs.

The model showed also undesirable outcomes. First, after congestion starts occurring, drivers chose a route around congestion via an infeasible path. The path C80-C16 (see the black arrow Figure 8.14b) is possible with a high capacity. Also the connection C16-C96 is possible. However, the sequence C80-C16-C96 is not possible due to a restriction in turning: the main traffic stream taking that route needs to continue straight (no right turns). In a future version, the routes can be coded using directional graphs, rather than via the destination-specific split fractions in the network, by which this problem can be overcome.

Secondly, traffic is assumed to have reached its destination once it arrives in the zone of arrival, which is not necessarily the case in reality. In fact, parkings can be full, leading to queues waiting at the entrance blocking other traffic, or traffic can keep circling to find an on-street parking spot. These outcomes should be included in a future version of the model; some ideas how to represent parking on a network level have already been developed (Geroliminis (2015))."

8.3.6 Face validation

"The previous section showed the results of the model where the model parameters were tuned to match the recurrent traffic patterns. The model has many variables (4 per zone for the MFD, one per neighbouring pair for the capacity, 2 for routing). There is a risk that the model is tuned to represent the reality, but the parameters are not correct – for instance due to overfitting. To test whether the model might be tuned towards the outcomes and does not have the processes modelled, we do face validation. In fact, two test cases which differ from the regular (calibrated) morning peak are run by the model, for which only the input parameters are differed. The model has not been calibrated further. The considered test cases are: (1) an excess demand due to visitors to the beach, and (2) an accident partially blocking the freeway A12 outbound. Both scenarios occur frequently, so the traffic engineers from the traffic management centre know the outcome. The situations are calculated, changing the demand for one origin (case 1), or the capacity of one edge (case 2).

The traffic state for case 2 is depicted in Figure 8.18e and f. In both cases the network becomes more congested at the locations which were expected by the practitioners. In case of the excess demand caused by visitors of the beach and leaving in the evening peak shows that congestion starts at the same locations as the average evening peak. However, due to the increased amount of people leaving the city, the level of congestion is higher and causes a spreading of congestion to upstream areas. While we did not quantitatively test the level of congestion with observed data, the phenomena that appeared seem to be face valid, as was confirmed by the The Hague Traffic

The first finding is that the model works reasonably well with a set of default parameters for all zones. This is surprising given the fact that the impact of loading patterns, traffic light settings and inhomogeneity are discussed extensively in literature. Apparently, a generic approach works reasonably well. With the adapted parameters, the traffic congestion is found at the right place, and the traffic dynamics are presented in a correct way. The working of the model parameters was considered intuitive by the practitioners of the Traffic Management Division: they were able to quickly adapt the parameters of the model to get a realistic outcome. Finally, the model worked very fast, well under the requirements set by the Traffic Management Division.

Shortcomings which came up in the model were that traffic in reality does not leave the network the moment it reaches a destination and that sometimes turning could be restricted. For both issues solutions will be tested. For the first, a limited reservoir in each zones could represent the parking capacity. In case of busy hours, this is full and vehicles have to circulate until there is space (Geroliminis (2015)). The modelling of these reservoirs could be similar to zones, with a supply function. For the routing explicit routes can be determined on beforehand, and during the route choice update the travel times on the predetermined routes are updated. That would avoid infeasible routes to be chosen."

8.4 Case study N201: extending OpenTrafficSim into a real time microsimulation

The N201 case study describes a real time simulation model to show (1) the functionality of the software and (2) the opportunities to modify and extend the *OpenTrafficSim* functionality. Subsequently, section 8.5 evaluates the software based on both the experiences of the case study and a further analysis of the software itself.

This project has been accomplished in 2015 and simulates a stretch of the Dutch road N201. The project uses a beta version of the *OpenTrafficSim* software project. For this project the *OpenTrafficSim* software has been modified and extended with new functionality in order to create a model that uses real time information from traffic light systems, in order to feed the traffic simulation with information about sensors and the state of the traffic lights. This project was commissioned by the province of North Holland and has been conducted by Sweco with the support of the *OpenTrafficSim* project team from Delft University of Technology.

The basic idea of the project is to fuse information from traffic sensors and traffic lights into a real time traffic simulation model. Such a hardware-in-the-loop simulation requires not only a valid traffic simulation model, but also poses requirements for the timeliness with which the simulation model interacts with the pulses from the traffic control system (Bullock, Johnson et al. (2004)). The common feature of vehicle actuated traffic control systems is that they use some type of vehicle detection and change the display of signal indications according to some prescribed traffic light control logic. This concept has been implemented by adding and extending the *OpenTrafficSim* classes with new methods and attributes.

The simulation model covers a road stretch of around 7 km as indicated by the blue coloured road in Figure 8.19 and contains seven junctions that are controlled by traffic light controllers. The controllers use pulse data from sensors to optimize the distribution of green time over the various turning movements. In the simulation model the traffic lights are simply copied from the traffic light controllers: the model does not implement its own control logic but only is only responsible for the movement of vehicles and its response for the traffic light states.

Figure 8.19: The blue line shows the simulated road

We have been developing two simulation variants: a real time system and a simulation that uses historical data. The traffic light controllers store all sensor pulses and traffic light states in a so-called VLOG data file.

The *real time variant* requires a synchronization between the simulation environment and the data-feeds from the controller. The DSOL simulation engine within *OpenTrafficSim* provides all functionality to ensure that the simulation runs in real time. Moreover, the simulation runs fast enough to simulate traffic, process the information from external data feeds and wait for the clock to start the next step of the simulation. The off-line variant uses *historical data* from the VLOG data, and is used for developing and testing the model. The main difference with the real time variant is the absence of the real time clock. The off-line simulation runs faster and is thus better suited for testing the model. In the remainder of this section we elaborate on the off-line simulation variant with historical data.

The simulation of the N201 requires a representation of the lane geometry at roads and junctions. This appears possible with the *OpenTrafficSim* and does not require any extension of the software. In addition, lane objects such as sensor, stop line, the signal head and its connection to a specific lane, and stop line are required. In order to connect the simulation with the pulse data from the VLOG data, additional functionality has been added to the *OpenTrafficSim* project. Firstly, the sensor object has been extended to allow for the generation of a vehicle whenever the sensor receives a pulse of a passing vehicle. The so-called

GenerateSensor class extends the **AbstractSensor** and connects the sensor with the **VehicleGenerator** class that provides the methods for the generation of vehicles at a specific sensor location. Only the sensors at the entrances of the N201 road network are denominated as **GenerateSensor**. Any other downstream sensor is not capable of generating new vehicles.

Before generating a vehicle, the **GenerateSensor** checks if there is actually sufficient space to insert a vehicle. The reason for such a check is the possibility that the sensor is occupied by a preceding vehicle. This could happen if a series of vehicles is generated at the sensor, drive to the junction, and then have to wait for a red light. When the waiting row hits back to the sensor, this impedes the insertion of a new vehicle. Such an event might indicate that the simulation induces longer waiting rows than in real world. A possible reason is that the average distance between vehicles in a waiting row in the simulation is longer than the actual behaviour in waiting rows. Other reasons might be a difference in the length of the vehicles or the distribution of vehicles over the lanes towards the junction. In case a vehicle cannot be inserted, the vehicle is added to a list of vehicles. At every timestep, the simulation checks if the sensor is still occupied. If not, the first vehicle from the list is generated at the sensor and removed from the ´waiting list´. The initial speed of the generated vehicle is chosen by the *calculateInitialSpeed* method and returns the minimum of (1) the vehicle′s maximum speed, (2) the legal speed limit and (3) the speed of leading vehicles in the simulation.

In addition to the pulses from the sensors, the VLOG data also provide information about the state of the traffic lights. The connection with *OpenTrafficSim* is provided through the *TrafficLight* that extends the **SimpleTrafficLight** class. The message from VLOG is translated to a signal-state event of a specific **TrafficLight**. The *setSignalState* method switches the traffic light for the specific lanes that it connects to. Figure 8.20 shows the simulation at one of the junctions of the N201. The colours at the stop-lines indicate the state of the traffic lights (red and green).

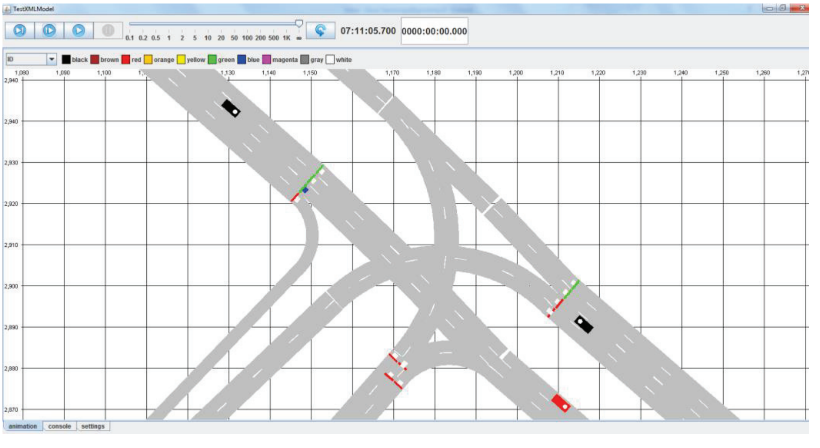


Figure 8.20: The blue line shows the simulated road

One of the results of this traffic simulation is the estimation of travel times over the N201. At this road, licence plate recognition cameras are used the measure the travel times of passing vehicles. Using only the data generated by traffic light controllers as input, the real-time simulation-model has been used to calculate the travel times. Figure 8.20 shows there is a reasonable match between the measured and calculated travel times, indicating that the simulation provides realistic travel times.

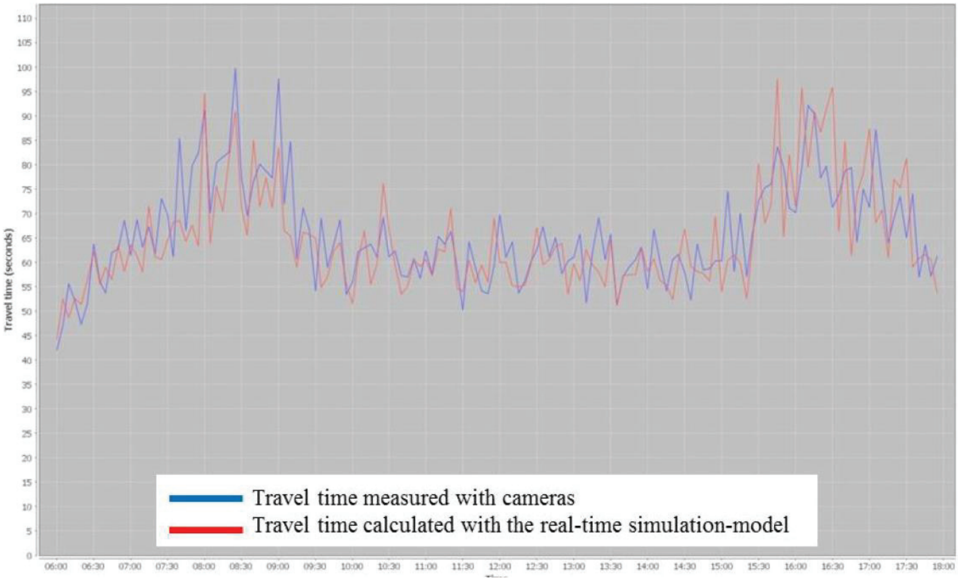


Figure 8.21: Comparison of travel times at the N201 measured with license plate recognition cameras (blue) and the real-time simulation-model (red)

8.5 Evaluation of the open source software project *OpenTrafficSim*

The target groups for *OpenTrafficSim* regard (1) education and study, (2) scientific and applied research and (3) innovative companies to simulate and test innovative applications. *OpenTrafficSim* has been developed with Java as the programming language. This section evaluates the current version of *OpenTrafficSim* (v1.01), by checking if and to what extend this open source software project matches the requirements as listed in section 3.7. The evaluation relates to three topics: the software quality, project governance and user friendliness.

8.5.1 Evaluation the implementation of the NTM application in *OpenTrafficSim*

The NTM model has been implemented in the *OpenTrafficSim* suite. While the application strongly deviates from the *OpenTrafficSim* microsimulation, the NTM still benefited significantly from the available functions and utilities.

Firstly, as the NTM is a simulation model, it perfectly fits with the simulation package DSOL. In the application, the DSOL simulator is used to schedule the events properly.

Secondly, the NTM project uses some of the basic objects from *OpenTrafficSim* such as the **OTSLink** and **OTSNode**. Both of them have been extended to meet the additional requirements for the NTM project. A graph is created from the links and nodes and used to find the shortest routes between origins and destinations. The length of one of the routes (Scheveningen –

Voorburg) has been compared with the result from Google Maps, and showed a comparable length. Figure 8.22 shows an example of the java code.

```
// test: from node 314071 (Scheveningen) to node 78816 (Voorburg)
NTMNode nSch = nodeMap.get("314071");
NTMNode nVb = nodeMap.get("78816");
DijkstraShortestPath<NTMNode, LinkEdge<NTMLink>> dijkstra = new DijkstraShortestPath<>(model.getLinkGraph());
GraphPath<NTMNode, LinkEdge<NTMLink>> sp = dijkstra.getPath(nSch, nVb);
System.out.println("\nScheveningen -> Voorburg");
System.out.println("Length=" + sp.getLength());
```

Figure 8.22: Code example for retrieving the shortest path

Thirdly, the NTM applies some external libraries that are already included in the *OpenTrafficSim* project. Examples are:

- *JGraphT*
  a Java library that provides mathematical graph-theory objects and algorithms.
- *Geotools*
  Provides standards compliant methods for the manipulation of geospatial data, for example to implement Geographic Information Systems

The use of these tools simplifies a lot of tasks, such as the import of GIS files and the construction of graphs and shortest paths.

8.5.2 Evaluation of the N201 simulation: extending the basic OpenTrafficSim simulation

The results from the N201 study shows that the *OpenTrafficSim* software provides a good framework for the simulation of traffic in an urban environment with traffic light controlled junctions. As the beta version was still under development, the case study also proved useful for testing the basic simulation methods with respect to car following and lane changing behaviour. During the project, the *OpenTrafficSim* software development team increased the quality of the simulation significantly. As a result, the simulation of the N201 ran without bugs and appeared adequate with respect to both the (visually) observed behaviour of the vehicles (car following and lane changing) and the resulting travel times (Figure 8.21).

8.6 Evaluation of OpenTrafficSim*:* the requirement analysis

This section evaluates the open source software project *OpenTrafficSim* by checking it on the requirements for open source software projects listed in section 3.7. This section evaluates the requirements and uses the results from case studies (NTM and N201).

8.6.1 Evaluation of the software quality

One of the key considerations during the development of the project, is to achieve a high software quality. This is a necessary requirement for (open source) software projects in order to allow users to understand, and thereafter adjust or extend the provided algorithms that for instance describe car following, lane changing or gap acceptancy. This section evaluates the software quality.

R-OpenSource.1.
A clear and well-structured design of the source code that enables users to understand and read the software

We evaluate this requirement by first reviewing the structure of the *OpenTrafficSim* project, and then explore the parts where the methods for transport and traffic simulation are codes. *OpenTrafficSim* is coded in Java and built as a collection of projects. The main projects are (https://opentrafficsim.org/index.php/getting-started):

- **dsol-base & dsol-core** (external)
  dsol is a generic event-based simulation core including many utility classes
- **djunits** (external)
  djunits allows for unit-safe calculations and programming without having to mind units
- **ots-base**
  contains some generic simulation utilities such as parameter management
- **ots-core**
  core of traffic simulation including network representation and macroscopic models
- **ots-kpi**
  stand-alone key-performance-indicator module, including trajectory sampling
- **ots-road**
  microscopic simulation of vehicular traffic
- **ots-trafficcontrol**
  event-based traffic control
- **ots-parser-xml & ots-parser-osm**
  OpenTrafficSim has several parsers to create networks from various sources, xml is the native format
- **ots-demo**
  contains demo's and should be the starting point for getting to know OpenTrafficSim

As can be seen, there are some projects that provide support for the simulation. A clear example is the *dsol* library, that is used for the traffic simulation. As a user, this library provides the methods for scheduling events, but does not require further knowledge of its internal structure.

There appears a clear structure when focussing on the projects that provide the functional classes for the traffic simulation. The *ots-core* project contains the core packages for the traffic simulation. Examples are:

- the *core.gtu*-package with for instance the GTU-class (GenericTravelUnit) that provides the basic functionality for all means of transport;
- the *core.geometry-package* with classes for lines, points and shapes and methods for spatial analyses such as computing the shortest distance between a point and a line segment;
- the *core.network-package* with basic classes for network, link and node objects.

The typical modules that regard road based traffic are placed in the **ots-road** project. This project contains packages that built upon the *core.gtu* (an object that can travel) and the *core.network*.

- The *gtu* classes refer to the behaviour of vehicles such as trip generation, route choice, path planning, car following and lane changing.

- The *network* classes describe the road objects such as a link with its characteristics in terms of lanes, alignment and lane based objects such as a sensor or traffic light.

Overall, the software provides a clear structure of project, packages and classes. This is supported by comments that describe the class, variables and methods, as well as descriptive variables names that indicate their role or function.

R-OpenSource.2.
A modular design to efficiently modify the code, for instance by replacing modules, adjusting methods or altering parameters

The *OpenTrafficSim* software project builds upon the DSOL simulation engine. The fact that DSOL is object oriented makes it easy to extend the available simulation objects in the library such as simulators, experiments and statistics, into traffic specific building blocks for *OpenTrafficSim*. In addition to this simulation engine, *OpenTrafficSim* provides a well-structured design as already has been described in the previous requirement analysis.

The classes start with core objects, that further extend into classes with more specific behaviours. Figure 8.23 shows a code example of the IDM car following principle. The figure first shows the method *followingAcceleration* to determine the car-following acceleration, possibly based on multiple leaders from the AbstractIDM class. The IDM and IDMplus class both extend the AbstractIDM class, and as the figure shows, provide two alternatives methods named *combineInteractionTerm*. This method is called in the last lines of the method *followingAcceleration*. This method contains the specific behaviour of the car following principle and is clearly separated.

As part of the N201 case study, some modifications have been tried and evaluated with respect to the parameters of driving behaviour in order to calibrate the length of the waiting rows before junctions. These modifications could be done efficiently: the parameter setting required an adjustment at only one place in the code, and provided the expected behaviour. As this example shows, it is relatively easy to replace the basic IDM module by variants with other equations. Furthermore, there is a reference in the comment section of the AbstractIDM class that refers to a url (https://en.wikipedia.org/wiki/Intelligent_driver_model) with a clear description of the intelligent driver model.

Based on these evaluations and the case study, we conclude that the code fulfils the requirements of a modular design.

AbstractIDM:

```java
public AbstractIDM(final DesiredHeadwayModel desiredHeadwayModel, final DesiredSpeedModel desiredSpeedModel)
{
    super(desiredHeadwayModel, desiredSpeedModel);
}

/**
 * Determination of car-following acceleration, possibly based on multiple leaders. This implementation calculates the IDM
 * free term, which is returned if there are no leaders. If there are leaders <tt>combineInteractionTerm()</tt> is invoked
 * to combine the free term with some implementation specific interaction term. The IDM free term is limited by a
 * deceleration of <tt>B0</tt> for cases where the current speed is above the desired speed. This method can be overridden
 * if the free term needs to be redefined.
 * @param parameters Parameters; Parameters.
 * @param speed Speed; Current speed.
 * @param desiredSpeed Speed; Desired speed.
 * @param desiredHeadway Length; Desired headway.
 * @param leaders PerceptionIterable&lt;? extends Headway&gt;; Set of leader headways (guaranteed positive) and speeds,
 *            ordered by headway (closest first).
 * @throws ParameterException If parameter exception occurs.
 * @return Car-following acceleration.
 */
@Override
@SuppressWarnings("checkstyle:designforextension")
protected Acceleration followingAcceleration(final Parameters parameters, final Speed speed, final Speed desiredSpeed,
        final Length desiredHeadway, final PerceptionIterable<? extends Headway> leaders) throws ParameterException
{
    Acceleration a = parameters.getParameter(A);
    Acceleration b0 = parameters.getParameter(B0);
    double delta = parameters.getParameter(DELTA);
    double aFree = a.si * (1 - Math.pow(speed.si / desiredSpeed.si, delta));
    // limit deceleration in free term (occurs if speed > desired speed)
    aFree = aFree > -b0.si ? aFree : -b0.si;
    // return free term if there are no leaders
    if (leaders.isEmpty())
    {
        return Acceleration.createSI(aFree);
    }
    // return combined acceleration
    return combineInteractionTerm(Acceleration.createSI(aFree), parameters, speed, desiredSpeed, desiredHeadway, leaders);
}
```

IDM:

```java
protected final Acceleration combineInteractionTerm(final Acceleration aFree, final Parameters parameters,
        final Speed speed, final Speed desiredSpeed, final Length desiredHeadway,
        final PerceptionIterable<? extends Headway> leaders) throws ParameterException
{
    Acceleration a = parameters.getParameter(A);
    Headway leader = leaders.first();
    double sRatio =
            dynamicDesiredHeadway(parameters, speed, desiredHeadway, leader.getSpeed()).si / leader.getDistance().si;
    double aInt = -a.si * sRatio * sRatio;
    return Acceleration.createSI(aFree.si + aInt);
}
```

IDMPlus:

```java
protected final Acceleration combineInteractionTerm(final Acceleration aFree, final Parameters parameters,
        final Speed speed, final Speed desiredSpeed, final Length desiredHeadway,
        final PerceptionIterable<? extends Headway> leaders) throws ParameterException
{
    Acceleration a = parameters.getParameter(A);
    Headway leader = leaders.first();
    double sRatio =
            dynamicDesiredHeadway(parameters, speed, desiredHeadway, leader.getSpeed()).si / leader.getDistance().si;
    double aInt = a.si * (1 - sRatio * sRatio);
    return new Acceleration(aInt < aFree.si ? aInt : aFree.si, AccelerationUnit.SI);
}
```

Figure 8.23: example of OpenTrafficSim code with two variants of car-following

R-OpenSource.3.
An incremental design that allows the extension of the program with new functionality in terms of modules and methods

The case studies typically show the opportunities to extend the available classes with new functionality. In the N201 case study, the basic sensor and traffic light classes were extended with additional functionality. With respect to the routing of vehicles the use of existing methods only required minor additions to implement en-route changes of routes by vehicle. Section 8.3 describes the implementation of the Network Transmission Model as an add-on to *OpenTrafficSim*. The NTM model uses code from *OpenTrafficSim* and combines it with classes that are stored in a separate package. We conclude that *OpenTrafficSim* provides an incremental design that allows for extensions.

R-OpenSource.4.
Maximal reuse and re-usability by using third party code and services.

The *OpenTrafficSim* project uses the MAVEN tool to manage dependencies. This includes a series of open source libraries from third parties. Examples are GeoTools, jUnit and PowerMock. As reported, the NTM case applies some external libraries that are included in the *OpenTrafficSim* project. The use of these tools simplifies a lot of tasks, such as the import of GIS files and the construction of graphs and computation of shortest paths. The *OpenTrafficSim* project therefore fulfils this requirement.

R- OpenSource.5
Software with a high code quality in terms of functionality, reliability, usability, efficiency, maintainability, and portability

Functionality
The software allows a simulation of highways with on- and offramps, and other types of roads with normal junctions, signalized junctions, roundabouts and priority junctions. With these functionalities, the *OpenTrafficSim* platform provides sufficient functionality for vehicle-based microsimulations.

Reliability
There are various ways to support the reliability of the software. The evaluations of the previous requirements already show a well-structured and modular design. This design supports reliability as it clearly structures the locations for coding functionality, while preventing for duplicate coding. In addition, the project uses various tools and methods to check the code. As is reported on the website, "java Generics are used to prevent flaws, such as accidental assignments of a speed value to a force, the ID of a ship to a container, etc. If at all possible we want the compiler to stop us from writing such assignments. If that really is impossible or creates very slow code, we like to see a runtime exception as early as possible." In addition, the OpenTrafficSim project uses Maven to generate reports on the software quality. Among them are reports on the coding style conventions, a copy paste detector (duplicate code detection for instance to prevent duplicate coding of methods) and a PMD source code analyser. This last tool finds common programming flaws like unused variables, empty catch blocks, unnecessary object creation, and so forth. It examines Java source code and looks for potential problems

such as possible bugs, dead code, suboptimal code and overcomplicated expressions. Finally, Jacoco is used for measuring and reporting Java code coverage. The code is tested by means of jUnit tests: a piece of code written by the developer that executes a specific method in the code to be tested and asserts a certain behaviour or state. Ope*nTrafficSim* has an overall test coverage of around 45%. The *OpenTrafficSim* applies a set of reporting tools to evaluate the quality of the software, which provides a good framework for reviewing the reliability of the code. With respect to the test coverage, an increase of the number of jUnit tests seems necessary.

Usability
The isolation of functions in separate modules allows the use of existing functionality. The possibility to extend classes in Java increases the usability, as existing classes can be used to as a basis for additional functionality. Based on the experiences from the case studies N201 and the NTM, the software proved to be useful and usable. The main issue is the absence of a graphical user interface (GUI) for editing the networks. Specifically for larger networks, this requires a rather tedious process for construction. So with respect to usability, an extension with a GUI is advised.

Efficiency
Code efficiency regards the reliability, speed and programming methodology used in developing codes for an application. Several of the previously mentioned arguments, also relate to the efficiency of the code. For example, the use of *djunits* guarantees data integrity and consistency. The supply of reusable components and external libraries are other examples for the support of efficiency. In addition, the previously mentioned reports and tests for code quality (see "Reliability") further optimize code efficiency. Examples are the use of error and exception handling. Finally, the speed of the simulation appears sufficient for the cases described in the previous sections. The evaluation with respect to the efficiency provides a positive vote.

Maintainability
The *OpenTrafficSim* software isolates pieces of code in separate components. This supports the maintainability of the code. Finished objects, such as the *dsol* simulation engine, do not need maintenance and function relatively autonomously.

Portability
The use of Java already guarantees the ability to compile into stand-alone programs that can run on any computer with a Java Runtime Environment installed, and platform independence (e.g., Windows, Mac and Linux). Therefore, *OpenTrafficSim* meets the condition of portability.

R-OpenSource.6.
A stable core platform that provides basic functionality and utilities with low variety and high re-usability

While backward compatibility is not fully guaranteed at this stage, many modules have already been in use and tested for a longer period. The development of the N201 simulation has been based on a previous beta-release (2015) of the *OpenTrafficSim* project and thus did not provide a stable release with respect to all classes and methods. As a result, the *OpenTrafficSim* project released several new beta releases. While this required some rework of the code that was specifically added for the N201 project, the overall process remained workable in terms of the amount of adjustments. As concluded previously, the absence of a GUI is the main functionality that is lacking for the evaluated version (1.01). The GUI should contain utilities for creating and editing items like the road network, traffic demand and vehicle characteristics.

8.6.2 Evaluation of the project governance

With respect to the governance of the open source software projects, two further requirements are evaluated.

R- OpenSource.7
A flexible project governance with conscious direction at the initial phase of the project and, once there is a critical mass, a movement of decision rights to those with knowledge

The *OpenTrafficSim* project is still in its initial phase. The project is currently directed by a group of developers from Delft University of Technology. Version 1.01 already provides sufficient functionality for external users, but a mature GUI is yet lacking. The development of the code until this version is written by this group. As has been presented in the previous section, there are conscious directions for the code quality and tests.

The transition towards the next phase of governance, where external users gain knowledge and experience and support further developments, requires additional actions. As mentioned, the addition of a GUI provides value added for all potential users. To second action considers the attraction of users which is further discussed in sub section 8.6.3.

R- OpenSource.8
Proper documentation of the software (preferably self-documenting)

Programming standards in the *OpenTrafficSim* project aim towards self-documenting code. This means that comments in the code are automatically translated into a structured documentation on each class and method of the *OpenTrafficSim* source code (http://opentrafficsim.org/docs/current/). The *OpenTrafficSim* project uses the Javadoc report generator in HTML. By randomly checking a number of classes, we observe that for most of these classes, Javadoc provides a description of both the class and its methods.

8.6.3 Attract sufficient users

A critical factor for open source projects is to attract a sufficient number of users that contribute knowledge. This requires:

R- OpenSource.9
The involvement of new entering members, to attract a sufficient number of users that contribute knowledge in the form of new or adjusted modules

OpenTrafficSim (version 1.01) contains sufficient functionality and quality to be used by new entering members. The source code dan be downloaded and used by everyone (https://opentrafficsim.org/index.php/getting-started). For a quick start, the demo projects provide a first glance of the traffic simulation possibilities. Yet, a basic guide to start and use the software, as well as a GUI to build and edit networks, is missing. For most new entering members support from the developers will therefore be useful and necessary for getting a deeper understanding of the software.

R- OpenSource.10

The provision of different entry levels for more and less experienced users

As has already been mentioned, the *ots-demo* project already contains a collection of demo projects that enables a user to retrieve visual insight in the typical driving behaviours, such as merging, car following and conflicts at junctions. The demos provide a simple GUI and allow the user to start a simulation, with the option to choose different scenario's and change some of the parameters. This already enables users to gain visual insight in the simulation with the current functions.

8.6.4 Summary

For most requirements the evaluation shows a positive result. Overall, the software provides a clear structure of projects, packages and classes. Based on these evaluations and the case study, we conclude that the code fulfils the requirements of a modular design, and provides an incremental design that allows for extensions. Moreover, *OpenTrafficSim* re-uses knowledge by using third party code and services. With respect to reliability, we notice a low coverage of jUnit tests that requires improvement.

The absence of a GUI is the main functionality that is lacking for the evaluated version (1.01). The GUI should contain utilities for creating and editing items like the road network, traffic demand and vehicle characteristics. This may hamper the attraction of newly entering users.

To improve this situation the following options could support the involvement of new users:

- Applying the *OpenTrafficSim* software for educational purposes, for instance by introducing it in the curriculum. The software could be typically used for teaching students the ins and outs of car following and lane changing models.
- Promoting the software by organizing dedicated conferences. The current software has sufficient mass for external users.
- Development of a graphical user interface that enables users to graphically create, edit and store networks.

Chapter 9

Conclusion and recommendations

9.1 Introduction

Software packages for the modelling of transport and traffic have found its way since the end of the previous century. While academia and research institutions often initiated these packages, their further development has often been taken over by private companies. In almost all cases, their software is proprietary and closed, which usually implies that the source code is not freely available. As its consequence, the underlying knowledge with respect to methods and algorithms is largely inaccessible. In absence of an open source modelling platform, the alternative for researchers is to code transport and traffic models (or parts thereof) from scratch. As re-use of the code is rarely a requirement, there is hardly any attention for the design and quality of the software. This prevents others to use and learn from the software, and hampers innovative research. A second issue is a lack of data standardization. All model packages apply their own data definitions and design which hampers the exchange of data between models. This requires an internationally accepted standard for geospatial data within the transportation domain that is currently lacking.

OpenTrafficSim is an example of a state-of-the-art open source traffic simulation modelling platform, which has been initiated by Delft University of Technology. This platform makes it possible to extend microscopic models with explanatory mental models, such that new behavioural theories can be tested and shared with the research community (van Lint, Schakel et al. (2016, Verbraeck (2017)). The basic idea of this open source software platform is to provide knowledge on methods and algorithms by means of well-documented programming code, and allows users to alter and extend it by adding new code. First of all, this enables researchers to concentrate on science, rather than repeatedly starting from scratch with designing and coding much common and auxiliary functionality required for a model to run. But perhaps the most important aspect is the opportunity to use the software as a basis for new and innovative developments. If these developments are implemented in the existing software environment, the innovations can be directly open to other users.

This thesis provides a design for the objects of the transport and traffic infrastructure, defined as the *transportation infrastructure, including the connections with the activity locations, and including its relationship with the means of transport services*. As reported in chapter 7, a significant part of this design has been developed by the *OpenTrafficSim* project-team and is implemented as one of the building blocks of *OpenTrafficSim*. The infrastructural objects that should be included in a transport and traffic model system are identified by reviewing the main steps of this model system, from activity generation towards the execution of trips. Specific attention is paid to the scalability and various levels of detail, the various modes of traffic, and the exchange of transport and traffic in- and output with external data sources.

9.2 Answer to the research questions

The first research objective aims to *derive the key requirements for the development of a successful open source software project for transport and traffic*. What is the best approach to create a transport and traffic model that enables model users and developers to reuse knowledge from that transport and traffic model environment, and/or extend it with new and innovative functionality? To answer this question two sub questions have been introduced:

Q1.1: What are the requirements and demands for an open source software modelling environment for transport and traffic?

Q1.2: How to organize this open source software project in terms of software development and governance structure to optimally support the (re-)usability of the system?

This topic is introduced in section 2.5 and elaborated in chapter 3. With listed requirements for three critical topics: software quality, project governance and the attraction of sufficient users. In chapter 8 these requirements are used to evaluate the *OpenTrafficSim* project. We present two case studies (section 8.3 and 8.4) that applied the *OpenTrafficSim* software. In both cases the core software has been extended with additional functionality. The experiences with *OpenTrafficSim* in these case studies are evaluated in section 8.5. Section 8.6 further evaluates *OpenTrafficSim* and shows that most of the requirements with respect to the software quality are met. With respect to the project governance the requirements are met. The last topic, the attraction of sufficient users, appears the main challenge for making *OpenTrafficSim* a successful open source software project.

The second research question focusses on the design of the infrastructure in transport and traffic models. *What is a good design of model objects for the transportation infrastructure of road based traffic, in order to fulfil the functional requirements and enable a proper data-exchange with other models and external data sources?*
To answer this question, three sub-questions have been raised*:*

Q2.1: What are the global requirements for the design of the transportation infrastructure when regarding the functional demands from the varying transport and traffic modelling approaches?

This question is discussed in *chapter 4* and concludes with an overview of the main components of transport and traffic models that are relevant for the design of the road infrastructure.

Q2.2: What are specific requirements for a design of the traffic infrastructure that enables a transition between global and detailed representations of a road traffic infrastructure and allows the modelling of all modes of traffic on this infrastructure.

Chapter 5 further explores the specific requirements for the design of a multi-scale transport infrastructure. This results in a series of requirements on three topics: (1) the design of the infrastructure to model vehicle operations and interactions, (2) a valid topology for determining routes and costs and (3) to allow various levels of detail. Section 5.6 presents an overview of the requirements.

Q2.3: What are the requirements to achieve a proper exchange of data between the transport and traffic model and external data sources, with a focus on the road based transportation infrastructure?

Models require data input to actually shape the model objects. Oppositely, models produce information that can be used as input for external databases again. This mutual exchange of data between models and geo-spatial databases requires a proper mapping of data objects at various levels of detail. In chapter 6 we analyse the requirements for geospatial databases to allow a proper data exchange with transport and traffic models. The main issues regard (1) a matching of data objects between models and external databases, (2) a mature data and complete GIS standard for the transport infrastructure, and finally (3) a distinction in levels of detail. These requirements are partly met by the current geospatial data-standards and require a further extension. This is elaborated in section 6.6.

In chapter 7 we present a design for the objects of the road infrastructure that meet the requirements from chapter 4, 5 and 6. As a large part of the objects is already implemented in *OpenTrafficSim*, two of the case studies in chapter 8 show its capability to model the infrastructure in both very detailed microsimulation models as well as a very coarse approach.

As a result, we answer all the research questions and provide:

- The requirements to create an open and free to code transport and traffic model environment
- The design of the data objects and structures that allow both global and detailed representations of a road traffic infrastructure within a transport and traffic model environment
- The conditions and developments that are required to create an efficient data-exchange between transport and traffic models, and with external data-sources

9.3 Research findings and conclusions

The first result of this research is an overview of the requirements to create a successful open source software project for research and education, that enables model users and developers to reuse knowledge from that model environment, and/or extend it with new and innovative functionality. This exploration considers both the desired functionality by globally reviewing the main modelling steps and the various methodological approaches, and the requirements in terms of quality of the software (design) and governance. As the evaluation shows, the open source software approach offers good prospects, but requires relatively high demands for the software, both in terms of quality and manageability. The main difference with closed software is that contributors operate at different organizations. This requires clear appointments about the coding with regards to design issues such as the structure of classes, objects and methods, documentation and code-style. Automated and self-documenting programming approaches, and a clear modular design, support new and unexperienced users to learn and understand the software, without the involvement of an experienced user.

In addition to these requirements the software should be usable for external users. To attract users at the initial stage, there needs to be a critical mass of functionality, combined with sufficient utility tools to make the software attractive to work with. The governance of the open source software projects at the start of this project, requires a project team that develops a first release with sufficient mass. Later on the governance may change, when other users become active and get involved in decisions about the project. As a result we provide a list of requirements for obtaining a successful open source software project, and used it to evaluate the *OpenTrafficSim* project.

The actual creation of a transport and traffic modelling platform requires a thorough preparation of the design of both the methods and the related object definitions. This thesis only concerns the design of the model objects that represent the transportation infrastructure. In order to distil and define the relevant objects in an integral modelling environment, we analyse the requirements with respect to the infrastructure for all phases of the transport and traffic modelling process. Chapter 4 describes the main stages of transport and traffic models, and provides an overview of the core objects of the infrastructure. In addition, chapter 5 explores the requirements for the representation of the infrastructure within a modelling environment that allows to represent multiple scales in terms of level of detail and handle multiple modes of transport. A proper data model that represents the most detailed level that is necessary for transport and traffic models, is the first requirement to enable multiple levels of detail. From this base level, the challenge is to generate coarser levels of detail by aggregation and generalization procedures.

The elaboration of generalization and aggregation procedures of the objects into less detailed representations, depends on the modelling requirements of coarser approaches. With respect to the modelling of traffic and transport *demand*, the individuals trips can be generalized by grouping of individual activity locations into traffic zones. For the assignment of transport and traffic, the mesoscopic and macroscopic approaches require significantly less details for the representation of the transportation infrastructure. This is mainly due to the absence of modelling the lane changing and car following behaviour. These behaviours are replaced by other methods to derive the road and lane capacity. This analytical derivation transforms the impact of discrete events into continuous formulae that use flow as an input. These functions typically require additional link-attributes, such as the number of lanes, function of the road, capacity, maximum traffic density, and the legal (or estimated) maximum speed. To *exchange* input data between these *levels of detail*, we first have to check if an attribute of the transport infrastructure at the lower levels of detail contains a behavioural component. If not, the attributes, such as the link length and legal speed limits, can be directly derived from the link and lane attributes of the detailed simulation network.

Finally, a further refinement of the level of detail, for instance to create an environment for driving simulators or to model (lateral) movements with more detail, appears possible by means of the principle of *extensibility*. This allows an extension of objects with new attributes and functions, for instance to refine the surface of a road by a curved regular grid.

This thesis presents a novel design for the transportation infrastructure that provides (1) the ability to create routes, (2) the representation of a valid and consistent topology, (3) objects for all road related modes (including trams), and (4) sufficient functional information to represent vehicle and driving behaviour (road description, including traffic management systems). The network is essentially based on a graph of links and nodes. The details of the road are defined in the cross section elements that are part of a link. It includes both the area where traffic moves or parks, and auxiliary objects that describe the further characteristics of the road such as a sidewalk, barrier and berm. A junction is composed by internal (virtual) links that connect the incoming and egress lanes, and explicitly define the allowed turning movements. In addition, the functional elements such as traffic light controllers and car parks are connected with the road network. These functional units use devices, such as sensors or traffic lights, to inform, measure and control traffic.

In order to connect trips with the network, the traffic zone is defined as an area that encompasses a collection of activity locations. At the highest level of detail, every activity location can be connected with the 'boarding' location for the first mode (car park, bus-stop, etc.) by means of a walk link. The combination of a graph together with the details of the road topology, results in a dual system that resembles both the requirements for microsimulation (detailed lane description) and the meso- and macroscopic (link based graph) network representations.

To efficiently build transport and traffic models, a smooth exchange of data between models and databases is essential. As part of the previously described design process, the reconciliation of model objects and their counterpart in data sources has therefor been an important topic (chapter 6). A straightforward relation between the modelled objects and external data is desirable, which requires that model objects align to the objects from standardized data models. This includes not only objects of the road infrastructure, but also the connection to activity locations, and infrastructure of other modes such as pedestrian (space) and transfer locations (public transport). A review of existing (open) data standards shows that CityGML appears a suitable choice as it (1) covers the complete urban and rural environment which supports topological consistency, and (2) distinguishes various levels of detail. Yet, to make it fully applicable for transport and traffic models, a further elaboration of the transportation part appears necessary. This thesis suggests an extension of CityGML to create a comprehensive description of the road and junction lay-out at various levels of detail, that is currently lacking.

9.4 Practical implications

This thesis research concentrates on the design of data objects that enables a representation of the transport infrastructure for transport and traffic models. We conclude that specifically for microsimulation models, a high level of detail is required for describing the modelled objects. This demand for a high level of detail does not only count for transport and traffic models. There is a growing interest from other fields, such as ITS services, with regards to high quality digital road maps. The increasing role and availability of data from mobile devices, with detailed information about the location of moving objects, requires maps with a higher level of detail and accuracy to match their actual position at a lane specific level (Neuhold, Haberl et al. (2017)). As this research shows, the current (open) data standards are not fully elaborated to enable a full representation of the map objects that are required for transport and traffic modelling at such a high level of detail. Specifically, data-objects of lane details and markings, as well as a description of the junction area with the connectivity between incoming and egress lanes, are lacking. As was concluded in the previous section this requires a further elaboration of the data standard on these specific topics. A second notion is that while many of the currently available networks, such as OpenStreetMap and other digital maps, do possess quite an amount of information with respect to the transport and traffic infrastructure, they also lack detailed information about junctions and lane configurations. This has been clearly illustrated in the case study in section 8.2. Thirdly we notice that many of these data sources do not align with the open data standards. OpenStreetMap for instance has quite a different data structure than the existing open data standards. The same accounts for other maps such as the Dutch road map (NWB). This raises the question how the digital maps should be further developed. A first step would be to compare and evaluate the data structures both for the data-standards and the digital maps, and investigate how their developments can be aligned to bridge the gaps.

A second practical recommendation regards the education on software. Specifically for students that are working on methods and modelling, a basic level of knowledge and skills with regards to software engineering would be beneficial. This enables students to develop code at such a quality level that it can be shared with and used by other developers. Moreover, such knowledge is useful to explore the available code from open source software projects, which helps to gain deeper insight into the details of transport and traffic modelling methods and algorithms.

9.5 Reflection and further research

This thesis has provided answers towards the design of an open software transport and traffic modelling platform. On the one hand by elaborating the requirements with respect to the software and governance of open source projects, and also by proposing a design for the objects that are required to create transport and traffic models with various scales and levels of detail. This information has been used for the development of *OpenTrafficSim*. This research has also led to new questions and paths of thought that did not fit in the scope or time constraints of the research. These are presented in this section as recommendations for future research.

An automated conversion of link and node attributes between levels of detail, appears possible for many attributes, excluding the variables whose value is impacted by the behaviour of a driver and vehicle. Another issue regards the fact that not all roads from a fully specified network (within a geo-database) are required for specific transport and traffic model applications. Generally, there is a higher density of both traffic analysis zones (TAZ) and roads in the study area, and a diminishing density at increasing distances from this study-area. An automated generalization process to create subsets of networks from a central database would be beneficially to create:

- A subset of roads from the geo-database at various levels of detail, and
- This subset of roads need not be evenly distributed over the total area.

Based on literature, there appear separate approaches from the field of cartography and geographic information systems, and the field of transport and traffic modelling. Further research, that combines the viewpoints from both approaches, may be worthwhile to enhance the quality of procedures to generalize networks from very detailed levels towards coarser descriptions. From the perspective of transport and traffic models, the required coarseness (or density) of a network depends on the combination of the modelling approach (a simulation has other requirements than a static assignment), the definition of the traffic zones and the characteristics of the road network (for instance urban or rural). We recommend an approach that combines these properties, with the knowledge and experiences from the domain of geographic information and analysis, to explore the opportunities to create maps for transport and traffic models at various levels of scale.

With respect to the exchange of data between transport and traffic models and databases, this research has shown that there is no open data standard with the required specifications to cover all requirements from transport and traffic models at a high level of detail. Specifically with respect to the description of the lane and junction topology, their appears lack of detail in the data standards (CityGML and GDF 5.0). Other standards such as OpenDRIVE, MAP, or proposals for the extension of CityGML, already provide suggestions for implementing these details. Further research on this topic would be beneficial for a further elaboration towards a mature standard for transportation and traffic. In addition to this theme, we have noticed that many digital road maps, such as OpenStreetMap or the Dutch Road map (NWB) do not align to any of these standards. Moreover, there appear differences between these maps, in terms of both differences in object design as well as their topology. While location referencing is used as an option to tackle the problem of accurately matching locations between dissimilar digital maps (see for instance Ebendt and Tcheumadjeu (2017)), another more structural solution to the problem is to diminish dissimilarities between maps. Research on this topic regards the process of matching both the design and definition of the objects, as well as the topographic and topological quality and requirements. In addition, the topic of joining maps, for instance for cycle paths and roads can be interesting to create an integrated map for various modes.

Bibliography

Abras, C., D. Maloney-Krichmar and J. Preece (2004). "User-centered design." Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications **37**(4): 445-456.

Arasan, V. T. and P. Vedagiri (2008). "Bus priority on roads carrying heterogeneous traffic: A study using computer simulation." European Journal of Transport and Infrastructure Research **8**(1): 45-64.

Arentze, T. and H. Timmermans (2000). Albatross: a learning based transportation oriented simulation system, Eirass Eindhoven.

Arentze, T. and H. Timmermans (2006). "Social networks, social interactions, and activity-travel behavior: a framework for microsimulation."

Arentze, T., H. Timmermans and D. Janssens (2006). Modeling short-term dynamics in activity-travel patterns: from aurora to feathers. TRB.

Auld, J., M. Hope, H. Ley, V. Sokolov, B. Xu and K. Zhang (2016). "POLARIS: Agent-based modeling framework development and implementation for integrated travel demand and network and operations simulations." Transportation Research Part C: Emerging Technologies **64**: 101-116.

Auld, J. and A. Mohammadian (2009). "Framework for the development of the agent-based dynamic activity planning and travel scheduling (ADAPTS) model." Transportation Letters **1**(3): 245-255.

Auld, J., A. Mohammadian and S. T. Doherty (2009). "Modeling activity conflict resolution strategies using scheduling process data." Transportation Research Part A: Policy and Practice **43**(4): 386-400.

Bains, M. S., B. Ponnu and S. S. Arkatkar (2012). "Modeling of Traffic Flow on Indian Expressways using Simulation Technique." Procedia - Social and Behavioral Sciences **43**(0): 475-493.

Baldwin, C. Y. and J. Woodard (2008). "The Architecture of Platforms: A Unified View." SSRN eLibrary.

Balmer, M. (2007). Travel demand modeling for multi-agent transport simulations: algorithms and systems, ETH.

Balmer, M., K. W. Axhausen and K. Nagel (2006). An Agent-Based Demand-Modeling Framework for Large Scale Micro-Simulations TRB.

Balmer, M., M. Rieser, K. Meister, D. Charypar, N. Lefebvre and K. Nagel (2009). "MATSim-T: Architecture and Simulation Times."

Barceló, J. (2010). Models, Traffic Models, Simulation, and Traffic Simulation. Fundamentals of Traffic Simulation. F. S. Hillier, Springer New York. **145:** 1-62.

Barrett, C., K. Berkbigler, L. Smith, V. Loose, R. Beckman, J. Davis, D. Roberts and M. Williams (1995). An operational description of TRANSIMS.

Bates J, B. M., Hanson P, McDonald D and Simmonds D Building a Strategic Model for Edinburgh. PTRC I9th Summer Annual Meeting.

Becker, T., C. Nagel and T. H. Kolbe (2011). Integrated 3D modeling of multi-utility networks and their interdependencies for critical infrastructure analysis. Advances in 3D Geo-Information Sciences, Springer**:** 1-20.

Becker, T., C. Nagel and T. H. Kolbe (2013). Semantic 3D modeling of multi-utility networks in cities for analysis and 3D visualization. Progress and New Trends in 3D Geoinformation Sciences, Springer**:** 41-62.

Beil, C. (2017). Detaillierte Repräsentation des Straßenraums in 3D-Stadtmodellen.

Bella, F. (2013). "Driver perception of roadside configurations on two-lane rural roads: Effects on speed and lateral placement." Accident Analysis & Prevention **50**: 251-262.

Bellemans, T., B. Kochan, D. Janssens, G. Wets, T. Arentze and H. Timmermans (2010). "Implementation framework and development trajectory of FEATHERS activity-based simulation platform." Transportation Research Record: Journal of the Transportation Research Board(2175): 111-119.

Ben-Akiva, M., J. Bottom, S. Gao, H. N. Koutsopoulos and Y. Wen (2007). "Towards disaggregate dynamic travel forecasting models." Tsinghua Science & Technology **12**(2): 115-130.

Ben-Akiva, M. E., S. Gao, Z. Wei and Y. Wen "A Dynamic Traffic Assignment Model for Highly Congested Urban Networks." Transportation Research Part C: Emerging Technologies **24**: 62-82.

Benenson, I., K. Martens and S. Birfir (2008). "PARKAGENT: An agent-based model of parking in the city." Computers, Environment and Urban Systems **32**(6): 431-439.

Bliemer, M., M. Raadsen, L. Brederode, M. Bell, L. Wismans and M. Smith (2015). Genetics of traffic assignment models for strategic transport planning. Australasian Transport Research Forum (ATRF), 37th, 2015, Sydney, New South Wales, Australia.

Bonsall, P., R. Liu and W. Young (2005). "Modelling safety-related driving behaviour—impact of parameter values." Transportation Research Part A: Policy and Practice **39**(5): 425-444.

Boxill, S. A. and L. Yu (2000). "An Evaluation of Traffic Simulation Models for Supporting ITS."

Bullock, D., B. Johnson, R. B. Wells, M. Kyte and Z. Li (2004). "Hardware-in-the-loop simulation." Transportation Research Part C: Emerging Technologies **12**(1): 73-89.

Burghout, W. (2005). Hybrid microscopic-mesoscopic traffic simulation. PhD, Royal Institute of Technology.

Burghout, W. (2005). "Mesoscopic simulation models for short-term prediction." PREDIKT project report CTR2005 **3**.

Cagdas, V. (2007). "A Base Research about GDF."

Campanella, M. C. (2016). Microscopic modelling of walking behaviour. doctoral thesis, Delft Universtiy of Technology.

Capra, E., C. Francalanci and F. Merlo (2008). "An Empirical Study on the Relationship Between Software Design Quality, Development Effort and Governance in Open Source Projects." Software Engineering, IEEE Transactions on **34**(6): 765-782.

Cats, O., J. West and J. Eliasson (2016). "A dynamic stochastic model for evaluating congestion and crowding effects in transit systems." Transportation Research Part B: Methodological **89**: 43-57.

Chandra, S. and U. Kumar (2003). "Effect of lane width on capacity under mixed traffic conditions in India." Journal of transportation engineering **129**(2): 155-160.

Chitturi, M. and R. Benekohal (2005). "Effect of lane width on speeds of cars and heavy vehicles in work zones." Transportation Research Record: Journal of the Transportation Research Board(1920): 41-48.

Ciari, F., M. Balmer and K. W. Axhausen (2008). A new mode choice model for a multi-agent transport simulation. STRC. Ascona.

Consortium, O. G. (2012). OpenGIS® City Geography Markup Language (CityGML) Encoding Standard T. H. K. Gerhard Gröger, Claus Nagel, Karl-Heinz Häfele.

Cruzado, I. and E. T. Donnell (2009). "Factors affecting driver speed choice along two-lane rural highway transition zones." Journal of Transportation Engineering **136**(8): 755-764.

Czerwionka, P. and M. Wang (2011). "Optimized Route Network Graph as Map Reference for Autonomous Cars Operating on German Autobahn."

da Cruz Vieira, W. and L. H. S. de Lelis (2015). "PROGRAMMING LANGUAGES IN ECONOMICS: A COMPARISON AMONG FORTRAN77, C++, AND JAVA." Revista de Economia e Agronegócio–REA **3**(3).

Daamen, W. and S. Hoogendoorn (2003). "Experimental research of pedestrian walking behavior." Transportation Research Record: Journal of the Transportation Research Board(1828): 20-30.

De Boer, A. and F. Penninga (2017). Samenhang NWB, BGT, BRT en BAG, Rijkswaterstaat WVL.

de Dios Ortuzar, J. and L. G. Willumsen (2011). "Modelling Transport."

Delp, S. L., F. C. Anderson, A. S. Arnold, P. Loan, A. Habib, C. T. John, E. Guendelman and D. G. Thelen (2007). "OpenSim: Open-Source Software to Create and Analyze Dynamic Simulations of Movement." Biomedical Engineering, IEEE Transactions on **54**(11): 1940-1950.

Di Tullio, D. and D. S. Staples (2013). "The Governance and Control of Open Source Software Projects." Journal of Management Information Systems **30**(3): 49-80.

Doherty, S. (2006). "Should we abandon activity type analysis? Redefining activities by their salient attributes." Transportation **33**(6): 517-536.

Duives, D. C., W. Daamen and S. P. Hoogendoorn (2013). "State-of-the-art crowd motion simulation models." Transportation research part C: emerging technologies **37**: 193-209.

Dupuis, M. (2015). OpenDRIVE Format Specification, Rev. 1.4. VIRES Simulationstechnologie GmbH.

Dwarampudi, V., S. S. Dhillon, J. Shah, N. J. Sebastian and N. Kanigicharla (2010). "Comparative study of the Pros and Cons of Programming languages Java, Scala, C++, Haskell, VB. NET, AspectJ, Perl, Ruby, PHP & Scheme-a Team 11 COMP6411-S10 Term Report." arXiv preprint arXiv:1008.3431.

Ebendt, R. and L. C. T. Tcheumadjeu (2017). "An approach to geometry-based dynamic location referencing." European Transport Research Review **9**(3): 38.

Edie, L. C. (1963). Discussion of traffic stream measurements and definitions, Port of New York Authority.

Farah, H. and T. Toledo (2010). "Passing behavior on two-lane highways." Transportation research part F: traffic psychology and behaviour **13**(6): 355-364.

Fellendorf, M. and P. Vortisch (2010). Microscopic traffic flow simulator VISSIM. Fundamentals of traffic simulation, Springer: 63-93.

Ferrett, L. K. and J. Offutt (2002). An empirical comparison of modularity of procedural and object-oriented software. Engineering of Complex Computer Systems, 2002. Proceedings. Eighth IEEE International Conference on, IEEE.

Florian, M., M. Mahut and N. Tremblay (2008). "Application of a simulation-based dynamic traffic assignment model." European Journal of Operational Research **189**(3): 1381-1392.

Gao, S., E. Frejinger and M. Ben-Akiva (2010). "Adaptive route choices in risky traffic networks: A prospect theory approach." Transportation Research Part C: Emerging Technologies **18**(5): 727-740.

Gentile, G., M. Florian, Y. Hamdouch, O. Cats and A. Nuzzolo (2016). The Theory of Transit Assignment: Basic Modelling Frameworks. Modelling Public Transport Passenger Flows in the Era of Intelligent Transport Systems, Springer: 287-386.

Gerlough, D. L. I. o. T. and E. Traffic (1956). Simulation of freeway traffic by an electronic computer.

Geroliminis, N. (2015). "Cruising-for-parking in congested cities with an MFD representation." Economics of Transportation **4**(3): 156-165.

Geroliminis, N. and C. F. Daganzo (2007). Macroscopic modeling of traffic in cities. TRB.

Glickstein, A. and S. L. Levy (1961). Application of digital simulation techniques to highway design problems. Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference. Los Angeles, California, ACM: 39-50.

Goebel, N., A. Skuballa, M. Mauve and K. Graffi (2016). "Aggregation and Map-Matching of Mobile Cellular Network Traces."

Goode, H. H., Pollmar, C. H., Wright, J.B. (1956). The use of a digital computer to model a signalized intersection. Proc. Highway Res. Board.

Goodell, H. (1997). End-user computing. CHI '97 Extended Abstracts on Human Factors in Computing Systems. Atlanta, Georgia, ACM: 132-132.

Gunduz, M., U. Isikdag and M. Basaranera (2016). "a Review of Recent Research in Indoor Modelling & Mapping." ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences: 289-294.

Gupta, S. and P. Vovsha (2013). "A model for work activity schedules with synchronization for multiple-worker households." Transportation **40**(4): 827-845.

Hackney, J. (2009). Integration of social networks in a large-scale travel behavior microsimulation, Massachusetts Institute of Technology.

Haefliger, S., G. von Krogh and S. Spaeth (2008). "Code Reuse in Open Source Software." Management science **54**(1): 180-193.

Han, A., Timmermans, Janssens and Wets (2009). Developing Dynamic Models of Activity-Travel Behavior: Principles, Mechanisms, Challenges in Data Collection and Methodological Issues. TRB.

Handford, D. and A. Rogers (2011). Modelling driver interdependent behaviour in agent-based traffic simulations for disaster management. Advances on Practical Applications of Agents and Multiagent Systems, Springer: 163-172.

Hardy, P., M. Hayles and P. Revell (2003). Clarity-a new environment for generalisation using agents, Java, XML and topology. Fifth workshop on progress in automated map generalization.

Henk, R., A. Ballard, R. Robideau, W. Peacock, P. Maghelal, M. Lindell, C. Prater, L. Loftus-Otway, P. Siegesmund and R. Harrison (2007). "Disaster preparedness in Texas." College Station, Texas. Technical Memorandum for the Texas Department of Transportation, Texas Transportation Institute.

Henkens, N. C. and G. Tamminga (2015). Capaciteitswaarden Infrastructuur Autosnelwegen. D. M. o. I. a. t. Environment.

Hochmair, H. H., D. Zielstra and P. Neis (2015). "Assessing the completeness of bicycle trail and lane features in OpenStreetMap for the United States." Transactions in GIS **19**(1): 63-81.

Hong, L., W. Jianping, C. Man and F. Tao (2008). A standardized data model for data sharing of GIS and microscopic traffic simulation system. Service Operations and Logistics, and Informatics, 2008. IEEE/SOLI 2008. IEEE International Conference on.

Hoogendoorn, R. G. (2010). State Of The Art Driving Theory And Modeling In Case Of Exceptional Conditions. Delft, Technical University of Delft.

Hoogendoorn, S. and V. Knoop (2012). Traffic flow theory and modelling, Edward Elgar Publishing Limited.

Hoogendoorn, S. P. (2001). "State-of-the-art of vehicular traffic flow modelling." Journal of Systems and Control Engineering **Vol 215 No 14**.

Hoogendoorn, S. P. and P. H. Bovy (2004). "Pedestrian route-choice and activity scheduling theory and models." Transportation Research Part B: Methodological **38**(2): 169-190.

Hoogendoorn, S. P., F. van Wageningen-Kessels, W. Daamen, D. C. Duives and M. Sarvi (2015). "Continuum theory for pedestrian traffic flow: Local route choice modelling and its implications." Transportation Research Procedia **7**: 381-397.

Horni, A., K. Nagel and K. W. Axhausen (2016). "The multi-agent transport simulation MATSim." Ubiquity, London **9**.

Howison, J. and J. D. Herbsleb (2011). Scientific software production: incentives and collaboration. Proceedings of the ACM 2011 conference on Computer supported cooperative work. Hangzhou, China, ACM.

Hunt, J. D. and K. J. Stefan (2007). "Tour-based microsimulation of urban commercial movements." Transportation Research Part B: Methodological **41**(9): 981-1013.

Jacobs, P. H. (2005). The DSOL simulation suite, TU Delft, Delft University of Technology.

Jacobs, P. H., N. A. Lang and A. Verbraeck (2002). D-SOL; a distributed Java based discrete event simulation architecture. Simulation Conference, 2002. Proceedings of the Winter, IEEE.

Jayakrishnan, R. and H. O. Kim, Jun-Seok (2006). Activity Chaining Model Incorporating Time Use Problem and Its Application to Network Demand Analysis

TRB.

Jiang, T., M. Miska, M. Kuwahara, A. Nakasone and H. Prendinger (2010). "Microscopic Simulation for virtual worlds with self-driving avatars."

Katoshevski, R., I. Glickman, R. Ishaq and Y. Shiftan (2015). "Integrating activity-based travel-demand models with land-use and other long-term lifestyle decisions." Journal of Transport and Land Use **8**(3).

Katz, J. H. (1963). "Simulation of a traffic network." Commun. ACM **6**(8): 480-486.

Kesting, A., R. Germ, M. Budden and M. Treiber. "Multi-model Open-source Vehicular Simulator." from www.movsim.org.

Kesting, A., M. Treiber and D. Helbing (2008). Agents for Traffic Simulation.

Kilpelainen, T. (1997). "Multiple representation and generalization of geo-databases for topographic maps." NASA(19980218655).

Knoop, V., D. De Jong and S. Hoogendoorn (2014). "Influence of Road Layout on Network Fundamental Diagram." Transportation Research Record: Journal of the Transportation Research Board(2421): 22-30.

Knoop, V. and S. Hoogendoorn (2013). "Empirics of a generalized macroscopic fundamental diagram for urban freeways." Transportation Research Record: Journal of the Transportation Research Board(2391): 133-141.

Knoop, V., G. Tamminga and L. Leclercq (2016). Network Transmission Model: Application to a Real World City. Transportation Research Board 95th Annual Meeting.

Knoop, V. L., A. Duret, C. Buisson and B. Van Arem (2010). Lane distribution of traffic near merging zones influence of variable speed limits. Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on, IEEE.

Knoop, V. L., S. Hoogendoorn and J. Van Lint (2013). The impact of traffic dynamics on macroscopic fundamental diagram. 92nd Annual Meeting Transportation Research Board, Washington, USA, 13-17 January 2013; Authors version, Transportation Research Board.

Knoop, V. L. and S. P. Hoogendoorn (2015). Simulation Model for Traffic Using Network Fundamental Diagrams. Traffic and Granular Flow'13, Springer: 379-384.

Knoop, V. L., H. van Lint and S. P. Hoogendoorn (2015). "Traffic dynamics: Its impact on the macroscopic fundamental diagram." Physica A: Statistical Mechanics and its Applications **438**: 236-250.

Krajzewicz, D., M. Bonert and P. Wagner (2006). The Open Source Traffic Simulation Package SUMO.

Krajzewicz, D., J. Erdmann, M. Behrisch and L. Bieker (2012). "Recent development and applications of SUMO–simulation of urban mobility." International Journal On Advances in Systems and Measurements **5**(3 and 4): 128-138.

Labetski, A., S. van Gerwen, G. Tamminga, H. Ledoux, J. Stoter, A. Labetski, S. van Gerwen, G. Tamminga, H. Ledoux and J. Stoter (2018). "A proposal for an improved transportation model in CityGML." International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences **42**.

Lakhani, K. R. and E. von Hippel (2003). "How open source software works: "free" user-to-user assistance." Research Policy **32**(6): 923-943.

Langerudi, M. F., M. Javanmardi, R. Shabanpour, T. H. Rashidi and A. Mohammadian (2017). "Incorporating in-home activities in ADAPTS activity-based framework: A sequential conditional probability approach." Journal of Transport Geography **61**: 48-60.

Langlois, R. N. and G. Garzarelli (2008). "Of Hackers and Hairdressers: Modularity and the Organizational Economics of Open-source Collaboration." Industry & Innovation **15**(2): 125 - 143.

Laurini, R. (2012). Importance of spatial relationships for geographic ontologies. Proceedings of the Seventh International Conference on Informatics and Urban and Regional Planning INPUT.

Laval, J. A. and F. Castrillón (2015). "Stochastic approximations for the macroscopic fundamental diagram of urban networks." Transportation Research Procedia **7**: 615-630.

Laval, J. A. and F. Castrillón (2015). "Stochastic approximations for the macroscopic fundamental diagram of urban networks." Transportation Research Part B: Methodological **81**: 904-916.

Leclercq, L., F. Marczak, V. L. Knoop and S. P. Hoogendoorn (2016). "Capacity drops at merges: analytical expressions for multilane freeways." Transportation Research Record: Journal of the Transportation Research Board(2560): 1-9.

Leclercq, L., C. Parzani, V. L. Knoop, J. Amourette and S. P. Hoogendoorn (2015). "Macroscopic traffic dynamics with heterogeneous route patterns." Transportation Research Procedia **7**: 631-650.

Lee, K. S., J. K. Eom and D.-s. Moon (2014). "Applications of TRANSIMS in Transportation: A Literature Review." Procedia Computer Science **32**(0): 769-773.

Lerner, J. and J. Tirole (2002). "Some Simple Economics of Open Source." The Journal of Industrial Economics **50**(2): 197-234.

Lewis-Evans, B. and S. G. Charlton (2006). "Explicit and implicit processes in behavioural adaptation to road width." Accident Analysis & Prevention **38**(3): 610-617.

Li Zhang , Y. W., Zhitong Huang and Mingzhou Jin (2010). Framework of calculating the measures of resilience (MOR) for intermodal transportation systems, Mississippi State University.

Liao, F., T. Arentze and H. Timmermans (2014). "Multi-state supernetworks: recent progress and prospects." Journal of Traffic and Transportation Engineering (English Edition) **1**(1): 13-27.

Liu, R., T. May and S. Shepherd (2011). "On the fundamental diagram and supply curves for congested urban networks." Transportation Research Part A: Policy and Practice **45**(9): 951-965.

Lorenz, B., H. J. Ohlbach and L. Yang (2005). "Ontology of transportation networks."

Macal, C. M. and M. J. North (2009). Agent-based modeling and simulation 2009 Winter Simulation Conference M. .

MacCormack, A., J. Rusnak and C. Y. Baldwin (2007). "The Impact of Component Modularity on Design Evolution: Evidence from the Software Industry." SSRN eLibrary.

Mahut, M. and M. Florian (2010). Traffic Simulation with Dynameq. Fundamentals of Traffic Simulation. F. S. Hillier, Springer New York. **145:** 323-361.

Marczak, F. and C. Buisson (2015). "A Macroscopic Model for Freeway Weaving Sections." Computer-Aided Civil and Infrastructure Engineering **30**(6): 464-477.

Martens, K., I. Benenson and N. Levy (2009). "The Dilemma of On-street Parking Policy: Exploring Cruising for Parking Using an Agent-based Model."

Meister, K., M. Balmer, K. W. Axhausen and K. Nagel (2006). Planomat: a comprehensive scheduler for a large-scale multi-agent transportation simulation. STRC. Ascona.

Michlmayr, M. (2005). Software Process Maturity and the Success of Free Software Projects. Software Engineering: Evolution and Emerging Technologies, IOS Press.

Milakis, D., B. van Arem and B. van Wee (2015). "a review of literature and directions for future research." Delft University of Technology.

Miska, M., K. Gajananan, E. Chung and H. Predinger (2011). A traffic simulation standard based on data marts. Proceedings of the Australasian Transport Research Forum 2011, PATREC.

Miska, M., E. Santos, E. Chung and H. Prendinger (2011). OpenTraffic - An Open Source Platform for Traffic Simulation. Australasian Transport Research Forum

Mohammadian, A. and S. T. Doherty (2006). "Modeling activity scheduling time horizon: Duration of time between planning and execution of pre-planned activities." Transportation Research Part A: Policy and Practice **40**(6): 475-490.

Neuhold, R., M. Haberl, M. Fellendorf, G. Pucher, M. Dolancic, M. Rudigier and J. Pfister (2017). Generating a Lane-Specific Transportation Network Based on

Floating-Car Data. Advances in Human Aspects of Transportation, Springer: 1025-1037.

Nuseibeh, B. and S. Easterbrook (2000). Requirements engineering: a roadmap. Proceedings of the Conference on the Future of Software Engineering, ACM.

Olstam, J. J. (2009). Simulation of Surrounding Vehicles in Driving Simulators, Linkoping.

Pau, M. (2002). "Speed bumps may induce improper drivers' behavior: Case study in Italy." Journal of transportation engineering **128**(5): 472-478.

Pingel, T. J. (2010). "Modeling slope as a contributor to route selection in mountainous areas." Cartography and Geographic Information Science **37**(2): 137-148.

Pinjari, A. R. and C. R. Bhat (2011). "Activity-based travel demand analysis." A Handbook of Transport Economics **10**: 213-248.

Planning, C. M. A. f. (1959). Chicago Area Transportation Study: FINAL REPORT. Chicago.

Prlić, A. and J. B. Procter (2012). "Ten Simple Rules for the Open Development of Scientific Software." PLoS Comput Biol **8**(12): e1002802.

Rämä, P. and R. Kulmala (2000). "Effects of variable message signs for slippery road conditions on driving speed and headways." Transportation research part F: traffic psychology and behaviour **3**(2): 85-94.

Raza, A., L. F. Capretz and F. Ahmed (2010). "“Improvement of Open Source Software Usability: An Empirical Evaluation from Developers' Perspective,”." Advances in Software Engineering **vol. 2010**: 12.

Raza, A., Luiz F. Capretz and F. Ahmed (2010). "Improvement of Open Source Software Usability: An Empirical Evaluation from Developers' Perspective." Advances in Software Engineering **2010**: 12.

Reichardt, L., J. Maurer, T. Fugen and T. Zwick (2011). "Virtual Drive: A Complete V2X Communication and Radar System Simulator for Optimization of Multiple Antenna Systems." Proceedings of the IEEE **99**(7): 1295-1310.

Rieser, M. (2010). "Adding transit to an agent-based transportation simulation: concepts and implementation."

Roure, D. and David (2009). Software Design for Empowering Scientists. G. Carole. **26:** 88-95.

Ruas, A., C. Gold, Y. Yue, A. G.-O. Yeh and Q. Li (2008). Road Network Model for Vehicle Navigation using Traffic Direction Approach. Headway in Spatial Data Handling, Springer Berlin Heidelberg: 613-629.

Rüetschi, U.-J. (2007). Wayfinding in Scene Space Modelling Transfers in Public Transport.

Schaap, J., S. Zlatanova and P. van Oosterom (2010). Towards a 3D geo-data model to support pedestrian routing in multimodal public transport travel advices, MSc Thesis.

Schakel, W., B. van Arem, H. van Lint and G. Tamminga (2013). A modular approach for exchangeable driving task models in a microscopic simulation framework. Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on, IEEE.

Schönauer, R., M. Stubenschrott, W. Huang, C. Rudloff and M. Fellendorf (2012). "Modeling Concepts for Mixed Traffic: Steps Toward a Microscopic Simulation Tool for Shared Space Zones." Transportation Research Record: Journal of the Transportation Research Board(2316): 114-121.

Schrock, S., H. Hawkins Jr and S. Chrysler (2005). "Effectiveness of lane direction arrows as pavement markings in reducing wrong-way movements on two-way frontage roads." Transportation Research Record: Journal of the Transportation Research Board(1918): 63-67.

Shah, S. K. (2005). "Motivation, Governance & the Viability of Hybrid Forms in Open Source Software Development." SSRN eLibrary.

Stadler, A. and T. H. Kolbe (2007). Spatio-semantic coherence in the integration of 3D city models. Proceedings of the 5th International Symposium on Spatial Data Quality, Enschede.

Stein, M. and A. Geyer-Schulz (2013). "A Comparison of Five Programming Languages in a Graph Clustering Scenario." J. UCS **19**(3): 428-456.

Tamminga, G., M. Miska, H. Van Lint, E. Santos, A. Nakasone, H. Prendinger and S. Hoogendoorn (2012). Design of an Open Source Traffic and Travel Simulation Framework.

Tamminga, G., L. van den Brink, H. Van Lint, J. Stoter and S. Hoogendoorn (2013). Toward GIS-Compliant Data Structures for Traffic and Transportation Models. Transportation Research Board 92nd Annual Meeting.

Tampère, C. M., R. Corthout, D. Cattrysse and L. H. Immers (2011). "A generic class of first order node models for dynamic macroscopic simulation of traffic flows." Transportation Research Part B: Methodological **45**(1): 289-309.

Tiwana, A., B. Konsynski and A. A. Bush (2010). "Research Commentary--Platform Evolution: Coevolution of Platform Architecture, Governance, and Environmental Dynamics." Information systems research **21**(4): 675-687.

Toledo, T., O. Cats, W. Burghout and H. N. Koutsopoulos (2010). "Mesoscopic simulation for transit operations." Transportation Research Part C: Emerging Technologies **18**(6): 896-908.

Transport, V. D. O. (2009). Implementing Activity-based models in Virginia. VTM Research Paper 09-01.

Transportation, V. D. o. (2009). Implementing Activity-Bases Models in Virginia. VTM Research Paper 09-01, VDOT.

Treiber, M., R. Germ and A. Kesting (2015). From drivers to athletes: Modeling and simulating cross-country skiing marathons. Traffic and Granular Flow'13, Springer: 243-249.

Ulugtekin, N., A. Dogru and R. Thomson (2004). "Modelling urban road networks integrating multiple representations of complex road and junction structures." Geoinformatics: 757-764.

van den Brink, L., J. Stoter and S. Zlatanova (2012). "Establishing a national standard for 3D topographic data compliant to CityGML."

Van den Brink, L., J. Stoter and S. Zlatanova (2012). Modelling an application domain extension of CityGML in UML. ISPRS Conference 7th International Conference on 3D Geoinformation, The International Archivees on the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XXXVIII-4, part C26, 16–17 May 2012, Québec, Canada, ISPRS.

van den Brink, L., J. Stoter and S. Zlatanova (2013). "Establishing a national standard for 3D topographic data compliant to CityGML." International Journal of Geographical Information Science **27**(1): 92-113.

van Essen, R. and V. Hiestermann (2005). “X-GDF”—THE ISO MODEL OF GEOGRAPHIC INFORMATION FOR ITS. ISPRS Workshop on Service and Application of Spatial Data Infrastructure, XXXVI (4/W6), Citeseer.

van Lint, H., W. Schakel, G. Tamminga, P. Knoppers and A. Verbraeck (2016). "Getting the Human Factor into Traffic Flow Models: New Open-Source Design to Simulate Next Generation of Traffic Operations." Transportation Research Record: Journal of the Transportation Research Board(2561): 25-33.

van Noort, M., B. van Arem and P. Byungkyu (2010). MOBYSIM: An integrated traffic simulation platform. Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on.

Varhelyi, A. (1998). "Drivers' speed behaviour at a zebra crossing: a case study." Accident Analysis & Prevention **30**(6): 731-743.

Verbraeck, A. (2017). Exploring the Effects of Perception Errors and Anticipation Strategies on Traffic Accidents-A Simulation Study. Advances in Human Factors in Simulation and Modeling: Proceedings of the AHFE 2017 International Conference on Human Factors in Simulation and Modeling, July 17–21, 2017, The Westin Bonaventure Hotel, Los Angeles, California, USA, Springer.

Vreeswijk, J. and R. Blokpoel (2012). Uniform control targets and balanced strategies for traffic network management. ITS World Congress.

Vreeswijk, J., B. Claassens, M. Snoek, R. v. d. Dries and M. Harmenzon (2016). Intersection Topology Format, Ministry of Infrastructure and the Environment.

Weibel, R. and G. Dutton (1999). "Generalising spatial data and dealing with multiple representations." Geographical information systems **1**: 125-155.

Weiner, E. (1986). Urban transportation planning in the United States: an historical overview (revised edition)**:** Medium: X; Size: Pages: 159.

Wigan, M. and P. Drain (2002). "Open transport modeling networking for transportation applications."

wiki.openstreetmap.org. (2017). "About OpenStreetMap." from https://wiki.openstreetmap.org/wiki/About_OpenStreetMap.

Wilkie, D., J. Sewall, W. Li and M. C. Lin (2015). "Virtualized Traffic at Metropolitan Scales." Frontiers in Robotics and AI **2**: 11.

Willemsen, P., J. K. Kearney and H. Wang (2006). "Ribbon networks for modeling navigable paths of autonomous agents in virtual environments." IEEE Transactions on Visualization and Computer Graphics **12**(3): 331-342.

Ye, X., R. M. Pendyala and G. Gottardi (2007). "An exploration of the relationship between mode choice and complexity of trip chaining patterns." Transportation Research Part B: Methodological **41**(1): 96-113.

Zheng, Z. (2014). "Recent developments and research needs in modeling lane changing." Transportation research part B: methodological **60**: 16-32.

Zhu, Q. and Y. Li (2008). "Hierarchical lane-oriented 3D road-network model." Int. J. Geogr. Inf. Sci. **22**(5): 479-505.

Summary

Over the past decades, transport and traffic models have become powerful tools for transport and traffic practitioners and academics all over the world. Most of these commercial model packages provide proprietary and closed source software, which generally implies that users are not enabled to freely edit, modify and extend the code. Specifically for some of the new fields of applications such as the modelling of in-car systems and other ITS systems, and more generally for scientific research, access to the algorithms is often essential. The de facto option then is to code traffic models (or parts thereof) from scratch. To prevent this loss of knowledge and experience, an approach is required where code reuse and proper documentation is stimulated.

The open source software project approach is one of the means to achieve these objectives, as it guarantees that the generated knowledge, in the form of source code and documentation, stands open to anyone. Open source projects require sufficient preparations with respect to project governance, the transport modelling architecture and its implementation in software (design and quality). Once these requirements are fulfilled, the availability of reusable and mature software allows for continuous development of new and improved models (activity based, car following, merging, gap acceptance, response to ITS, etc.). This thesis determines the requirements of such an open source software project for transport and traffic modelling and evaluates *OpenTrafficSim*, an open source traffic simulation suite. The main motivation for setting up *OpenTrafficSim* is the need for a platform that provides free to use code for traffic simulation. While there have been numerous publications that are focused on the development of dedicated modelling methods and algorithms, far less attention has been paid to the architecture, design and practical implementation of a comprehensive transport and traffic modelling system. This thesis aims to derive the key requirements for the development of a successful open source software project for transport and traffic. After this analysis, we evaluate the OpenTrafficSim project on the basis of these requirements. As is shown, the

In addition we regard the way of modelling the transportation infrastructure in an open source transport and traffic model. This requires a design that firstly meets the functional requirements for transport and traffic models and secondly enables an efficient data-exchange between models and external data-sources.

There is a wide range of transport and traffic model approaches that varies from global and strategic towards very detailed and operational. These models contain a broad collection of various interacting methods and components. In almost all cases, the definition of the model objects is not standardized, but depends on the choice of the model package provider. Moreover, we observe that even within a package the definition of the model objects is not aligned between different levels of detail. One of the objectives of this thesis is to create a model design that allows an efficient exchange between various levels of detail and modes of traffic.

To represent the real world, such as the transport infrastructure and the activities and their locations, the models require input data. More and more, these input data are collected and stored in a uniform way, based on open geospatial data standards. The basic philosophy is to collect these data only once, and support multiple use of these data. To support such a sustainable (re)use of these external data within the models, a match between the model objects

with objects from external data sources is required, to allow an efficient and stable data-exchange at various levels of detail.

As this thesis shows, the current open data standards do not yet provide all required data sufficiently for the purpose of transport and traffic. Specifically, the modelling of lane details and markings, as well as a description of the junction area with the connectivity between incoming and egress lanes, is lacking. The open data format CityGML is proposed as a feasible standard to align data-input objects with their counterpart in the model. The main arguments are (1) the coverage of both the transport infrastructure and the surrounding environment that represent the activity locations, and (2) the recognition of multiple levels of detail. A further extension of CityGML for transportation is proposed, in order to fully support the model's data requirements.

A first requirement for the design of model objects that enables multiple levels of detail, is a proper data model that represents the most detailed level. From this base level, all other levels of detail are realized by aggregation. The identification of this base level starts with an analysis of transport and traffic modelling methods at the highest level of detail, regarding their requirements in terms of objects, relations and input-data. Specific attention is paid to exchange locations where modal infrastructures meet and connect.

The switch to lower levels of detail impacts both the representation of transport and traffic demand and the assignment of traffic. At the highest level of detail, demand is represented by a trip-pattern object that contains the sequence of trips between activities performed by a specific person. By means of aggregation procedures, the trip-pattern can be modified into mode specific origin-destination matrices that are generally used to represent demand at lower levels of detail. Likewise, the representation of activity locations at various spatial resolutions can be enabled by a hierarchical zoning system, for instance combining individual addresses and postal area codes at various levels of detail. The modelling and representation of traffic flows at various levels of detail is often classified by microscopic, mesoscopic and macroscopic. At the highest level of detail, the traffic flow phenomena are largely determined by the driver/vehicle behaviour. The meso- and macroscopic approaches either average the individual behaviours and/or aggregate traffic units. The switch from microscopic to macroscopic approaches entails the replacement of behaviour as a result of discrete events, by methods that estimate the aggregate characteristics of these events. The main differences between micro and macroscopic approaches regards the substitution of describing the behaviour of vehicles and their interactions by aggregate and continuous functions that require flow, parameters and attributes of the infrastructure that mimic the impact of vehicle interactions. As these less detailed models do not model the movements, new attributes (for instance estimated average speed, jam density) and functions (such as the fundamental diagram) are required to estimate the impact of all these individual movement choices on an aggregated level. The description of the infrastructure for the meso- and macroscopic networks therefor requires both a generalization of the links and junctions, and a change of their attributes. The transfer of attributes that contain behavioural aspects, is not straightforward and is explained by example.

Finally, this thesis proposes a novel design for the transportation infrastructure at the, for modelling required, highest level of detail. The network is essentially based on a graph of links and nodes, and includes the details of the road in the cross section object of a link. It includes both the area where traffic moves or parks, and auxiliary objects that describe the further characteristics of the road such as a sidewalk, barrier and berm. A junction is composed by internal (virtual) links that connect the incoming and egress lanes, and explicitly define the allowed turning movements. In addition, the functional elements such as traffic light controllers and car parks are connected with the road network. These functional units use devices, such as sensors or traffic lights, to inform, measure and control traffic. In order to connect trips with the network, the traffic zone is defined as an area that encompasses a collection of activity locations. At the highest level of detail, every activity location can be connected with the 'boarding' location for the first mode (car park, bus-stop, etc.) by means of a walk link. The combination of a graph together with the details of the road topology, results in a dual system that resembles both the requirements for microsimulation (detailed lane description) and the meso- and macroscopic (link based graph) network representations.

Samenvatting

De afgelopen decennia hebben de verkeers- en vervoermodellen zich meer en meer ontwikkeld tot nuttige hulpmiddelen voor verkeersmodelleurs en academici. Het grootste deel van deze commerciële model pakketten betreft 'closed source software'. Dat impliceert dat gebruikers in de meeste gevallen geen mogelijkheid hebben om de achterliggende code te wijzigen dan wel aan te vullen. Vooral bij nieuwe toepassingen, zoals het modelleren van 'in-car' systemen en andere intelligente voertuigsystemen, en in meer algemene zin voor wetenschappelijk onderzoek is toegang tot de algoritmes vaak essentieel. Als dat niet mogelijk is, resteert het zelf programmeren van verkeersmodellen als laatste optie. Omdat de kwaliteit van deze code vaak van onvoldoende kwaliteit is, bestaat er behoefte aan een aanpak waarbij het hergebruik van code wordt gestimuleerd.

De open source software benadering is een van de mogelijkheden om deze doelstelling te bereiken. Het biedt namelijk de garantie dat de ingebrachte kennis, in de vorm van code en documentatie, voor iedereen beschikbaar is. Om tot een succesvol open source softwareproject te komen, is een grondige voorbereiding nodig. Dit betreft enerzijds de organisatie van het project waarbij in de initiële fase een goede sturing noodzakelijk is om het project in de steigers te zetten. Een tweede aspect betreft de kwaliteit van de software. Naast een goed functioneel ontwerp, is ook de kwaliteit van de software van belang. Tenslotte moet er ook voldoende functionaliteit geboden worden om de software voor de gebruikers toegankelijk te maken. Het aan de TU Delft ontwikkelde open source softwareproject *OpenTrafficSim* biedt een volwaardige simulatie van het verkeer, waarin het voertuigvolg- en strookwisselgedrag volgens de meest recente inzichten is gemodelleerd. Daarbij is het mogelijk om zowel autosnelwegen en delen van het onderliggend wegennet te simuleren. Uit de evaluatie van OpenTrafficSim blijkt dat het ontwerp van de software aan de meeste eisen voldoet. Het project is onderverdeeld in meerdere pakketten en modules, waarbij de functies en methodes op een overzichtelijke manier zijn gegroepeerd en gecodeerd. De verschillende methoden zijn daardoor relatief eenvoudig aan te passen, te vervangen door alternatieven, en ook uit te breiden met nieuwe functies. Aandachtspunten uit de evaluatie betreffen met name de gebruiksvriendelijkheid voor nieuwe gebruikers: er is nog geen grafische gebruikersinterface en onvoldoende documentatie. De methoden en attributen worden weliswaar van commentaar voorzien (JavaDoc), maar een handleiding ontbreekt.

Het tweede onderwerp van deze studie betreft het ontwerp van de weginfrastructuur in verkeers- en vervoersmodellen. Het modellenspectrum kent een breed scala aan methoden, om het verkeer te beschrijven. Afhankelijk van het doel van het model kan de aanpak daarbij variëren van globaal tot gedetailleerd. Voor modeltoepassingen wordt voor het merendeel gebruikt gemaakt van commerciële pakketten. Deze modellen hebben allen een eigen datastructuur. Enerzijds is dat verklaarbaar vanwege de verschillen in schaalniveau. Een microsimulatie model beschrijft de infrastructuur in veel meer detail dan een macroscopisch model. Toch blijkt dat er ook bij modellen van eenzelfde detailniveau geen uniforme definitie te zijn van de infrastructuurbeschrijving. Dit bemoeilijkt de uitwisseling van data tussen de modellen. Het is niet mogelijk om de netwerken tussen de pakketten direct uit te wisselen.

Om de modelinfrastructuur te bouwen is invoerdata nodig. Meer en meer komen er topografische bestanden beschikbaar om dergelijke netwerken te bouwen. Een belangrijk onderdeel daarbij is het standaardiseren van de beschrijving van de data-objecten. Door hier duidelijke en eenduidige afspraken over te maken, kunnen de data veel gemakkelijker worden uitgewisseld. Deze afspraken worden vastgelegd in geografische datastandaarden. Om de uitwisseling met verkeersmodellen mogelijk te maken is het van belang dat ook de verkeersmodellen hun infrastructuur zodanig modelleren dat deze aansluit op de geografische datastandaard. Daarbij moet bovendien ook rekening gehouden worden met de verschillen in detailniveau. Een wegennetwerk in macroscopische modellen is veel minder gedetailleerd dan een simulatiemodel.

Van de beschikbare data-standaarden selecteren we CityGML onder meer omdat deze meerdere 'levels of detail' onderscheidt en daarnaast ook de omgeving van de weg (gebouwen, terrein) in beeld brengt. Analyse van de CityGML objectencatalogus laat zien dat niet alle onderdelen van de infrastructuur voldoende zijn beschreven. Een uitbreiding van CityGML is dan ook noodzakelijk. Enerzijds is het nodig om ontbrekende objecten op te nemen. Een duidelijk voorbeeld is het ontbreken van het object kruispunt, met informatie over opstelstroken en toegestane afslagbewegingen. Daarnaast ontbreekt een beschrijving van de infrastructuur in de vorm van links en knopen, waarmee een routeerbaar netwerk is te genereren.

Een tweede aspect is de mogelijkheid om de infrastructuur op meerdere detailniveaus te beschrijven. Een eerste voorwaarde om dit mogelijk te maken, is de beschrijving van de infra op een hoog detailniveau. Vanuit dit niveau kunnen meer globale beschrijvingen van deze netwerken worden gegenereerd. Het grootste verschil tussen de netwerken van de simulatie- en de macroscopische modellen betreft de beschrijving van de rijstroken. In de macroscopische modellen is het niet nodig om de geometrie van de rijstroken te beschrijven, maar volstaat het vastleggen van het aantal stroken. Het aantal rijstroken is eenvoudig af te leiden vanuit het simulatienetwerk, net als andere attributen zoals de maximumsnelheid en de voorrangssituatie. Dit is echter niet mogelijk voor alle objecten. Een voorbeeld hiervan is de capaciteit van een weg. In microsimulatiemodellen is de capaciteit van een weg het resultaat van het voertuig volg- en strookwisselgedrag. In de mesoscopische en macroscopische modellen wordt dit gedrag niet of veel beperkter gemodelleerd. Daarmee is de capaciteit geen modeluitkomst. In plaats daarvan wordt de capaciteit dus een invoervariabele, en zijn er andere methodes nodig om de rijtijden te bepalen.

Uiteindelijk zijn er vanuit het perspectief van de verkeersmodellen twee detailniveaus relevant. De meest gedetailleerde beschrijft de netwerken op het niveau van rijstroken (simulatie), terwijl de andere uitgaat van rijbanen (macroscopisch). In het ontwerp van de weginfrastructuur dat we in deze studie presenteren, komt dit onderscheid duidelijk naar voren. We onderscheiden het linkniveau (macro) en koppelen daar een beschrijving van het wegprofiel (micro) aan. Naast het definiëren van de objecten die nodig zijn om een volwaardig netwerk te creëren. Daarbij gaat het zowel over de delen van de waar het verkeer rijdt en parkeert, als ook de directe omgeving zoals de berm, vangrail en voetpad. Ook het kruispunt modelleren we expliciet object, met alle toegestane bewegingen tussen de ingaande en uitgaande takken. Ook functionele objecten zoals een verkeerslichtenregeling worden expliciet vastgelegd. Deze functionele eenheden gebruiken andere objecten zoals een detector en verkeerslicht om het verkeer te reguleren. Dit ontwerp is voor een deel ingebouwd in *OpenTrafficSim*, en blijkt daarbij veel wegconfiguraties (aansluitingen en kruispunten) te kunnen simuleren. Tenslotte blijkt het ontwerp van de weginfrastructuur voor verkeersmodellen goed aan te sluiten op de beschrijving van de infra volgens de CityGML datastandaard (inclusief de voorgestelde uitbreidingen).

About the author

Curriculum Vitae

Guus Tamminga was born in Assen on the 24th of February 1960.

After finishing high-school, he completed his master´s degree in Environmental Economics at the University of Groningen in 1985.

His professional career started at the Dutch Agricultural Economics Institute at The Hague. Here, he worked on the development and application of the Dutch Regionalized Agricultural Model (1986-1992).

From 1992 onwards Guus works at Sweco (previously Grontmij) at the mobility department.

In 2009 he combined this job with a PhD study at the Transport & Planning department of the faculty of Civil Engineering & Geosciences of Delft University of Technology.

Currently, he fully works at Sweco again, and enjoys collaborating on a great variety of projects, varying from the development and application of transport and traffic models to data analysis and GIS.

Author’s publications

van Lint, H., W. Schakel, G. Tamminga, P. Knoppers and A. Verbraeck (2016). "Getting the Human Factor into Traffic Flow Models: New Open-Source Design to Simulate Next Generation of Traffic Operations." Transportation Research Record: Journal of the Transportation Research Board(2561): 25-33

Knoop, V., G. Tamminga and L. Leclercq (2016). Network transmission model: Application to a real world city, No. 16-1526.

Tamminga, G., P. Knoppers and J.W.C. van Lint; “Open Traffic: a toolbox for research”, 3rd workshop ABMTRANS, Procedia Computer Science, 2014

Tamminga, Guus, Linda van den Brink, Jantien Stoter, Hans van Lint; “Naar een loepzuivere kaart”, Verkeerskunde 2/2013

Tamminga, G., L. van den Brink, H. Van Lint, J. Stooter, S.P. Hoogendoorn;
‘Towards GIS compliant data structures for traffic and transportation models’, congress proceedings, TRB 2012

Schakel, W., B. van Arem, H. van Lint and G. Tamminga (2013). A modular approach for exchangeable driving task models in a microscopic simulation framework. Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on, IEEE.

Tamminga, G., M. Miska, E. Santos, H. Van Lint, A. Nakasone, H. Prendinger, S. Hoogendoorn; ‘Design of an Open Source Framework for traffic and travel simulation’, Transport Research Record, Journal of the Transportation Reserach Board, No. 2291

Tamminga, Guus, Tu, Daamen, Hoogendoorn; ‘Influence of departure time spans and corresponding network performance on evacuation time’, Transport Research Record, Journal of the Transportation Reserach Board, No. 2234

Hoogendoorn, R., G. Tamminga, S. Hoogendoorn and W. Daamen (2010). Longitudinal driving behavior under adverse weather conditions: Adaptation effects, model performance and freeway capacity in case of fog. Intelligent transportation systems (itsc), 2010 13th international ieee conference on, IEEE.

Tu, H., Guus Tamminga, H. Drolenga, J. de Wit, W. van der Berg; ‘Evacuation Plan of the City of Almere: Simulating the Impact of Driving Behavior on Evacuation Clearance Time’, ICEM 2009

Tamminga, G.F, A. Rijkee, F.J. de Jong; ‘Kunnen we het toekomstig verkeersbeeld simuleren met de modellen van vroeger’, Colloquium Verkeerplanologisch Speurwerk 2008

Schreuder, M.A., L. Molenkamp, G.F. Tamminga, M.E. Kraan; ‘Vulnerability of a national roadwork’, INSTR 2007

Tamminga, G.F, J. Zee en F. de Jong; ‘Parkeerzoekverkeer gesimuleerd: helpt een parkeerverwijssysteem?’, Colloquium Verkeerplanologisch Speurwerk 2005

N. Cohn, G.F. Tamminga, J. Zee; ‘Naar een modelstelsel; ervaringen uit Limburg en Antwerpen’, Colloquium Vervoersplanologisch Speurwerk 2003

Dick Ettema, Guus Tamminga, Harry Timmermans, Theo Arentze; ‘A multi-agent model system of departure time and route choice under travel time uncertainty’,

Tamminga, G.F., D. Ettema, P. Veeke, L. Smit; ‘Dynamiek in het NRM Noord-Brabant’; in Verkeerskunde oktober 1999

Tamminga, G.F., P.H.G. van Bekkum en G.A. van Velzen; ‘Dynamische modelleren met TRIPS’; in: Verkeerskunde januari 1997

Wossink, G. and G. Tamminga (1991). Scenario assessment integrating environmental and agricultural developments for research at farm and sector level. Proc. 25th EAAE seminar The environment and agricultural resources management. Viterbo, Italy.

G.F. *Tamminga*. H. *von Meyer*. D. *Strijker*. F.E. *Godeschalk*. Agriculture in the less favoured. areas of the ec-10, The Hague, Agricultural Economics Research institute LEI-DLO, The Hague, 1991.

TRAIL Thesis Series

The following list contains the most recent dissertations in the TRAIL Thesis Series. For a complete overview of more than 200 titles see the TRAIL website: www.rsTRAIL.nl.

The TRAIL Thesis Series is a series of the Netherlands TRAIL Research School on transport, infrastructure and logistics.

Tamminga, G., *A Novel Design of the Transportation Context of OpenTrafficSim,* T2019/4, March 2019, TRAIL Thesis Series, the Netherlands

Lin, X., *Controlled Perishable Goods Logistics: Real-time coordination for fresher products*, T2019/3, January 2019, TRAIL Thesis Series, the Netherlands

Dafnomilis, I., *Green Bulk Terminals: A strategic level approach to solid biomass terminal design*, T2019/2, January 2019, TRAIL Thesis Series, the Netherlands

Feng, Fan, *Information Integration and Intelligent Control of Port Logistics System,* T2019/1, January 2019, TRAIL Thesis Series, the Netherlands

Beinum, A.S. van, *Turbulence in Traffic at Motorway Ramps and its Impact on Traffic Operations and Safety,* T2018/12, December 2018, TRAIL Thesis Series, the Netherlands

Bellsolà Olba, X., *Assessment of Capacity and Risk: A Framework for Vessel Traffic in Ports,* T2018/11, December 2018, TRAIL Thesis Series, the Netherlands

Knapper, A.S., *The Effects of using Mobile Phones and Navigation Systems during Driving,* T2018/10, December 2018, TRAIL Thesis Series, the Netherlands

Varotto, S.F., *Driver Behaviour during Control Transitions between Adaptive Cruise Control and Manual Driving: empirics and models,* T2018/9, December 2018, TRAIL Thesis Series, the Netherlands

Stelling-Kończak, A., *Cycling Safe and Sound,* T2018/8, November 2018, TRAIL Thesis Series, the Netherlands

Essen, van M.A., *The Potential of Social Routing Advice,* T2018/7, October 2018, TRAIL Thesis Series, the Netherlands

Su, Zhou, *Maintenance Optimization for Railway Infrastructure Networks*, T2018/6, September 2018, TRAIL Thesis Series, the Netherlands

Cai, J., *Residual Ultimate Strength of Seamless Metallic Pipelines with Structural Damage*, T2018/5, September 2018, TRAIL Thesis Series, the Netherlands

Ghaemi, N., *Short-turning Trains during Full Blockages in Railway Disruption Management,* T2018/4, July 2018, TRAIL Thesis Series, the Netherlands

Gun, van der J.P.T., *Multimodal Transportation Simulation for Emergencies using the Link Transmission Model,* T2018/3, May 2018, TRAIL Thesis Series, the Netherlands

Van Riessen, B., *Optimal Transportation Plans and Portfolios for Synchromodal Container Networks,* T2018/2, March 2018, TRAIL Thesis Series, the Netherlands

Saeedi, H., *Network-Level Analysis of the Market and Performance of Intermodal Freight Transport,* T2018/1, March 2018, TRAIL Thesis Series, the Netherlands

Ypsilantis, P., *The Design, Planning and Execution of Sustainable Intermodal Port-hinterland Transport Networks,* T2017/14, December 2017, TRAIL Thesis Series, the Netherlands

Han, Y, *Fast Model Predictive Control Approaches for Road Traffic Control,* T2017/13, December 2017, TRAIL Thesis Series, the Netherlands

Wang, P., *Train Trajectory Optimization Methods for Energy-Efficient Railway Operations,* T2017/12, December 2017, TRAIL Thesis Series, the Netherlands

Weg, G.S. van de, *Efficient Algorithms for Network-wide Road Traffic Control*, T2017/11, October 2017, TRAIL Thesis Series, the Netherlands

He, D., *Energy Saving for Belt Conveyors by Speed Control,* T2017/10, July 2017, TRAIL Thesis Series, the Netherlands

Bešinović, N., *Integrated Capacity Assessment and Timetabling Models for Dense Railway Networks,* T2017/9, July 2017, TRAIL Thesis Series, the Netherlands

Chen, G., *Surface Wear Reduction of Bulk Solids Handling Equipment Using Bionic Design,* T2017/8, June 2017, TRAIL Thesis Series, the Netherlands

Kurapati, S., *Situation Awareness for Socio Technical Systems: A simulation gaming study in intermodal transport operations,* T2017/7, June 2017, TRAIL Thesis Series, the Netherlands

Jamshidnejad, A., *Efficient Predictive Model-Based and Fuzzy Control for Green Urban Mobility,* T2017/6, June 2017, TRAIL Thesis Series, the Netherlands

Araghi, Y., *Consumer Heterogeneity, Transport and the Environment*, T2017/5, May 2017, TRAIL Thesis Series, the Netherlands

Kasraian Moghaddam, D., *Transport Networks, Land Use and Travel Behaviour: A long term investigation,* T2017/4, May 2017, TRAIL Thesis Series, the Netherlands

Smits, E.-S., *Strategic Network Modelling for Passenger Transport Pricing,* T2017/3, May 2017, TRAIL Thesis Series, the Netherlands

Tasseron, G., *Bottom-Up Information Provision in Urban Parking: An in-depth analysis of impacts on parking dynamics,* T2017/2, March 2017, TRAIL Thesis Series, the Netherlands

TRAIL

Summary

Guus Tamminga works as a transport and traffic modelling advisor at Sweco. He performed his PhD research (part-time) at Delft University of Technology. His aim is to intensify the dissemination of knowledge from academic research to practitioners in the field of transport and traffic.

About the Author

This dissertation covers two questions. How to make a transport and traffic model that enables model users and developers to reuse the existing code, learn from it, improve methods, and/or extend it with innovative functionality? And what is a good design of the modelled transportation infrastructure for road based traffic, in order to fulfil the functional requirements and enable a proper data-exchange with other models and external data sources?

TRAIL Research School ISBN 978-90-5584-247-6

Erasmus University Rotterdam
Radboud University
university of groningen faculty of economics and business
TU Delft Delft University of Technology
UNIVERSITY OF TWENTE.
TU/e Technische Universiteit Eindhoven University of Technology