KTH Infrastructure

Hybrid microscopic-mesoscopic traffic simulation

Wilco Burghout

Doctoral Dissertation
Royal Institute of Technology
Stockholm, Sweden 2004

Royal Institute of Technology
Department of Infrastructure
Division of Transportation & Logistics
Centre for Traffic Simulation
Teknikringen 72
SE-100 44 Stockholm
Sweden

TRITA-INFRA 04-035
ISSN 1651 - 0216
ISRN KTH/INFRA/--04/035--SE
ISBN 91-7323-099-5

Abstract

Traffic simulation is an important tool for modelling the operations of dynamic traffic systems and helps analyse the causes and potential solutions of traffic problems such as congestion and traffic safety. Microscopic simulation models provide a detailed representation of the traffic process, which makes them most suitable for evaluation of complicated traffic facilities and Intelligent Transportation Systems that often consist of complex traffic management, safety and information systems. Macroscopic and mesoscopic models on the other hand, capture traffic dynamics in lesser detail, but are faster and easier to apply and calibrate than microscopic models. Therefore they are most suitable for modelling large networks, while microscopic models are usually applied to smaller areas.

The objective of this thesis is to combine the strengths of both modelling approaches and diminish their individual weaknesses by constructing a hybrid mesoscopic-microscopic model that applies microscopic simulation to areas of specific interest, while simulating a surrounding network in lesser detail with a mesoscopic model.

Earlier attempts at hybrid modelling have concentrated on integrating macroscopic and microscopic models and have proved difficult due to the large difference between the continuous-flow representation of traffic in macroscopic models and the detailed vehicle-and driver-behaviour represented in microscopic models. These problems are solved in this thesis by developing a mesoscopic vehicle-based and event-based model that avoids the (dis)aggregation problems of traffic flows at the inter-model boundaries. In addition, this thesis focuses on the general problems of consistency across the entire hybrid model.

The requirements are identified that are important for a hybrid model to be consistent across the models at different levels of detail. These requirements vary from network and route-choice consistency to consistency of traffic dynamics across the boundaries of the micro- and mesoscopic submodels. An integration framework is proposed that satisfies these requirements. This integration framework has been implemented in a prototype hybrid model, MiMe, which is used to demonstrate the correctness of the solutions to the various integration issues. The hybrid model integrates MITSIMLab, a microscopic traffic simulation model, and Mezzo, the newly developed mesoscopic model. Both the hybrid model and the new Mezzo model are applied in a number of case studies, including a network in the North of Stockholm, which show their validity and applicability. The results are promising and support both the proposed integration architecture and the importance of integrating microscopic and mesoscopic models.

Keywords: Traffic simulation, Traffic models, Mesoscopic, Microscopic, Hybrid

Acknowledgements

This is the place to admit that while there appears only one author on the cover, this work, just as any other, is a product of the interaction with and support from many people. Among the people I would like to thank for their support and inspiration are:

My main supervisor Professor Karl-Lennart Bång for his encouragement, advice and guidance throughout the process of developing this PhD thesis. His solid knowledge of traffic engineering helped me to broaden my view and approach the problems from both computer science and traffic engineering perspectives.

My supervisor Professor Ingmar Andréasson for his enthusiasm, guidance and continuous interest in my work. Ingmar was ready to discuss any subject, no matter how detailed and provided insight on subjects varying from model design to programming issues. It was his idea to start a project to integrate microscopic and mesoscopic simulation models, and to make the Mezzo mesoscopic simulator event-based.

Professor Haris N. Koutsopoulos provided guidance and insight throughout the project. Our weekly telephone conversations helped me in many ways. I am much indebted to him for his unselfish support and how to do research is only one of the things I learned from him.

Thanks to Moshe Ben-Akiva, director of the MIT Intelligent Transportation Program, as well as Tomer Toledo and Constantinos Antoniou at the MIT ITS lab, for providing the MITSIMLab source code and assisting me with MITSIMLab - related issues of the implementation of the MiMe hybrid model.

Thank you to my 'office-mates' Xiaoliang Ma and Masatu Chiguma and all other colleges at the Centre for Traffic Simulation and the Trafik & Logistik unit of the Infrastructure department at KTH.

I would like to thank Vinnova and Stockholms Stads Gatu- och Fastighetskontor who financed this PhD project.

Furthermore I thank Allogg AB and the Swedish National Road Administration for providing high-quality measurement data of speeds and time-headways from the E4 Essingeleden.

I am very grateful for the support and encouragement I received from my family, and especially my parents for bringing me up as someone who is free to think for himself.

And finally, to you Raluca my deepest thanks for always being by my side and looking at what I'm doing with the right mix of interest, encouragement and humorous scepticism. Your question "so what is it all good for, then?" never failed to make me focus on what it was I was supposed to be doing and why. It also made me smile.

Stockholm, November 2004,

Wilco Burghout

Contents

Chapter 1. Introduction

1.1 Background

Traffic and transportation are essential to all economies around the world, getting people and goods to places where they are wanted, needed, and useful. As a consequence, problems in the transportation system have a large impact on almost all areas of economic activity, and are therefore given more and more attention by planners and policy makers. In the last decades we have seen a large increase in traffic and transport demand, which has created and worsened capacity problems in the infrastructure, resulting in traffic jams and delays. This happened despite a continuous effort by authorities to extend and improve the traffic infrastructure to meet the increased demand. As an added complication, the space for extending the infrastructure has become limited, especially in metropolitan areas where the described capacity problems occur.

As the demand may be expected to continue to grow for some time to come, it is not difficult to see the need to invest not only in improved infrastructure, but in the planning and management of this infrastructure as well.

Although there exist many modes of transport, including train, flight, waterways, cycling and walking, the bulk of traffic consists of vehicles on the road network. Logically, most planning and research efforts have focused on this mode of transport. Planning and managing congested road traffic networks requires insight into the aspects of traffic flow operations, such as what causes congestion, what determines the time and place of traffic breakdown, how does congestion propagate through the network, etc.

In the last decade the development of so-called Intelligent Transportation Systems (ITS) has resulted in an increased effort in developing traffic planning and management tools. ITS is a term that is used for a large range of traffic information and control technologies. When integrated into the transport infrastructure system and into the vehicles themselves, such technologies help monitor and manage traffic flow, reduce congestion, provide alternate routes to travellers and increase safety. An example of such ITS systems are Advanced Traveller Information Systems (ATIS) that provide travellers with real-time information along their routes. Other ITS systems are Advanced Traffic Management Systems (ATMS) that collect real-time traffic information, used to optimise real-time traffic control systems and Incident Management Systems. ITS systems provide traffic operators with the information and control tools to respond swiftly to incidents on the traffic network (such as accidents). These systems are called 'Intelligent' because of their use of advanced communication technologies, real-time information and real-time control. The detailed and disaggregated nature of the information that these models use, as well as their direct way of influencing the traffic behaviour, has called for new methods to study the traffic system.

After some seventy years of research in traffic flow and the application of its findings to the planning and management of traffic, the discipline has developed a wide variety of methods and tools it can use. For an overview of the state of the art in traffic flow research, see (May, A.D. 1990), (Daganzo, C. 1997) or (Gartner, N.H. *et al.* 1997).

Besides the familiar tools such as handbooks and manuals (e.g. the Highway Capacity Manual (HCM 2000)) augmented with calculation utilities, the use of traffic flow models has become common. There exist a large number of such models, and they are usually characterised along two dimensions: the level of detail in which they describe the traffic processes, and the way they operate in producing the answers (operationalisation).

The operationalisation of models is usually classified as *analytical*, where the solution to a set of differential equations describing the traffic system is obtained analytically (using calculus), or *simulation*, where the successive changes of the traffic system over time (space-time dynamics) are reproduced (approximated) in the model.

Analytical models can be both static and dynamic, but will usually compute the *result* of a given traffic problem, using numerical methods to produce the solutions. Simulation models on the other hand *follow* the dynamics of the traffic system, and give in a sense a continuous view of the state of the traffic system over time. This characteristic of simulation models is an advantage over analytical models, since it gives more information and insight into what is happening to the traffic system under study. However, until recently the computational cost at which this advantage came was simply too high for simulation to be used on any traffic system that consisted of more than a couple of intersections. But with the steady advance of computation power, this issue now is of lesser importance, and recent years have seen increased use of traffic simulation models.

The levels of detail in simulation models range from macroscopic via mesoscopic to microscopic. Macroscopic models describe the traffic at a high level of aggregation as flow (the number of vehicles per hour that pass a certain point), without considering its constituent parts (the vehicles), whereas microscopic models describe the behaviour of the entities making up the traffic stream (the vehicles) as well as their interactions in detail. Mesoscopic models are at an intermediate level of detail, for instance describing the individual vehicles, but not their interactions.

In order to represent the traffic system that is studied in a suitable way, two aspects and their interaction need to be modelled. On the one hand there is the *Supply side* that consists of the traffic network (the roads and intersections) and its performance together with all the control and information systems (traffic lights, Variable Message Signs (VMS), speed limits, etc.). On the other hand there is the *Demand side* that consists of the travellers and their behaviour. In other words, the drivers want to go from some place to another (Demand) and the traffic infrastructure provides the means to do that (Supply). These sides interact in the way that travellers react to the speed limit signs, the conditions on the roads and so forth, by making different choices (route choices, speed choices etc.), and control and information systems adapt to the choices of drivers.

Traditionally a lot of effort has gone into modelling equilibrium conditions in the interactions between the demand and supply side. In these models the route-choice decisions of the travellers (demand) are modelled given a network with certain characteristics (supply). Given the choices of the travellers, the roads in the network will have certain performance (travel time, flow). If a lot of travellers use the same road, it will get congested and the travel time (cost) will increase. Just as in regular markets, any alternative with a shorter travel time (lower cost) will therefore become more attractive. Using some algorithm the route choices and resulting network performance are iterated until equilibrium arises in the route choice and network performance. (Wardrop, J.G. 1952) defined the condition for equilibrium in route choices and network performance as follows:

· *All routes actually used between an origin and a destination have the same travel time (cost) and*
· *this travel time (cost) is not larger than the travel time along any other route between that origin-destination pair.*

In other words, for each origin-destination pair, all routes that are used by the drivers will have equal cost (travel time), and there is no (unused) route with a lower travel time. An example of such a model is EMME/2 (INRO-Consultants 1996).
The modelling of the demand (travellers wanting to use the network) and supply side (the traffic network and its performance) by this type of models is usually done in a *static* way. This means that the traffic flows on the network are assumed to stay the same during the study time period. This assumption allows for fast and guaranteed convergence to the equilibrium, but ignores the changes of the traffic situation over time (dynamics). The problem with this approach is that not only the demand varies greatly over time, but the performance of the network (and thus travel times on routes) shows great fluctuations as a result of congestion on certain roads, queues that build up and dissipate, and so on. In addition, while the static approach assumes that a certain demand results *instantaneously* in certain traffic flows on the network, in reality the effect of demand on the performance on different roads in the network occurs after a certain period of time, simply because it takes a certain amount of time to get from the origin to that road. And that amount of time is again dependent on the time-varying traffic flows in the network (how many vehicles are on the roads).
While these facts are known, these models continue to be used for predicting traffic network performance on larger networks, and may produce results that are very far from reality (especially in congested networks), as reported in (Merritt, E. 2003) and (Bång, K.-L.E. 2000).

The inherent *dynamic* nature of traffic is not represented by the *static* models, which leads to poor predictions of traffic performance, especially during periods of congestion. As a result, focus has shifted to dynamic models, such as CONTRAM (Leonard, D.R. et al. 1989), DYNASMART (Jayakrishnan, R. et al. 1994) and DYNAMIT (Ben-Akiva, M. et al. 1997).

1.1.1 Macroscopic simulation

Dynamic, meaning time-variant, modelling of the traffic flows has become common nowadays. Dynamic macroscopic models such as the LWR model (by

Lighthill and Whitham, (Lighthill, M.H. & Whitham, G.B. 1955) and Richards (Richards, P.I. 1956)) describe the evolution of traffic over time and space using a set of differential equations. Often analogues of physical phenomena are used in defining the differential equations, such as those describing traffic like flows in fluids or gases. The solution to these equations can be obtained analytically or using simulation. When evaluating a single segment of road, analytical approaches can still be used, but when the temporal and spatial interactions of traffic flows in road networks need to be evaluated, the method of solution that is used is normally simulation.

In Daganzo's Cell Transmission Model (Daganzo, C. 1994; 1995), the LWR continuum model is discretised into cells. The road is represented by a number of small sections (cells). The simulation model keeps track of the number of vehicles in each cell, and every time step it calculates the number of vehicles that cross the boundaries between adjacent cells. This flow from one cell to the other depends on how many vehicles can be sent by the upstream cell and how many can be received by the downstream cell. The amount of vehicles that can be sent is a function of the density in the upstream cell and the number that can be received depends on the density in the receiving cell. The lagged cell transmission model (Daganzo, C. 1999) is a refinement of this scheme, where the amount of vehicles that a cell can receive (from the adjacent upstream cell) depends on the density some time earlier in the cell.

Another model that uses simulation to describe the propagation of vehicle flows is METACOR (Elloumi, N. et al. 1994) (Papageorgiou, M. et al. 1989). METACOR is based on another analytical continuum model, developed by Payne (Payne, H.J. 1971). The road is divided into cells, for which at discrete time intervals the flow, speed and density are calculated evaluating macroscopic differential equations. In each time interval, also interactions of consecutive cells in terms of speed and concentration are calculated. Step by step the temporal and spatial dynamics of the traffic system are approximated by these calculations. This way the interacting cells make up a road, and roads can be connected via intersections. Because of the explicit modelling of the interactions of the *entities* of the system (in this case the road cells and intersections), and the relative ease of modelling the (small) entities, the modelled area can be extended to include large road networks.

The common term for simulations that model traffic as flows is *macroscopic* simulation. The use of these tools has grown extensively, and been facilitated by the development of extensive traffic measurement systems that have been installed in major urban areas and motorways. An additional factor that helped especially macroscopic models gain popularity is the fact that the data needed for such models (flow counts, speeds) is at the same level of aggregation as the data supplied by the measurements.

1.1.2 Microscopic simulation

While dynamic assignment in general can be studied using the macroscopic simulators, the need has arisen to understand at least part of the traffic system at a more detailed level. It has been found that 'details' at the macroscopic level, such as the length of an on-ramp or the settings of signal control, are often constraining when it comes to the maximum (capacity) and nominal flows through such sections, and the study of the vehicular interactions is needed to discover and understand such constraining factors.

Whereas the macroscopic models often exhibit a minimalist approach, so that an efficient solution can be reached, do the new generation of models aim at modelling the process of vehicular traffic in detail. This type of models, that try to describe the actions and reactions of the particles that make up the traffic as accurately as possible, are called *microscopic models.*

In microscopic models, traffic is described at the level of individual vehicles and their interaction with each other and the road infrastructure. Normally this behaviour is captured in some set of rules of behaviour which determine when a vehicle accelerates, decelerates, changes lane, but also how and when vehicles choose and change their routes to their destinations. The models that govern the vehicle's behaviour can often be divided into a car-following model, a lane-change model, and a route-choice model. The car-following model describes the breaking and accelerating patterns that result from interaction of the driver with the vehicle in front as well as other objects (such as speed limits, road curvature, etc.). The lane-changing model describes the decisions when to change lanes, based on the driver's preferences and the situation in both the current lane and other lanes (speed of vehicle in front, sufficiently large gap in adjacent lane, etc.) The route-choice model describes how drivers determine which path to take from their starting location (origin) to their destination, and how they react to traffic and route information along the way.

In addition, the traffic control to which the vehicles (drivers) react, is described in detail: the signs, traffic signals, the way these signals are operated, but also the location and operation of traffic detectors.

The demand in microscopic models is normally represented in one of two ways. One method is to model the flows of traffic that enter the network, together with the turning percentages at each intersection (i.e. the percentage of vehicles that turn left, right or go straight, for each intersection approach). Another method is to divide the modelled network up in zones and define the number of vehicles that want to travel from each zone to each other zone in an *Origin/Destination matrix (OD matrix)*. Normally these demand patterns vary over time, so there is an OD matrix for each time period. This last method of representing demand can be a more accurate representation of reality, since it describes from where to where vehicles travel. This allows for modelling the route choice explicitly (as a behavioural model).

Where in the macroscopic models the speeds, flows and densities are model variables; in the microscopic model these are aggregated measures, resulting from the interactions of all vehicles with the infrastructure and other vehicles. This means that its measurement is not unlike the measurement of these flows in the real system, with the detectors being modelled and simulated as well. As is discussed later on, there are also drawbacks to this approach, and most notably these are the amount of detail needed when modelling a road network, as well as the effort needed to calibrate the large amount of model parameters. Examples of microscopic models are VISSIM (PTV 2003) and (Fellendorf, M. 1996), AIMSUN/2 (Barcelo, J. et al. 1997), Paramics (Smith, M. et al. 1994), MITSIMLab (Toledo, T. et al. 2003) and CORSIM (US-DoT 1995).

1.1.3 Mesoscopic simulation

A third ‘class’ of traffic simulation models is gaining popularity. So-called mesoscopic models fill the gap between the aggregate level approach of macroscopic models and the individual interactions of the microscopic ones. Mesoscopic models normally describe the traffic entities at a high level of detail, but their behaviour and interactions are described at a lower level of detail.

These models can take varying forms. One form is vehicles grouped into packets, which are routed through the network (CONTRAM, (Leonard, D.R. et al. 1989)). The packet of vehicles acts as one entity and its speed on each road (link) is derived from a speed-density function defined for that link, and the density on the link at the moment of entry. The density on a link is defined as the number of vehicles per kilometre per lane. A speed-density function relates the speed of vehicles on the link to the density. If there is a lot of traffic on the link (the density is high), the speed-density function will give a low speed to the vehicles, whereas a low density will result in high speeds. The lane changes and acceleration/deceleration of vehicles is not modelled.

Another mesoscopic paradigm is that of individual vehicles that are grouped into cells which control their behaviour. The cells traverse the link and vehicles can enter and leave cells when needed, but not overtake. The speed of the vehicles is determined by the cell, not the individual drivers' decisions (DYNAMIT (Ben-Akiva, M. 1996)).

Alternatively, a queue-server approach is used in some models (DYNASMART (Jayakrishnan, R. et al. 1994), FASTLANE (Gawron, C. 1998), DTASQ (Mahut, M. 2001)), where the roadway is modelled as a queuing and a running part. The lanes can be modelled individually, but usually they are not. Although the vehicles are represented individually and maintain their individual speeds, their behaviour is not modelled in detail. The vehicles traverse the running part of the roadway with a speed that is determined using a macroscopic speed-density function, and at the downstream end a queue-server is transferring the vehicles to connecting roads. This last approach combines the advantages of dynamic disaggregated traffic stream modelling (since the vehicles are modelled individually), with the ease of calibration and use of macroscopic speed/density relationships. The capacities at the node servers follow from saturation flows and their variance (measured or calculated). Signal controlled intersections can be modelled by replacing the queue servers with gates that open and close according to the states of the signal control (green / amber / red). Adaptive signal control is harder to model since the positions of the vehicles on the link are not known, and therefore it is difficult to know when they pass detectors connected to the signal control. Another advantage of the representation of individual vehicles is the possibility of modelling disaggregated route-choice. This is important when en-route changes of routes need to be modelled, for instance when evaluating ITS systems that help drivers decide their routes.

Another type of mesoscopic model uses cellular automata. In these models the road is discretised into cells that can either be empty or occupied by a vehicle. The vehicles follow a minimalist set of behaviour rules (most notably the Nagel-Schreckenberg rules (Nagel, K. & Schreckenberg, M. 1992)), which determine for

each time step the number of cells that are traversed by the vehicle (TRANSIMS (Bush, B.W. 2000)).
The main application area of mesoscopic models is where the detail of microscopic simulation might be desirable but infeasible due to a large network, or limited resources available to be spent on the coding and debugging of the network.

1.2 Limitations of simulation models

As mentioned before, simulation models have a number of advantages over analytical models, but also some disadvantages. Some of the disadvantages are generic to simulation models. One of them is the relatively large computational cost compared to analytical solutions, another the need for calibration to conditions specific to the traffic system to which they are applied. Most disadvantages are particular to the specific type of simulation model. This section will try to assess the limitations of macroscopic and mesoscopic models on the one hand, and microscopic models on the other.

1.2.1 Limitations of macroscopic and mesoscopic simulation

While macroscopic and mesoscopic models should be easier to calibrate than microscopic models, due to their few and directly measurable parameters, their application is limited to cases where the interaction of vehicles is not crucial to the results of the simulation. For instance, analysis of merging areas usually requires the explicit simulation of gap acceptance behaviour of the vehicles, as well as a precise reproduction of the geometrical features of the ramp and freeway. Due to the high-level aggregate representation of traffic and road geometry in macroscopic (and to a lesser degree mesoscopic) models, these facilities cannot be accurately replicated and analysed. Accurate modelling of adaptive signal control is difficult in both macroscopic models and mesoscopic models. In macroscopic models, the vehicles are not modelled individually and in mesoscopic models the positions and behaviour of vehicles are approximated. When these vehicle positions are not known, or are inaccurate, it is difficult to simulate the activations of detectors that are used in the adaptive control system. This brings us to a general problem of comparing results from macroscopic models with real life detector data. While it is possible to obtain measures such as flows and speeds from these models, the exact location of the measurements can be difficult to determine. Normally they are

averages over a whole link or segment, while in real life the location of a detector can make a difference on the measured flows, speeds and occupancies.
These types of models are most successfully applied to large scale networks and long time periods, where the shortcomings due to the limited detail of these types of models may not be important. For applications where these shortcomings can have a larger effect on the results, including modelling of intelligent transport systems (ITS) and the examples mentioned above, one should consider applying microscopic models instead.

1.2.2 Limitations of microscopic simulation

The last decade has seen an increase in the number of microscopic simulation models and their application by road authorities, planners and consultants alike. The reason for their popularity has not always been their greater potential for modelling traffic processes, but also their ability to visualize the results in a realistic and appealing way.

Although microscopic simulation models have proven their value in detecting, analysing and understanding a large range of traffic problems, their application has not been without problems. A number of reports exist on problems in calibrating the models and the small number of reports on attempts to validate microscopic models speaks for itself. For an account of calibration and validation one can read (Toledo, T. et al. 2003) where the MITSIMLab model has been adapted to, calibrated to and validated against the traffic situation on the northern Stockholm arterials. (Merritt, E. 2003), (Chu, L. et al. 2004) provide other accounts of calibration and validation. What is more serious however, is the fact that such calibration efforts, while generally recognized as desirable and needed for credible simulation results, are generally considered too time-consuming and costly.

Problems applying microscopic models

Most notably the effort needed to code the road surface network has been quoted as one of the biggest problems when applying microscopic simulation. The road needs to be described in detail, such as exact lengths of sections, lane drops, lane marking, curvature, etc., and the resulting simulation is highly sensitive to errors in these details. For instance a wrongly coded ‘full stripe’ (meaning a lane-change prohibition) can cause many vehicles to reroute or not being able to make the necessary lane change for a turn, resulting in artificial queues in the simulation. But even when all efforts to avoid such (minor) errors have been made, the real world can often be represented in many ways, which makes it a matter of design experience how to code the road geometry. And again, different decisions can lead to radically different simulation results.

When the road network is coded, the signal control facilities need to be coded. Here the main problem seems to be the wide variety in control systems, as well as the limited availability of control plans or for advanced controllers even descriptions of their functioning.

However, the most important issue when applying microscopic models is dealing with their sensitivity to variations in the demand (the amount of vehicles that 'want to' enter the modelled network). A small increase in the demand on a heavily trafficked road can increase the number and severity of vehicle interactions (breaking for the vehicle ahead), which leads to a *capacity breakdown*. This is the transition from a relatively stable stream of traffic at high speed, to stop-and-go traffic. This phenomenon is common in real-life traffic and is reproduced in microscopic simulation models. However, it implies that the demand needs to be modelled very much like the real traffic situation, or the simulation results may be far from reality. The sensitivity to variations in demand is less for macroscopic and mesoscopic models because they usually do not capture this phenomenon at the same level of detail.

In most microscopic models the demand is represented by (time-variant) Origin/Destination (OD) matrices. At the same time the area that is modelled is relatively small, so that the modelled origins and destinations are not really the places where drivers start and end their trips, but rather the places where they enter and exit the area of study. This means that a modelled origin or destination is in fact an aggregation of real origins or destinations that lie outside the area of study. So when in reality a driver would take a different route, he might enter the study area at a different location. But since the vehicle trips in the OD matrix are defined for the *artificial origins and destinations* that lie on the boundaries of the study area, the simulation will not represent this type of route changes. Moreover, measurements of origin-destination traffic *at the artificial origins and destinations* do not capture the real demand between those locations, just the traffic that *happens* to enter and exit the area of study at the specific locations.

In addition, the period during which traffic planners want to study traffic systems is usually the peak hour, where the network is highly (and maybe periodically over-) saturated. This means that small fluctuations in the traffic demand and traffic flow in the network result in radically different outcomes in the situation. In the simulation, roads may become congested and queues interlock where they shouldn't or don't where they should. Obviously, any of the other mentioned sources of error will be amplified because of this.

Problems calibrating microscopic models

When a model is applied to a given location it is assumed that the parameter values are valid for that particular situation (as well as for similar situations). In order to adapt the model so that the parameters *are* valid an effort needs to be made to adapt them to the intended class of situations (e.g. motorway traffic in the Stockholm area). When this adaptation has been performed successfully the model has been *calibrated*. This adaptation of the parameter values can be performed manually, using trial and error, or automatically using some form of optimisation algorithm to find increasingly better values.

The calibration of a model is an intensive, expensive and time-consuming task. Data needs to be collected in order to find correct values for the many parameters that are part of the microscopic model. Such parameters include desired speed distributions, gap acceptance distributions, acceleration and deceleration rates (both nominal and maximum), minimum car-following gaps, etc. Obviously, some of these parameters are difficult to measure directly, and are therefore estimated using aggregate measures of effectiveness to evaluate the performance of the model with a certain parameter setting. Usually the measures of effectiveness (MOEs) are travel times on links or routes and speeds, flows and occupancies over detector stations. The values resulting from the simulation runs are compared with those from measurements.

In a systematic way the values of parameters are varied to find some optimal combination with respect to the defined MOEs. Since these measures are normally only indirectly related to the parameters in the model, no assurance can be given that the found parameter set represents the *correct* values for those characteristics, since more than one set of parameter values can result in the desired values for the MOEs.

Often, calibration of micro models does not provide very general results, meaning that the resulting set of parameters is often difficult to transfer to another location

(Toledo, T. et al. 2003), (Chu, L. et al. 2004) and (Lind, G. et al. 1999). Furthermore, the attempts to find more general parameter values in large networks usually strand because of exactly the same reason: the micro-parameters are not even transferable from one location in the network to another. It has been suggested that the most probable cause for this is a tendency of model-builders to represent features of vehicle and driver behaviour in (too) much detail, resulting in very many parameters to calibrate. In addition, the parameters are usually not independent, meaning that the effect of changes in one parameter might overlap with the effect of changes in another, which complicates the calibration set-up and the use of common mathematical optimisation procedures.

1.3 Problem Statement

While current macroscopic, mesoscopic and microscopic approaches have proven their value in analysing and planning traffic infrastructure and control, they have also shown limitations in their applicability, most of which are inherent in the nature of the models.

As described above, microscopic models have proven to be difficult and time-consuming to calibrate and difficult to apply because of their richness in parameters and their dependency on large sets of fine-grained, accurate input data. Macroscopic and mesoscopic models on the other hand have shown their ability to accurately model dynamics in traffic demand, but lack in their ability to simulate modern traffic management tools such as ITS and adaptive traffic control.

A hybrid approach could remedy these shortcomings by limiting the size of microscopic simulation to specific areas where detail is needed, reducing application and possibly calibration efforts and costs, while a larger surrounding mesoscopic network will provide realistic demand and boundary conditions.

The proposed hybrid model should have the following features:

- *Windows* of microscopic simulation *within* a mesoscopic simulation model. The reason why the surrounding assignment model should be dynamic is that only this way one can model the propagation of queues and the effects of queues on the route choice of drivers *en route*. Note that in this setting a change of route in the mesoscopic model could effectively mean a *change of entry/exit point to the microscopic area*.

- *Automatic* provision of fully dynamic input demand flows for the microscopic simulation by the mesoscopic model. This solves a major problem of microscopic simulation models: the supply of time variant Origin Destination matrices (or input flows and turning percentages) for every application. However, the problem of providing or estimating the larger dynamic OD matrix remains.

- Increased accuracy of simulation. Compared to applications that only apply a mesoscopic model, the combined model provides the higher level of detail of microscopic simulation where needed, while limiting the need for detailed network coding and calibration to the focus area.

- Naturally the computation time and memory space needed for processing a combined micro-mesoscopic network should be less than for a pure microscopic network of the same size. This allows us to increase the area of study, so that network effects of local disturbances can be studied.

- The mesoscopic and microscopic models operate in parallel, synchronously and are connected to each other. This means that one can apply the combined model as one integrated hybrid model.

Research questions

1. Is it possible to integrate Microscopic and Mesoscopic models into a hybrid traffic simulation model?
2. Is it possible to define conditions for an integrated model so that it is coherent and consistent with both submodels?
3. What would be needed to satisfy these consistency conditions?
4. Is it possible to implement the conditions and produce a working, consistent and coherent hybrid simulation model?

1.4 Objective

The objective of this thesis is to investigate the possibility of integrating mesoscopic and microscopic traffic simulation models in such a way that the result is a consistent and coherent hybrid simulation model. A framework is defined for

integrating microscopic and mesoscopic traffic simulation models. The framework is first used to identify, define and study integration conditions.
A new mesoscopic model is developed with the conditions for integration in mind, since existing mesoscopic models were deemed unsuitable for integration with microscopic models. The integration framework is then used to combine the newly developed mesoscopic model and an existing microscopic model to show the feasibility of the developed ideas. The integrated model is tested and evaluated, emphasising the correctness, consistency and effectiveness of the interfacing of the two models.

1.5 Contributions

This thesis contains the following contributions to the existing body of research[1]:

1. Mezzo, a new mesoscopic traffic simulation model is presented, which is event-based, vehicle based and simulates the forming and propagation of queues. The queue dissipation is simulated using shockwaves, thereby ensuring correct spatial and temporal location of congestion. This model is especially suited for combination with microscopic models, since it is vehicle based and event-based, but is equally well suited for stand-alone use on large networks. Due to the structure of the route-choice, the model does not need to iterate, once a set of historical link travel times and routes are generated. This means that even large networks can be simulated faster than real time.
2. A framework has been developed for integrating microscopic and mesoscopic traffic simulation models into one hybrid model. This framework contains a set of requirements that need to be met, as well as an integration architecture to meet these requirements. The issues that are presented and resolved range from consistent meso-micro boundary transitions, consistent network representation and route choice, to inter-model communication and synchronisation.
3. As part of the integration architecture, a new method of initial speed generation for vehicles entering microscopic models is presented. This method proves to be superior to the existing method and other methods tested, in that it creates minimal initial acceleration disturbances, and increases the implicit capacity of the entry links in the microscopic model. Field data supports the underlying assumptions of highly correlated speeds

[1] The results on hybrid modelling have been published in (Burghout, W. *et al.* 2004) and (Burghout, W. *et al.* 2005)

for close-following vehicles, and decreasing correlation when the time headway to the vehicle in front increases. This method is general for micro simulation models.

4. Another contribution that is part of the integration architecture is the mesoscopic virtual link, which allows for a continuous representation of the network, where the pre-trip route choice decisions are taken on the mesoscopic level. The virtual links are abstractions of the (sub)paths in the microscopic area, and for each pair of boundary entry and exit nodes there is a virtual link for each used (predetermined) microscopic subpath. This allows for a consistent representation of routes throughout the mesoscopic network, and the collection of time dependent travel times for the virtual links, which is necessary for a consistent route choice over both the meso and micro parts of the hybrid model.
5. Conversely, microscopic virtual links in the microscopic model provide the flexibility needed for en-route diversions. Microscopic virtual links represent the continued path from boundary exit nodes to the vehicles' final destinations. Using these microscopic virtual links enables vehicles in the micro model to choose alternative routes that go through a different exit node.
6. The integration framework has been implemented in a working hybrid meso-micro model (MiMe) consisting of the Mezzo mesoscopic and Mitsim microscopic simulation models. The models communicate and synchronise every micro (Mitsim) time step via a Parallel Virtual Machine (PVM), which can contain multiple physical computers. This implies that the Mezzo and Mitsim components can be run on different computers.
7. The Mezzo mesoscopic model and the MiMe hybrid model are evaluated and tested using both laboratory case studies as well as field data from a small network in the north of Stockholm (Brunnsviken). The results show a consistent behaviour at the meso-micro boundaries, even in the presence of queue propagation and dissipation.
8. The Mezzo model applied to the Brunnsviken network showed good correspondence to the measurement flows for most of the locations. The MiMe application to the same network also showed good correspondence to the measurement flows for most locations, and indicates the applicability and relevance of applying microscopic modelling in certain areas of the network where the mesoscopic modelling may be too coarse.

9. As part of the Mezzo mesoscopic model, a new algorithm for generating path set and historical link travel times is presented, where the path (choice) set for the (pre-trip) route choice and the (equilibrium) historical link travel times are generated using consecutive iterations, alternating between simulation to find new travel times (which are averaged with the previous set of travel times), and a shortest path algorithm to find new routes (given the new travel times). This method generates one or few paths for origin-destination (OD) pairs with little demand that go through uncongested parts of the network, and many paths for OD pairs that go through heavily trafficked parts of the network, provided there are sufficient alternatives. Convergence is reached when for a specified number of iterations the output travel times do not differ significantly (given a specified margin) from the input travel times, and no new routes are found.

1.6 Limitations

One of the limitations of this thesis is that it concerns mainly mesoscopic-microscopic hybrid traffic simulation models, not macroscopic-microscopic or macroscopic-mesoscopic hybrid models. In the second chapter of the thesis a number of existing macroscopic-microscopic hybrid models are discussed.

Other limitations concern the implementation of the Mezzo mesoscopic model. At this point the model is not a complete, commercial package, and there are limitations in the way signal control and route-choice is dealt with (simple multinomial Logit, based on travel times only). Moreover, the model is not yet optimised, nor sufficiently validated. The validation of a simulation model requires a large number of applications to real networks, so that its performance on different real-life facilities can be evaluated.

The implementation of the MiMe hybrid model is limited in that the microscopic virtual links (that offer the possibilities of changing exit node with en-route diversions in the micro area) are not yet implemented. This means that for the moment en-route diversions in the microscopic area are possible, but are restricted to use the same exit node to the meso area. Like the Mezzo model, the hybrid MiMe model needs to be further validated in real-life tests.

1.7 Structure of the thesis

The remainder of this thesis consists of a literature review in chapter 2, where existing hybrid traffic simulation models are discussed. Chapter 3 presents the newly developed mesoscopic model, Mezzo. In Chapter 4 the conceptual framework and implementation of the MiMe hybrid micro-meso model is presented. The conceptual framework consists of the requirements to an integrated model as well as the integration architecture, in which the ideas and theoretical structures for integrating a mesoscopic and microscopic simulation model are presented. In chapter 5 the Mezzo mesoscopic model and the MiMe hybrid model are evaluated and tested on both laboratory and real networks (from the North of Stockholm). The evaluation of the hybrid model focuses on the integration methodology and functionality of the boundary transitions. In Chapter 6 the findings in this thesis are concluded and directions for further research are identified.

Chapter 2. Literature Review

2.1 Introduction

In the previous chapter the existing types of traffic simulation models were described, as well as their merits and problems. The proposed solution would be a hybrid model that consists of a mesoscopic combined with a microscopic simulation model. In this chapter known (hybrid) combinations of traffic simulation models at different levels of detail are reviewed. The focus is on which type of models are coupled, the way in which the models are coupled and how the interfaces at the boundaries are dealt with.

First the combination of static network assignment models with dynamic simulation models is considered. In these cases the combination of the models does not make for a hybrid simulation model, but some of the issues are relevant in the following discussion on hybrid simulation models. The following two sections describe a number of attempts at hybridisation of mesoscopic/macroscopic and microscopic simulation models. The discussion focuses on the types of models involved, the hybridisation schemes employed and the reported results. The concluding section discusses the different types of hybridisation and the methods reviewed, and which lessons can be drawn from them. This discussion will be used in chapter 4 to define the requirements for the integration architecture.

2.2 Static assignment with simulation

In this section three known examples are described, where a static network assignment model is integrated with a simulation model. Since only one of the two models is a simulation model, and the other static in nature, they do not deal with most issues that are relevant in this study, but are included for the sake of completeness. In the first example, simulation is used with the assignment model, to provide better representation of the dynamically varying capacity in a certain part of the network (mostly intersections). In the second and third examples the static assignment is used to generate a base-network and provide the demand for the micro simulation model.

2.2.1 Saturn

One early example of integration of models at different levels of detail is SATURN (Vliet, D.v. & Hall, M. 1998), which combines a static assignment model over a larger network with simulation of the traffic in a small area of the network. Traffic in the buffer network follows speed/flow (or rather travel time / flow) relationships. This means that a certain (static) flow over a link corresponds to a certain travel time. The routes that traffic takes along the network are decided by the expected travel time on links, which in its turn is dependent on the amount of traffic (flow) that wants to use those links. Unsurprisingly, lower travel times are preferred over higher ones.

By iterating the route assignment and calculation of the resulting flows on the links in the network, a user equilibrium is found, where the travel times on the links cannot be decreased by change of routes (Stochastic User Equilibrium or elastic assignment) (Wardrop, J.G. 1952).

In the simulation network, links are represented by the number of lanes, length, and travel time. This travel time is usually the free-flow travel time, but may also be produced using a speed/flow function. The turning movements are modelled by flow-dependent saturation flows, and their operation is simulated using arrival flow patterns, opposing flows, and gap acceptance. The results of the simulation are the junction delays, which are input to the assignment part of SATURN for the next iteration.

The iterations with consecutive simulations, assignments, simulations, etc. terminate when the convergence conditions are met. In simple terms, this means that the route choices and travel times on both the static part of the network and the simulated part, do not change much from one iteration to the other. The simulation model that is part of the SATURN package would nowadays be considered a macroscopic or maybe mesoscopic simulation model.

2.2.2 Aimsun/2 and Emme/2

A more recent integration of a static assignment model with traffic simulation is the combination of AIMSUN/2 and EMME/2 (Montero, L. et al. 1998). EMME/2 (INRO-Consultants 1996) is a macroscopic static assignment model that uses volume/delay functions and a user-equilibrium assignment in much the same fashion as the SATURN assignment part. AIMSUN/2 (Barcelo, J. et al. 1997) is a microscopic simulation model that simulates traffic according to behavioural rules (see Chapter 1). The *traffic demand* in AIMSUN/2 can be one of two forms. It can be the input flows (flows on the incoming links at the boundary of the network) and turning percentages (the percentage of vehicles turning left, right or going straight for each intersection). Or it can be provided in the form of an Origin / Destination matrix (OD matrix). In order to represent the variation of the demand over time, an OD matrix normally has slices for each time interval in the study period.

The integration of the two models consists of the consecutive operation of the two models in a fashion alike the operation of one iteration of SATURN. For the input from EMME/2 to AIMSUN/2, the link and turning flows can be used to generate the input flows and turning percentages. Alternatively, the time-sliced OD matrices from EMME/2 (optionally updated using matrix estimation macro with field measurements of link-counts) are generated interactively with the user, who specifies the relevant time interval for each OD-matrix slice and the vehicle type. Also, EMME/2 can calculate a transversal matrix for a micro simulation sub-area.

In contrast to the operation of SATURN, the resulting travel times on simulated links are not fed into the EMME/2 assignment for a new round of assignment and simulation. This means that the demand in the micro simulation model is not dependent on the traffic performance on the links in the micro simulation area.

2.2.3 Visum and Vissim

PTV's microscopic simulation model VISSIM (PTV 2003) can use (sub)network topology, and OD demand matrices that are exported from the Visum static assignment model in much the same way as in the Emme/2 – Aimsun/2 combination. In addition a routes set can be imported from Visum, which may improve the quality of the Vissim simulation. As in the case of Emme/2-Aimsun the resulting travel times are not fed back into Visum for iterated simulation-assignment (PTV 2004).

2.3 Mesoscopic with microscopic simulation

2.3.1 Paramics and Dynasmart

The microscopic traffic simulation model PARAMICS (Smith, M. et al. 1994) and the mesoscopic model DYNASMART (Jayakrishnan, R. et al. 1994) have been combined in an embedded structure (Oh, J.-S. et al.) and (Jayakrishnan, R. et al. 2001). Primary objective of the combination was to provide a better route-choice capability for the micro simulation, by using the DYNASMART path dynamics.

The authors argue that the route-choice in micro simulation models is inadequate to deal with large networks, mainly because of the high level of detail in network representation, which leads to complex path computations. One reason for the high level of detail is that changes in road geometry often require multiple links (and thus nodes) to represent a single road.

Mesoscopic networks (Such as in DYNASMART) on the other hand, have a more coarse representation of the network, and the authors argue that the nodes in a mesoscopic network description correspond more or less to the decision points in drivers' route choice behaviour. In addition, DYNASMART has been developed explicitly to incorporate path dynamics, route guidance and driver decisions in a realistic way.

In the integrated structure, the route-choice is effectively taken out of PARAMICS and replaced by route-choices from DYNASMART. In order to do this, the PARAMICS network is fed in simplified (abstract) form to DYNASMART. The abstracted network is generated by elimination of all nodes that have only one incoming and one outgoing link. In addition, the vehicle positions and link costs are communicated to DYNASMART, which takes care of the routes and the route-choice behaviour. These route choices and routes are then fed back to PARAMICS.

This type of hybrid model does not combine the different simulation models to simulate part of the network in one and part of the network in the other model, but combines the path dynamics and route choice of DYNASMART with the microscopic simulation of PARAMICS.

2.3.2 Metropolis and MITSIMLab

In (Nizard, L. 2002) a combination of the microscopic traffic simulator MITSIMLab (Mitsim) (Ben-Akiva, M. et al. 1997) and the mesoscopic simulator Metropolis (Parma, A.d. et al. 1996) is described. First the mesoscopic and microscopic simulations are run on the same (large) network to calibrate the link performance (travel time) functions of Metropolis to the travel times produced by Mitsim. Then the mesoscopic simulation is run on a larger network, and the microscopic simulation on an inner subnetwork.

The report describes how the network and demand representation of Metropolis are mapped to Mitsim. The link exit flows from Metropolis into the Mitsim subnetwork are used to create a time-dependent OD matrix that is input to Mitsim, which is then run on the network. The resulting travel-times are then used to create new link travel time functions for the links in Metropolis. This way the performance of the Metropolis links are calibrated to the travel times produced by Mitsim. This procedure is iterated until the travel times produced by Mitsim and Metropolis correspond.

The study reports some problems in ensuring the conservation of vehicles (i.e. no vehicles get lost nor are 'extra' vehicles created) across the boundaries between the models. It also reports problems in constructing monotonously increasing travel time functions (from the travel times in Mitsim), which are required by Metropolis.

When the travel time functions are calibrated, the network (N) is divided into a small sub network (S) and a larger outer network (N\S). The outer network is simulated by Metropolis and the subnetwork by Mitsim. First Metropolis is run on the entire network (N), and the link flows into the subnetwork (S) are recorded. Then Mitsim is run on the subnetwork, using the newly produced link flows. Finally the flows exiting Mitsim are input to the simulation of Metropolis on the outer network (N\S). The travel times in the combined simulation are compared with the travel times that are produced by Mitsim and Metropolis alone on N. As may be expected the resulting link travel times when running only Mitsim and only Metropolis on N are quite far apart, whereas the combined simulation gives travel times that are in the middle between those produced by Mitsim and Metropolis alone.

2.3.3 Transmodeler

Transmodeler (Yang, Q. & Slavin, H. 2002) is a new traffic simulation model under development that builds on Caliper Corporation's TransCAD GIS technology and includes simulation on macroscopic, mesoscopic and microscopic levels. Networks and OD matrices can be imported from the static assignment and GIS component.

Users can choose a selection of links and define them to be either macroscopic, mesoscopic or microscopic, and the appropriate traffic models are used to simulate the behaviour of traffic in the network. Alternatively, the different levels of fidelity (macroscopic, mesoscopic or microscopic) can be assigned based on link type, or a user-defined formula. The microscopic, mesoscopic and macroscopic models run together at different time step sizes and different parts of the network, and the model that runs fastest has to wait for the slower models to finish at the end of each time step.

In microscopic segments, the vehicles calculate their acceleration rates and make decisions on lane changes, as you would find in other microscopic models. Transit systems and actuated traffic control are modelled as well.

The mesoscopic model is also vehicle based, but does not represent explicitly the lanes vehicles are in. The position of vehicles is tracked, and the vehicles move in groups called "traffic cells". Speed-density relationships determine the speed of the traffic cells. Traffic cells heading for the same downstream link are grouped into 'traffic streams'. Traffic streams in the same segment may impede each other, and queues are modelled explicitly and may spillback from downstream links.

The macroscopic model uses an aggregated delay function similar to those used in static assignment models to compute the average travel time of vehicles over a segment, and for various turning movements at the downstream end of the link.

The routing is common for the entire model, and vehicles are polymorphic, in that they change their type (macroscopic, mesoscopic or microscopic) dynamically as they move over different parts of the network. No mention has been made (yet) of how coupling issues between the microscopic, mesoscopic and macroscopic parts are resolved. As this model is currently being developed, and most of the information described above is obtained from personal communication, the exact working of the model may differ from the information provided here.

2.4 Macroscopic with microscopic simulation

2.4.1 Pelops and Simone

The microscopic traffic simulator PELOPS (Cremer, M. & Meissner, F. 1993) and the macroscopic simulator SIMONE (Hochstaedter, A. et al. 1999) have been coupled into an integrated model (Lerner, G. et al. 2000), (Kates, R. & Poschinger, A. 2000) and (Poschinger, A. et al. 2000). This coupling was one of the first attempts at an 'in-the-loop' integration of the microscopic simulation into the macroscopic buffer network. The articles focus on the coupling between the models, consisting on the one hand of a stochastic disaggregation algorithm connecting the macroscopic to the microscopic simulation, and on the other hand an aggregation algorithm connecting the microscopic simulation to the macroscopic model.

The aggregation algorithm synchronises the larger time step of the macro model to a number (n) of time steps in the micro model. An Euler integration method is used to couple and combine the information needed in each time step of the respective models by considering the last time-step as a predictor of the information needed in the next. With regard to disaggregation at the entrances to the microscopic model the article lists a number of important microscopic characteristics to generate:

- Distributions of time gaps
- Speed fluctuations
- Autocorrelation of velocity time-series

The article focuses on the generation of autocorrelated speeds, while taking deterministic time gaps to which some random noise is added. They discuss the ways in which speed data could be produced and mention the two obvious alternatives that have been applied before:

- Use deterministic speeds and an extra stretch of road in the microscopic simulation where the vehicles can redistribute themselves. Apart from the obvious problem that in a coupled environment one cannot simply insert a 'sufficiently long stretch of dummy road', the statistical properties are rather unfavourable.
- Ignore the autocorrelations: see discussion above. The model coupling will generate unnecessary and artificial friction by inducing braking, overtaking etc. As is discussed in chapter 4, this artificial friction causes the capacity of the entry links to be reduced substantially.

Instead they propose a first order integrated moving average process to generate auto-correlated speeds using Ferrari's (Ferrari, P. 1988) model:

$$v_i = v_{i-1} + l a_i - (1 - l) a_{i-1} \tag{1}$$

Where,

a_i are drawn from a normal distribution with mean 0 and standard deviation σ_a

λ moving average parameter

The parameters σ_a and λ were estimated from individual vehicle data. This solution provides correlated speeds, but they are independent of the time gaps. As is discussed later on, the level of correlation of speeds is high for small time gaps, but decreases when the time gap increases. By ignoring the dependence of speed correlation on time gaps, the method results in speeds that are more correlated than reality for large gaps and less correlated than in reality for small gaps. The latter effect will still induce artificial braking and acceleration of vehicles entering the microscopic area.

2.4.2 Micmac

The SITRA B+ (Gabard, J.F. & Breheret, L. 2000) microscopic model and SIMRES macroscopic model are coupled in a prototype interface MICMAC (Magne, L. et al. 2000). SIMRES is based on METANET (Messmer, A. & Papageorgiou, M. 1990), and models the road in discrete cells, for which the flow, density (or concentration) and speed are calculated for each time step T. The article makes special notice of the compatibility between the car-following model that is used in the microscopic simulation and the macroscopic model that is used. By observing experimental flow/density measurement data and the fundamental diagram, the authors state that any model needs to satisfy the following constraints:

(1) $Q(K)=Q_{max}$ at $K= K_{max}$
(2) $dQ/dK=0$ at $K=K_{max}$
(3) $Q(K)=0$ at $K=0$
(4) $dQ/dK=V_l$ at $K=0$, and
(5) $Q(K)=0$ at $K= K_{jam}$.

Where:

Q	= Flow (veh/h)
Q_{max}	= Maximum flow
K	= Density (veh/km)
K_{max}	= Density at maximum flow Q_{max}
V_l	= Speed limit

Constraint (1) states that the flow/density function reaches its maximum (capacity) at a density of K_{max}. Constraint (2) states that at K_{max} the slope of the flow/density function is flat (zero). According to constraint (3) the flow at zero density should be zero. Constraint (4) states that the slope of the flow/density function is equal to the speed limit (or free-flow speed in this case) at zero density. Constraint (5) states that the flow at jam density equals zero.

It is argued that if both models observe these constraints the macroscopic model and the microscopic model are equivalent in steady state. The authors observe that most existing macroscopic models observe these equations, and that most microscopic models don't.

The original car following model used in the microscopic SITRA B+ does not satisfy the constraints mentioned above. In particular constraints (2) and (4) are violated, since the maximum speed at inifinite headway (at K=0) is infinite. The authors then adapt the car-following model by introducing the speed limit and a minimum density at which the speed becomes a function of the headway (or density).

The article continues with a description of how the two models are interfaced. The disaggregation from the last macroscopic cell to the first microscopic segment uses a Poisson distribution for the time gaps with the mean=1/Q(K,T) from the macroscopic segment, where Q(K,T) is the flow for time period T, given the density k. The speeds for the vehicles entering the microscopic area is derived from a truncated normal distribution with mean =V(K,T) from the upstream macroscopic cell (no mention is made as to what determines the standard deviation), where V(K,T) is the (average) speed in the downstream cell, given the density k and time period T. For the lane choice a binomial distribution is used, based on the initial speed of the vehicle and the density. The path of the vehicle downstream is not taken into account in the lane assignment, which may lead to unnecessary lane-changes if vehicles need to change lanes immediately after entering, in order to continue on their paths. Since the Poisson process used for the generation of the vehicles may produce too many or too few vehicles in a given time period (due to the randomness of the process), each time step the mean of the process is adjusted to account for deviations from inflow from the macroscopic segment in the previous periods. Finally the density in the microscopic segment is calculated and propagated upstream to the macroscopic cell.

The aggregation process takes the (average) speed, flow and density in the last microscopic segment and communicates it to the downstream macroscopic cell. In the microscopic segment a virtual vehicle is added with speed=V(K,T) from the macroscopic segment. The exiting vehicles that have no leader follow this virtual vehicle.

The synchronisation of the models is sequential in the sense that first an iteration of the macroscopic model is run, followed by n iterations of the microscopic model, followed by a new iteration of the macroscopic model, etc. The data is communicated via files in a temporary directory. It should be noted that the large aggregation time (macroscopic iteration interval) would smooth over most disturbances created at the boundaries of the two models.

To show the feasibility of the hybrid model, two experiments are done using a single link that consists of a macroscopic segment followed by a microscopic segment, which is followed by a macroscopic segment. It is 3-lane road, and the demand is 4000 vehicles in the first hour and 2000 vehicles in the second. The second scenario introduces an incident, which blocks the exit segment. The results are compared to those of the macroscopic simulator by itself, and show that with the original car-following model the results from hybrid model deviate from those produced by the macro model, but that with the modified car-following model the hybrid model produces results that are close to the macroscopic model by itself.

2.4.3 Hystra

Another hybrid model combining macroscopic and microscopic models is Hystra (Bourrel, E. 2003) , (Bourrel, E. & Lesort, J.-B. 2003), (Bourrel, E. & Henn, V. 2002) As in the MICMAC model, the Hystra model attempts to use equivalent microscopic and macroscopic models. Whereas MICMAC uses the existing SIMRES macroscopic model and adapts the SITRA B+ model to fit some observed conditions, does Hystra use a micro- and macro model that are both based on the LWR traffic flow theory (Lighthill-Whitham-Richards (Lighthill, M.H. & Whitham, G.B. 1955), (Richards, P.I. 1956) which consists of the following three equations:

$$\frac{\partial K}{\partial t}+\frac{\partial Q}{\partial x}=0 \qquad \text{(conservation equation)} \qquad (2)$$

$$Q = KV \qquad \text{(definition of relation of flow to average speed)} \qquad (3)$$

$$Q = Q_e(K) \qquad \text{(equilibrium relation between flow and density)} \qquad (4)$$

Where Q is flow, K is density and V is average speed.

For the macro model the Strada (Buisson, C. et al. 1996) model is used, which is a space-time discretisation of the LWR equations, much like Daganzo's cell transmission model (Daganzo, C. 1994),(Daganzo, C. 1995) that was described in chapter 1.

For the car-following model two different models have been tried. Initially an 'ideal' model was used that is based on a model by (DelCastillo, J.M. 1996). It determines the speed of a vehicle as follows:

$$V_{n+1}(t + \partial t) = V_n(t) \qquad (4)$$

Where

V_{n+1} (t+ dt) = Speed of vehicle n+1 at time t+ dt.

V_n (t) = Speed of vehicle n at time t.

dt = The delay with which the speed 'disturbance' of vehicle n propagates to vehicle n+1

The difficulty with this model is to calculate the delay dt, which varies according to the situation. The authors determine this by calculating the speed with which the shockwave travels, given the speeds of vehicles n and n+1 and their spacing. (See further (Bourrel, E. & Henn, V. 2002) for exact formulation of the shockwave speeds and response time calculation). The authors report a 'perfect fit' with the macroscopic representation, but note that this model is deterministic and any attempt to extend it (such as introducing randomness in the headways or speeds) invalidates the equivalence of the micro and macro models, and the conservation of flow across the boundaries is difficult to guarantee. The authors conclude therefore that this model is of very little practical use.

Therefore in (Bourrel, E. 2003) and (Bourrel, E. & Lesort, J.-B. 2003) the Newell (Newell, G.F. 1961) optimal velocity model is used, which determines the speed of a vehicle as a function of the spacing between the vehicle and the lead vehicle (with a certain delay dt):

$$V_{n+1}(t + \partial t) = G(X_{n+1}(t) - X_n(t)) \qquad (5)$$

Where

X_n(t) = position of vehicle n at time t

G = a function that determines the speed as a function of the vehicle spacing.

Since density is the inverse of (average) vehicle spacing, with small dt, an equilibrium speed-density relation can be used that is equivalent to the equilibrium flow density relation in the LWR. Therefore the model produces (with small enough dt) results that are very close to those of the LWR theory. The authors show that this model can be extended with stochastic time headway generation and desired speeds. The stochastic desired speeds result in individual functions G for different vehicles. Due to the fact that the optimum velocity model allows for unlimited acceleration/deceleration (since the function G operated on spacing *only*), the generation of speeds and time headways independently of each other poses no problem of artificial friction here.

Both macro and micro models are discrete-time models, and the time step for the macroscopic model is set to be a multiple of the microscopic time step. At each macroscopic time step the two models exchange boundary information.
Because of problems with the transitions between the two models' boundaries, a transition cell is introduced at each boundary. This transition cell operates in the way that both the aggregate (density, flow) and disaggregate values (vehicle positions, speeds) are computed with the respective models. This way the boundary conditions for the upstream macroscopic cell are provided (the exit flow and density), as well as the vehicle trajectories needed in the downstream microscopic segment. The vehicle trajectories in the transition cell are adapted to fit the macroscopic flow and density values.

The disaggregation of the flows into vehicles in the transition cell upstream of the microscopic segment works by generating with uniform time intervals the number of vehicles that represents the entering flow from the macro model during the macroscopic time step. Each vehicle is entered into the cell if there is the minimum space $s_{min} = 1/K_{max}$ with a speed that is derived from the equilibrium speed / density relation that both models observe. If there is not enough space, the entry will be delayed until there is. This method has a drawback that under congested situations the vehicles are generated with a speed equal to 0, and therefore require a sufficiently long transition cell to acquire a realistic speed.

In the transition cell connecting the microscopic to the macroscopic segment, the correct exit conditions for the microscopic cell upstream are given by a 'ghost vehicle' in a similar fashion as the virtual vehicle described in MICMAC. The last vehicles on the microscopic segment follow this vehicle, which has a fictive location on the macroscopic cell. Its speed is calculated from the density on that cell and the equilibrium speed/density relation, and its position by extrapolating the last exited

vehicle (as if it would have travelled with the equilibrium speed ever since it entered the macroscopic cell).
Since the macroscopic simulation downstream defines a maximum number of vehicles that can enter the segment during a macroscopic time step (which is the aggregation step), the ghost vehicle is not enough to ensure this upper boundary of the exit flow is observed. In addition a minimum exit gap is calculated as the inverse of the imposed exit flow by the macro model downstream. Using this minimum exit gap, vehicles that would exit with a gap smaller than this, are delayed. This should ensure the proper propagation upstream of congestion.

The model is tested using a single lane of macro cells with one micro cell and transition cells in the middle. The flows are compared to the flows that the macro model (STRADA) produces by itself.

2.4.4 Micro-Macro Link

Helbing et al. describe in (Helbing, D. et al. 2002), (Hennecke, A. et al. 2000) a car-following model (IDM) and a macroscopic model (GKT) that are both derived from the same gas-kinetic differential equations. The functioning and compatibility of the two models is described in detail and mention is made of an in-the-loop simulation of these two models on a single link. In a section Micro-Macro link they describe how from most simple car-following models an equivalent gas-kinetic model can be derived by interpolation of the spatial-temporal velocities and densities. Unfortunately the methods used to ensure correct boundary transitions are not mentioned, nor do the authors refer to another publication where they are described2. From the graphs that show multiple configurations of macroscopic and microscopic cells, the propagation of flows and densities seem consistent over the boundaries. An interesting note is the comment by the authors that for the simulation of an on-ramp merging scenario, all cells are in microscopic mode, except for the on-ramp merging sections. The authors argue that:

"While traffic simulations with microscopic single-lane models are more intuitive and detailed than macroscopic simulations, a microscopic

[2] The article (Helbing, D. *et al.* 2002) says: "the formulation of dynamic interface conditions is a particularly tricky task which cannot be discussed here". Personal communication with the authors revealed that the boundary interface conditions are not and will not be published.

implementation of on- and off-ramps would require rather complex multi-lane models with lane-changing rules. Ramps can be treated much easier by macroscopic models, where they just enter by a simple source term in the continuity equation." (Helbing, D. et al. 2002)

Usually the reason for combining microscopic and macroscopic models is the fact that microscopic models represent such facilities as on-ramp merging much better than the macroscopic ones, since the lane-changing and gap-acceptance behaviour is modelled explicitly. Here the authors seem to acknowledge the difficulty of extending the single-lane IDM model to incorporate lane changing and gap-acceptance, and still have a formulation that is equivalent with the macroscopic model.

2.4.5 SmartCAP and SmartAHS

The microscopic simulator for automated highway systems (AHS) SmartAHS (Antoniotti, M. et al. 1998) is currently being interfaced with the SmartCAP (Broucke, M. et al. 1996) mesoscopic (according to the taxonomy introduced in chapter 1 it would classify as macroscopic) simulator (Horowitz, R.e.a. 2002). SmartAHS has a typical microscopic simulation model that describes the vehicles' movements in detail, and SmartCAP describes traffic in terms of continuous flows that propagate through the network. On top of the traffic simulation models, both contain an additional control module for the automated highway systems. At the boundaries of the two models transition zones are placed. At the macro to micro boundary the continuous flows are disaggregated into 'ghost platoons' of microscopic vehicles, and at the micro to macro boundary the 'ghost vehicles' are projected in front of the exiting microscopic vehicles, in a way similar to the virtual vehicles in MICMAC and HYSTRA.. The ghost platoons are generated and placed onto the transition zone upstream of the micro area, to ensure that a proper dispersion of the vehicles entering the micro area. At this moment, to the best of the author's knowledge, no details of the algorithms for aggregation and disaggregation are published.

2.5 Discussion and lessons learned

In the 1950's (Gazis, D.C. et al. 1959) showed how car-following models can be integrated to obtain macroscopic speed/density equations. One example is the non-linear General Motors car-following model (Gazis, D.C. et al. 1961), which has been used in many microscopic simulation models. These car-following models use a single formulation to describe the acceleration and deceleration behaviour of vehicles.

However, nowadays most microscopic models use different regimes for different car-following situations such as emergency braking, free-flow driving, merging and normal car-following ((PTV 2003), (Barcelo, J. et al. 1997), (Ben-Akiva, M. et al. 1997)). Each of these regimes usually has its own model governing the behaviour, and in many cases acceleration and deceleration situations are treated with different models (or parameters) as well. Therefore it has become very difficult to derive macroscopic speed/density relations from these behavioural models.

In addition, the steady-state equivalence between microscopic and macroscopic models that can be obtained by deriving a micro model from the macro equations applies only to the car-following model per-se. A micro model has many more components such as lane-changing and gap-acceptance, which cannot be included in the derivation. This means that nothing can be said of the equivalence of the complete microscopic model and the macroscopic model, based on their formulations.

Thus even though hybrid models such as Hystra and Micro-Macro Link are attempting to create an equivalency between the macro and micro representations by deriving a microscopic model from the macroscopic equations, it is difficult to establish such a relation, especially under non-steady state conditions. In addition, the approach of deriving micro models from macro equations limits which microscopic models can be derived, and especially prohibits the multi-regime approach, which is now employed by most microscopic models to approach real driver behaviour as closely as possible. In fact, in the case of Hystra it is shown that attempts to extend the deterministic car following model to include stochastic desired speeds and time intervals already pose problems to the coupling scheme. Further extensions to include basic micro model components such as lane changing, multiple vehicle classes and limits on acceleration/deceleration should also prove difficult with this type of model.

In the Micmac approach, a number of conditions are defined to which the fundamental diagrams of both macro and micro models need to conform. For the same reasons as pointed out above, the theoretical conformance of an entire

microscopic model cannot be established a priori, only that of the car-following model. When other crucial components such as gap-acceptance and lane-changing are taken into account, it is impossible to establish theoretical correspondence.
Instead, the approach taken in the Mitsim-Metropolis combination seems more viable. Here the link-cost functions of Metropolis are successively calibrated to the 'observed' travel times from Mitsim. By calibrating the two models to each other a practical equivalency can be established, which may be more useful since it compares the performances of the two models as a whole.

Most of the hybrid models described focus on the issues of aggregation/ disaggregation of traffic at the boundaries between submodels. In the case of the macroscopic models the continuous flow representation of traffic in the macro model is very different from the vehicle representation in the microscopic model. This is the main cause of the need for detailed aggregation/disaggregation procedures. In mesoscopic models the traffic is represented using vehicles (albeit with simplified behaviour), which avoids most of the problems in disaggregation and aggregation, such as discretisation of flows into vehicles, and generation of arrival time headways at the boundaries. An added advantage of using a vehicle representation throughout the entire hybrid model is that it allows the use of uniform route-choice and other driver decision models.
Some important aspects in combining a macro or meso model with a micro model have not yet been discussed in literature. Most of the work seems to have focused on aggregation-disaggregation issues and theoretical model equivalence. What is missing is a view of the entire range of issues concerning the combination of two models. As is discussed in Chapter 4, the integration of network representation to represent both detailed microscopic and coarser mesoscopic networks it not trivial. Furthermore, the integrated representation of route-choice is also an important aspect that thus far has not been discussed.

Chapter 3. Mezzo: a mesoscopic simulation model

3.1 Introduction

As described in chapter 1, the existing traffic simulation models can be subdivided into three classes: macroscopic, mesoscopic and microscopic. The macroscopic models represent traffic as a (indivisible) flow, whereas the microscopic models describe the behaviour of individual drivers, their vehicles and their interactions. The mesoscopic model class contains a number of methods of representing traffic that hold the middle between the macroscopic and microscopic representation of traffic.

One such method is the one used in Contram (Leonard, D.R. et al. 1989), in which the network is represented by nodes and links, and the vehicles on those links are grouped into packets that travel from origins to destinations (although packets can consist of only one vehicle). DYNAMIT (Ben-Akiva, M. 1996) on the other hand uses a cell model for vehicle movements along the link. A cell can be considered a platoon of vehicles and travels with a certain speed over the link. The vehicles can move between cells, but cannot overtake. A third paradigm is the modelling of individual vehicles that follow macroscopic speed/density relationships on the link, with a queue-server at the nodes to account for delays caused by traffic signals and the interaction with traffic from other directions. Examples of such models are DTASQ/MFT (Mahut, M. et al. 2002), FASTLANE (Gawron, C. 1998) and DYNASMART (Jayakrishnan, R. et al. 1994)

This queue-server paradigm with a link model where the speed is a function of the density on the link is especially interesting, since it combines the queue-server approach, which models well the friction in bottlenecks such as intersections and on/off ramps, with the dependency of traffic speed on the density of traffic directly in front. This is the approach taken in the newly developed mesoscopic model Mezzo.

3.2 Why Yet Another Simulation Model?

As we have seen, there already exist a number of mesoscopic models, so why not integrate one of them with a microscopic model?

The most important reason for choosing to develop a new simulation model was that none of the existing models were deemed suitable to be integrated/interfaced with a microscopic model. As discussed before and later on in this thesis, there are a large number of issues to be tackled when combining two simulation models.

For a mesoscopic model to be a candidate for integration in a hybrid model it needs to be:

- *Easily adaptable*. This requires usually access to the program source code, so that changes can be made as needed in the integration process. This is one of the largest obstacles when using 3rd party models such as Contram or Dynasmart, since usually the owners are not willing to give out the source code.
- *Able to communicate with a micro model*. This requires that the model can be easily extended to communicate with another model. In terms of the operation of traffic assignment, it needs to be free of internal iterations such as those in CONTRAM. In the latter type of models, iterations are needed *within one simulation run* to obtain a correct assignment. These iterations complicate the interoperation of such a model with another (microscopic) model.
- *Able to synchronize with a micro model*. Most mesoscopic and virtually all microscopic simulation models calculate the changes over time in small time-steps. In the case that both models use time-steps, these need to be synchronised, and in case they use different time-step sizes (microscopic models use typically 0.1 second, and mesoscopic models 60-300 seconds) the traffic that flows between these two models needs to be aggregated and disaggregated. As discussed in chapter 2, these aggregation / disaggregation processes can pose difficulties in providing correct translation of traffic dynamics across the meso-micro boundaries. However, with an event-based mesoscopic model, the synchronisation and (dis-) aggregation problems disappear. An event-based model has a continuous representation of time, and only the moments of *change* in the system state are modelled: events. The synchronisation of an event-based mesoscopic model to a time-step microscopic model is easy, since the time-steps of the microscopic model are considered (extra) events in the mesoscopic model. Since only one time step is used (from the microscopic model) there is no aggregation-

disaggregation problem. A more detailed description of event-based versus time-stepped simulation follows later in this chapter (section 3.9.1).

- *Compatible with microscopic models in representation of traffic.* This property is inherent in our definition of mesoscopic models, but is worth mentioning, since most current work on hybrid models involves macroscopic and microscopic models. As mentioned before, microscopic models represent traffic as separate vehicles with their own behaviour. When such models are combined with a more aggregate model (mesoscopic or macroscopic), it is advantageous to choose a model that also represents traffic in terms of *vehicles*. In the other case (macroscopic models) the traffic is represented as a *continuous flow* which needs to be disaggregated (divided) into separate vehicles at the boundary into the microscopic model, and aggregated (combined) at the boundary from the microscopic into the macroscopic model. As seen in chapter 2, this aggregation / disaggregation is a difficult subject in itself, and presents a large obstacle in the integration of the two models.

3.3 Network Structure

The traffic network in Mezzo is represented by a graph that consists of nodes and links. The nodes are the points where multiple traffic streams join or diverge, such as intersections, on / off ramps, as well as origins or destinations of traffic. Links represent the roadway between such nodes, and are unidirectional. This means that a regular street is usually represented by two links, one for each direction. The lanes on a road are not represented separately. Nodes have usually multiple incoming and outgoing links, and are considered the main sources of friction in the traffic streams. See figure 1 for an object model of the node-link structure of Mezzo. See Appendix A for a complete Object Model of the structure of Mezzo, and Appendix B for a short introduction to Object Model notation.

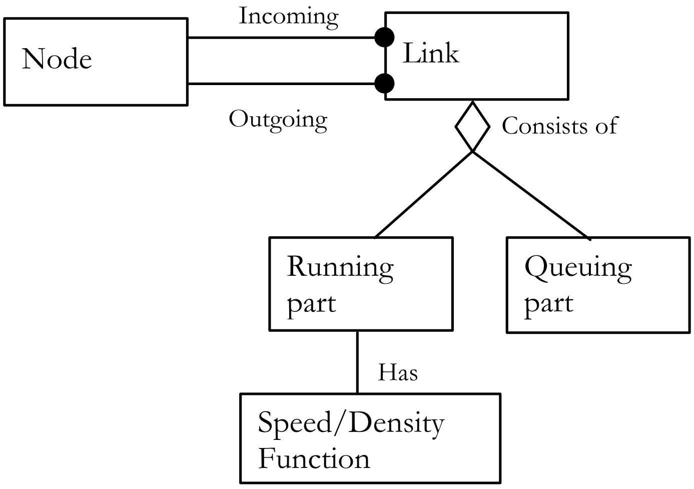


Figure 1. Object model of the Mezzo network structure. See Appendix B for legend of Object Model symbols and notation

3.3.1 The Link Model

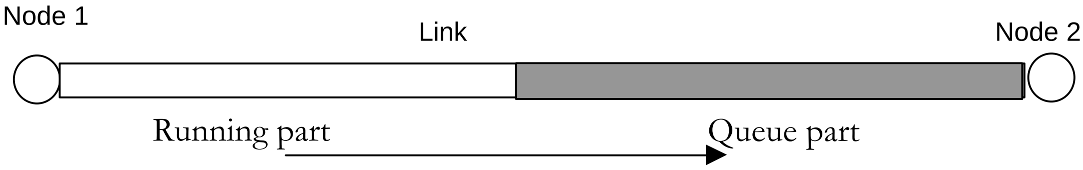


Figure 2. Representation of a link in Mezzo.

As can be seen in figures 1 and 2, the link is divided into two parts. The running part and the queue part. The *queue part* starts at the downstream node and grows towards the upstream node, when the incoming flow exceeds the outgoing flow on the link. For instance, when a traffic light at the downstream node goes to red, the queue part will grow.

The *running part* is the part of the link that contains vehicles that are on their way to the downstream node, *but are not (yet) delayed by the downstream capacity limit (for instance the traffic light)*. This means that the boundary between the running part and queue part is dynamic, and usually varies over time, depending on the variations in the inflow and outflow. In the case of an empty link, there is no queue, and the running part occupies the whole link. Conversely, if the whole link is full, the queue occupies the whole link, and there is no running part.

When a vehicle enters a link, it enters the running part with a speed that is a function of the density (how many vehicles per kilometre per lane) on that running part. A low density (only a few vehicles) results in a high speed and a high density in a low speed. This speed is used to calculate *the earliest time a vehicle can reach the downstream node (earliest exit time).* The vehicles on the link are ordered according to their earliest exit time, and this defines which vehicles are in the queue part at any point in time.

The queue part is defined to contain at time t, *those vehicles that have an earliest exit time smaller than t.* In other words, all vehicles that should have exited the link, if it were not for some delay caused at the downstream node.

It is important to note that the density is calculated on the running part only. This is needed to ensure that there is no double counting of the delay a vehicle experiences. If a vehicle is in the queue, it is safe to assume that the speed with which it advances is determined by the speed of the queue dissipation, which is normally a function of what happens at the down stream node. The delay caused by interactions with vehicles in front (which is what the speed/density relation captures) *is already included in the queue delay.* If the density of the whole link were used for the speed/density function that determines the speed of vehicles, the delay caused by the queued vehicles would be *counted doubly:* firstly in the traversal speed (which is low due to the high density in the queue) and secondly in the queue delay.

3.3.2 The Node model

Turning movements

At the downstream end of the link, the node connects it to other links. Each connection of an incoming to an outgoing link is called *turning movement* (even the straight going movements). See Figure 3 for an example.

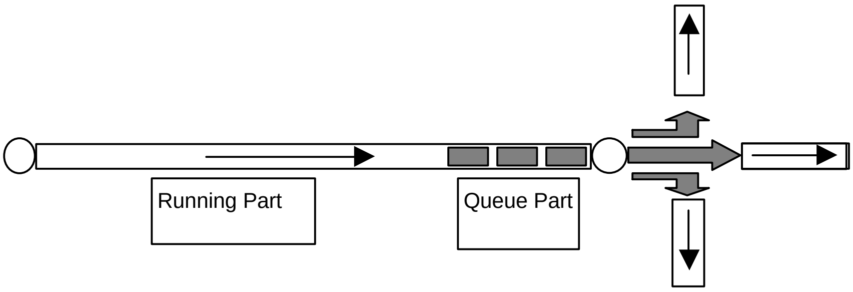


Figure 3. Three turning movements connecting one incoming link to three outgoing links.

These turning movements have a limited capacity for through-putting vehicles to the following links. In real traffic, this capacity may depend on various factors. In the case of a signalised intersection, these are mainly the green time that is allocated to the turning movement, the number of incoming and outgoing lanes, possible interactions with opposing turning movements (for instance a vehicles in a left turn that wait for the traffic going straight ahead in the opposite direction) and queuing vehicles from other turning movements blocking the access.

In the model the capacity of the turning movements is represented by queue-servers. The vehicles that are part of the queue segment are picked one-by-one by such a server and transferred to the next link, if there is room for them on that link. Each turning movement has an independent server that only picks the vehicles whose route requires them to make that turning. Figure 4 shows three turning movements: straight, left and right turn. The server for the vehicles moving straight cannot process more vehicles until the destination link has become unblocked.

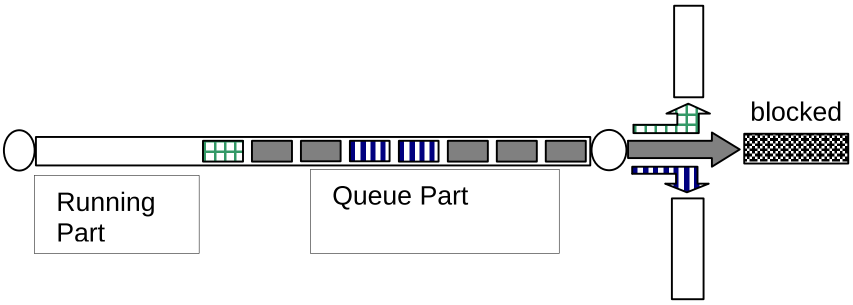


Figure 4. Queue servers at turning movements serving vehicles.

However, the fact that the queue of one turning movement can block access to the other turning movements needs to be represented. For instance it may be unlikely that the last vehicle in the queue (chequered) could pass the whole queue of vehicles heading straight, and make the left turn. On the other hand, it may or may not be possible for the vehicles heading for the right turn (striped) to pass, depending on the exact configuration of the approach (e.g. number of lanes and length of turning pockets). In many cases, there is a turn pocket that contains vehicles for a specific turning movement (see figure 5).

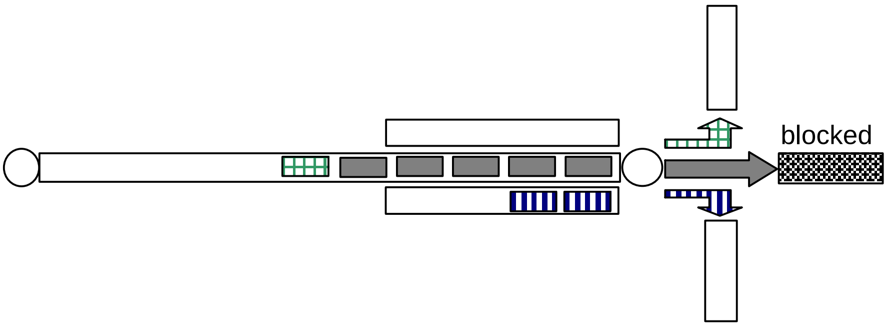


Figure 5. Turning pockets and blocking of turning movements by other turning movements.

On the other hand, this is a mesoscopic model and therefore a trade-off between precision and simplicity in representation of traffic. Since lanes and the individual behaviour of vehicles are not represented, it cannot represent the turning pockets and their operation directly. But it is possible to define for each turning movement a *queue look-back limit*, which is the maximum number of vehicles from the front of the (1-dimensional) queue that a server can "look back" to find a vehicle that is heading for its specific turning movement. So in the example in figure 5, if the look back limit for the left turn is 6 vehicles, then the left turning vehicle in the back of the queue (green) cannot exit until it has advanced to at least position 6 in the queue.

3.4 Turning Servers

The servers processing the vehicles through the turning movements are modelled using a stochastic process. This process generates the time intervals (time headways) between the processing of consecutive vehicles (in that direction). These processes represent the mean and variation of the throughput capacity of that turn, and are related to the *saturation flow rate and capacity* for that movement (see (May, A.D. 1990)). The saturation flow is the number of vehicles per hour per lane that can go through a specific turning movement in an intersection, if it would have green continuously. This depends on the number and width of the lanes approaching the intersection, the direction (left and right turning vehicles go slower than those going straight), traffic composition and many other factors. The capacity of that movement is the saturation flow rate times the percentage of time that turning has a green light. The inverse of the capacity of a turning movement gives the mean time interval between two vehicles going through that movement.

There is one server for each lane in the outgoing link of a turning movement. Although lanes in a link are not represented separately, the representation of the capacity is more accurate this way. The time headways that can be observed on a two-lane road can be as low as 0 seconds, since two vehicles on different lanes can pass in virtually the same instant. When we look at the time headways on each lane by itself, the smallest value that is observed is usually bigger than 0.5 seconds, and it is physically impossible for two vehicles to pass in the same instant. Since the passing of vehicles on two different lanes can be regarded as independent, they should be represented by independent processes in the model.

It is not completely clear which stochastic process should be taken to represent the capacity of turning movements. For the moment the capacity is assumed to vary according to a (truncated) normal distribution, which is a common way to represent the time headways of traffic at high flows (May, A.D. 1990), but different server types can be easily introduced.

A possible extension of the node server concept would be to introduce interdependence between turning movements. In the case of signal control with fixed signal plans the flow is now determined by the proportion green time in each direction. This could be extended by implementing the actual signal plans and introduce events for turning movements getting green and red (ignoring yellow). For adaptive control the green time can be adapted to the length of queues on each link. Actuation of detectors by vehicles cannot be simulated directly, because the exact positions of vehicles on the link are not known.

For opposing flows, such as a left turn that crosses the opposing straight-going flow, the outflow of the turning movement that has to give way can be calculated from the saturation flow – actual flow on the opposing movement. This procedure can be refined in the future to include probabilistic gap-acceptance in the opposing flows.

3.5 Speed Density functions

The traversal speed of vehicles on the links is determined by the density on the running part. This speed/density function can take different forms. In the Mezzo model the type of speed/density function used can be specified by the user. For an overview of research into speed/density functions can be found in (May, A.D. 1990) and more recently in (DelCastillo, J.M. & Benitez, F.G. 1995). The oldest known formulation of a speed /density relationship is by Greenshields (Greenshields, B.D. 1935):

$$V(k) = V_{free}\left(1 - \frac{k}{k_{jam}}\right) \tag{1}$$

Where

V = speed (km/h)

V_{free} = Free flow speed (km/h)

k = Density

k_{jam} = Jam Density, the maximum possible value for k.

Since V_{free} and k_{jam} are constants, this leads to a linear relationship between speed and density. Other important speed-density relationships are the one by (Greenberg, H. 1959):

$$V(k) = V_c \ln\left(\frac{k_{jam}}{k}\right) \tag{2}$$

Where

V_c = Speed at maximum flow (capacity)

(Underwood, R.T. 1961)

$$V(k) = V_{free} \exp\left(-\frac{k}{k_c}\right) \tag{3}$$

Where

k_c = Density at maximum flow (capacity).

and (Drake, J.S. et al. 1967):

$$V(k) = V_{free} \exp\left(-\frac{1}{2}\left(\frac{k}{k_{jam}}\right)^2\right) \tag{4}$$

Equations (1) – (4) are all examples of analytical expressions that were formulated to fit empirical data on speed and density. Later interpretations were sought to explain the models' parameters and functional form (see for instance (Gerlough, D.H. & Huber, M.J. 1975) for a behavioural justification of the Greenshields model). Some of them have rather interesting behaviour at the boundaries of the domain of k. Greenberg's model (2) for instance, will yield infinite speeds when k→0.

Another family of speed-density curves have been derived from (microscopic) car-following theories (Gazis, D.C. et al. 1959), (May, A.D. 1990):

$$V(k)=V_{free}\left(1-\left(\frac{k}{k_{jam}}\right)^{a}\right)^{b} \tag{5}$$

With
a, b = model parameter

This last functional form is very general and if a=b=1 it is equivalent to Greenshields model (1). Many car-following models can be shown to produce a speed-density function, under steady state conditions, that is of the form of (5). One example is the non-linear General Motors car-following model (Gazis, D.C. et al. 1961), which has been used in many microscopic simulation models. Furthermore, since the Mezzo model is to be combined with a microscopic simulation model, this functional form seems to be a good starting point to obtain a speed-density function that is compatible. Even though nowadays most microscopic simulators do not have car-following models that can be integrated to directly obtain the steady state speed-density function. (For instance the multi-regime models such as the Wiedemann model of VISSIM (PTV 2003), or the Gipps model of Aimsun (Barcelo, J. et al. 1997). Mitsim uses a multi-regime model based on an extension of the GM model, with different parameter sets for acceleration and deceleration (Ben-Akiva, M. et al. 1997))

However when considering which formulation suits best our purposes, all mentioned formulations have two main problems from a behavioural / phenomenological point of view. Firstly, the speed of traffic loaded at densities approaching jam density will be approaching 0 (km/h), which would mean that link travel times would approach infinity. Secondly, at low densities, the speed has been shown empirically not to depend on density, but to remain around V_{free}. (DelCastillo, J.M. & Benitez, F.G. 1995). These anomalies can be solved by introducing a minimum speed V_{min} and a minimum density k_{min}, and maximum density k_{max} replacing k_{jam}. This leads to the following general formulation:

$$V(k)=\begin{cases} V_{free} & ,if\ k<k_{\min} \\ V_{\min}+\left(V_{free}-V_{\min}\right)\left(1-\left(\dfrac{k-k_{\min}}{k_{\max}-k_{\min}}\right)^{a}\right)^{b} & ,if\ k\in[k_{\min},k_{\max}] \\ V_{\min} & ,if\ k>k_{\max} \end{cases} \tag{6}$$

Where

V_{min}	= minimum speed
k_{min}	= minimum density where speed is still a function of density
k_{max}	= maximum density where speed is still a function of density

This formulation is the same as the one used in DYNAMIT (Ben-Akiva, M. 1996), and in a simplified form in DYNASMART (Jayakrishnan, R. et al. 1994), where a=1 and K_{min}=0.

It has the following properties:

1. At k=0, V=V_{free}
2. At k=k_{max}, V=V_{min}
3. Speed decreases monotonically with density in the range k_{min} to k_{max}
4. Speed is independent of density if k <= k_{min}: V=V_{free}
5. Speed is independent of density if k >= k_{max}: V=V_{min}

To our purpose, this formulation has a number of advantages. Firstly the jam density in (5) means that traffic on the link has the maximum possible density, and is stopped. This corresponds to the functioning of the Mezzo queue segment, and it should therefore be kept out of the behaviour on the running part. Secondly the minimum speed is needed to ensure proper queue dissipation. The vehicles in the queue segment have an earliest exit time calculated from their speed. At maximum density this means V_{min}. In order to correctly represent the dissipation of a queue (for instance when a traffic light goes to green), this speed needs to be equal to the queue dissipation speed.

Thirdly, when calibrating the parameters of the function, k_{max} can be the maximum observed density, and k_{min} the minimum density where an influence on speed can be observed.

3.6 Dealing with inhomogeneous traffic conditions.

When simulating the travel times of traffic on a link using a speed/density function, or any other function that relates travel time to the conditions on the link, the assumption is that these conditions are homogeneous for the whole link. In this case one would assume that the density is homogeneous for the whole link. Since any limitation at the exit (such as traffic lights) invalidates this assumption (the density in the queue at the traffic light is much higher than the density in the remaining part) it was necessary to divide the link into a queuing part and a running part, and state that the speed/density is calculated over the running part only. While this solution solves the problem when a queue is building up, it does not solve the problem in the case of queue dissipation (for instance when the traffic light goes to green).

3.6.1 Shockwaves in traffic

In order to discuss the properties of queue propagation and dissipation, the theory of shockwaves (Lighthill, M.H. & Whitham, G.B. 1955), (May, A.D. 1990) is introduced briefly. Shockwaves in traffic are defined as discontinuities in the speed/flow/density in the space-time domain. In other words, they are the boundaries between sudden changes in density (and thus flow and speed) over space. These boundaries may be stationary, as in the case of a fixed bottleneck, where traffic downstream and upstream of the bottleneck have different densities (and thus speeds and flows). The boundaries can also move over time, as in the case of a queue building up at the approach of a signalised intersection. (May, A.D. 1990) identifies six types of shockwave:

- Frontal stationary: a stationary boundary between a higher and a lower density, where the density upstream of the boundary is higher than downstream. This is the case at a red traffic light, where the road downstream is (almost) empty, and the queue is forming upstream of the stop line.
- Backward forming: an upstream moving shockwave where the density is higher downstream than upstream of the boundary. The back of a queue is a typical example of this type of shockwave.
- Forward recovery: a downstream moving shockwave, where the density is higher downstream than upstream of the boundary. When there is a stationary queue (for instance at a lane drop) and the demand drops below the bottleneck capacity, the back of the queue will move forward until it dissolves at the bottleneck. This is an example of a forward recovery shockwave.

- Rear stationary: A stationary shockwave where the density is higher downstream than upstream of the boundary. If in the example of a bottleneck (above) the demand equals the bottleneck equals capacity, there will be a rear stationary shockwave at the back of the queue.
- Backward recovery: An upstream moving shockwave where the density is higher upstream than downstream of the boundary. In the case of the example with the red traffic light, when the light turns green, the vehicles at the queue front will one by one start driving. This means that the boundary between queued vehicles (at high density) and driving vehicles (at a lower density) moves upstream.
- Forward forming: A downstream moving shockwave where the density is higher upstream than downstream of the boundary. An example of this type of shockwave is when a slow moving truck causes a queue to form behind it.

In the Mezzo mesoscopic model, most of these shockwaves are represented correctly. The frontal stationary shockwave is normally represented by the limited capacity of the turning movements at the nodes. The backward forming and frontal recovery shockwaves are represented by the server process at the node, in combination with the horizontal representation of the queue. In other words, the vehicles in the queue occupy space, and therefore, with a growing (shrinking) queue, the queue back moves backwards (forwards) with the lengths of the vehicles joining (leaving) the queue back (front). The rear stationary shockwave is also represented correctly as it is the boundary case between a backward forming and a frontal recovery shockwave, for the reasons explained above.

The backward recovery shockwave, however, is not taken care of by the mechanisms presented thus far. In the case of a queue of stationary vehicles in front of a traffic light dissipating when the light turns green, there is no mechanism providing the backward moving boundary between high density queued vehicles, and the vehicles that have started to drive.

3.6.2 The speed of shockwaves

The speed w_{AB} of a shockwave that divides the traffic conditions A and B is known from traffic flow literature (May, A.D. 1990) and is given by:

$$\omega_{AB} = (q_A - q_B)/(k_A - k_B). \qquad (7)$$

So to find the speed of the shockwave, we can look at the fundamental flow/density diagram, locate (q_A, k_A) and (q_B, k_B) and find the speed as the slope of the line between these two points (see Figure 6).

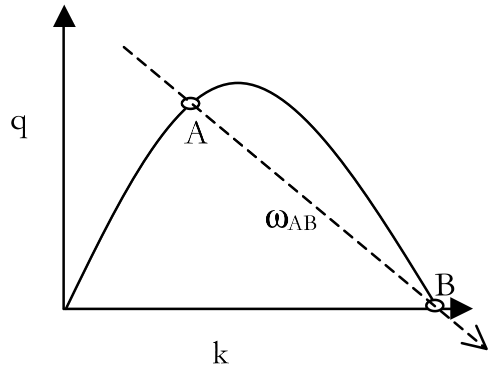


Figure 6. Shockwave speed from fundamental diagram.

3.6.3 Queue dissipation: The problem

Consider the following network with four nodes and three links.

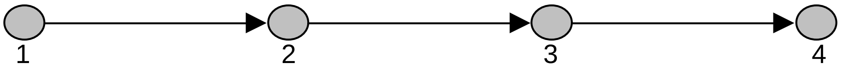


Figure 7. Simple network

When there is a queue building up from node 3 (for instance because of a red light blocking the exit of the link), the situation will look as follows after some time (t=1):

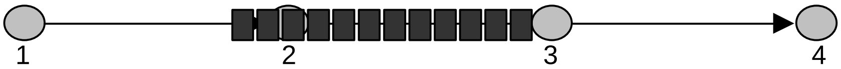


Figure 8. Queue at t=1

In terms of shockwaves, the queue build up at the back is a backward forming shockwave. In reality when the traffic light turns green, and the queue starts to dissipate at t=1, a backward recovery shockwave is created and the queue front will move upstream, while the queue will continue to grow at the back. Usually the growth at the back is a little slower, so that after a while the queue will dissolve. In other words the backward recovery shockwave will move faster than the backward forming shockwave, and catch up with it. When the queue dissolves, a new forward recovery shockwave is created since the density upstream of the queue back is lower than the density downstream of the queue front.

Normally at t=2 (sometime after the backward recovery shockwave has come into existence and before it reaches the backward forming shockwave) the situation would be:

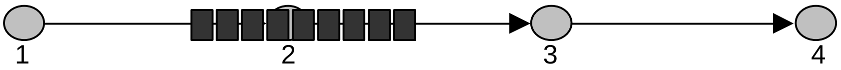


Figure 9. Queue in reality at t=2

However, due to the structure of Mezzo (and especially since the individual vehicles' position is not known, and neither can vehicles be hindered by specific other vehicles), at t=2 the situation would be:

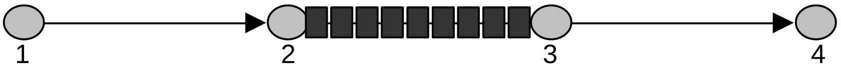


Figure 10. Queue in Mezzo at t=2

So in short, in Mezzo the queue front would remain at the downstream node of the link where the queue started. In other words, there is a frontal stationary shockwave, where there should have been a backward recovery shockwave.

3.6.4 The Solution: shockwave calculation

In Mezzo, when a vehicle is in the queue part (depending on the earliest exit time) it can in principle exit the link immediately. In the case of queue dissipation this is not correct. At any time, a vehicle in the queue has to wait for the backward recovery shockwave to reach it, and then calculate the time to drive to the stop line. Mezzo keeps track of each time a vehicle wants to exit the link, but it can't because the downstream link is blocked. When the link becomes unblocked and vehicles start moving, a backward recovery shockwave comes into existence. After its creation, it travels with a certain speed, which is a function of the difference in flows and densities before and after the front of the queue. The proposed method is the following:

1. When a link exit has been blocked and becomes unblocked, the speed of the backward recovery shockwave is calculated. Now, for each vehicle the time is known when the shockwave will reach it, and it will start to accelerate.
2. When the vehicle starts to accelerate, the time to drive to the end of the link is calculated using the speed corresponding to the density downstream of the shockwave.
3. The sum of these two times becomes the new earliest exit time for the vehicle.
4. This is done until a vehicle is reached that does not need to update its earliest exit time (it has an earliest exit time that is higher than the *updated* earliest exit time of the vehicle in front), or when the last vehicle on the link is reached.
5. When the link is completely full, the time when the shockwave reaches the upstream node is calculated, and the entry from the upstream node is blocked until that time.

This scheme will update the earliest exit times for the vehicles, and more correctly model the effect of queue dissipation in the link. Obviously, all vehicles that have been updated in this way have now become part of the 'running segment', and new vehicles entering the link from upstream are impeded by it (it is part of the density). This scheme is implemented in Mezzo and reproduces the backward propagation of a queue properly, as is shown in figure 14. In this figure the cumulative flow over time is plotted for eight consecutive links. Since the cumulative flow for a link means the number of vehicles that have entered the link (the link in-flows are presented here), the slope of the lines are the flow rate and the distance between the lines (each line represents the cumulative flow on one link) represents the density on the link. In this scenario a blocking incident is created directly downstream of link 8 at time t=1200 and lasts until time t=1450.

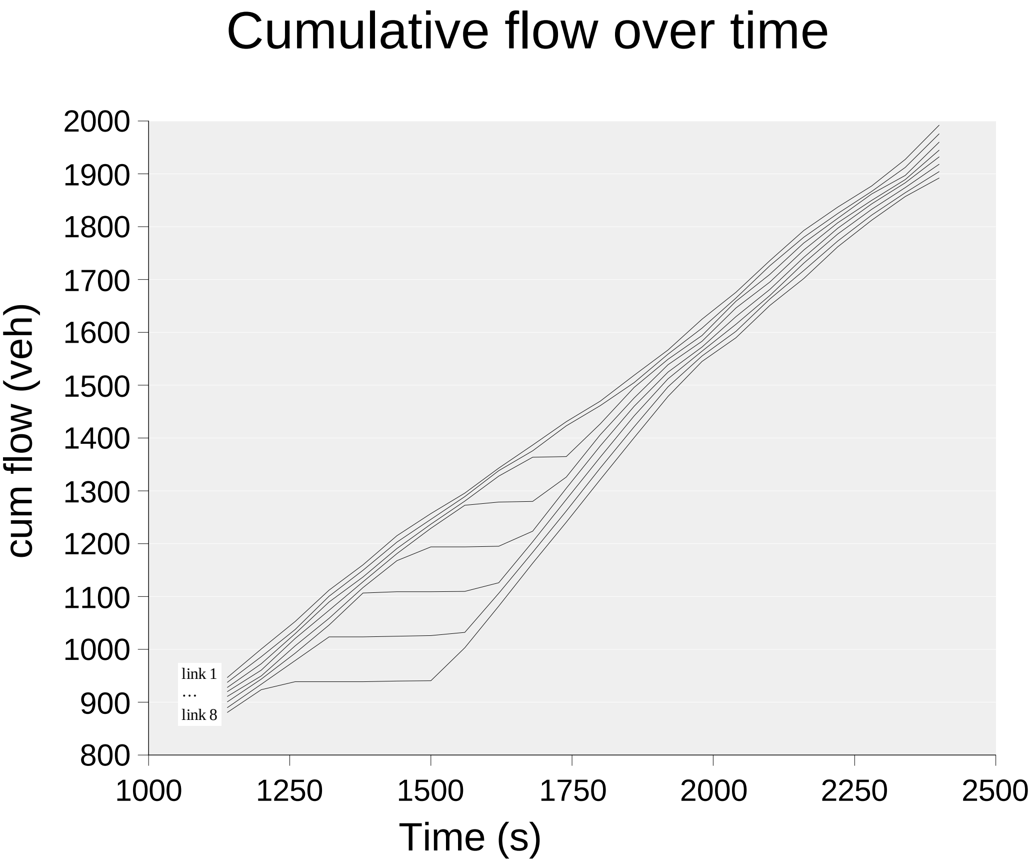


Figure 14. Cumulative flow on eight consecutive links. During and after a blocking incident at the exit of link 8.

The incident shows up in the figure 14 as the part of the line where the slope=0, and thus no vehicles are flowing into the link. During the queue build up phase it can be seen that at t=1250s link 8 has filled up and that link 7 is full at around t=1310s, link 6 at t=1370s, link 5 at t=1450s etc. So the backward propagation of the queue *back* (backward forming shockwave) is represented correctly. Given the link length of 300 meters, this amounts to a backward forming shockwave speed of around 4-5 m/s. (the speed of this shockwave depends on the inflow rate and the length of the vehicles).

The queue dissipation starts at t=1450 on link 8 and propagates via link 7, etc. to link 3 where it arrives at t=1700s. On link 3 the shockwave from the queue front has caught up with the shockwave of the queue back and the queue dissolves, which is why links 1 and 2 are unaffected by the incident.

3.6.5 Further extensions to deal with inhomogeneity

As reported by (Daganzo, C. 1994),(Daganzo, C. 1995) and (Carey, M. & Ge, Y.E. 2003), the problem with representing inhomogeneous traffic conditions on a link can be solved partly by introducing a discretisation of the link into segments (or cells), and assume that at least for these segments the traffic conditions are homogeneous. In the limit when the segment size goes to 0 this would be trivially true.

Carey & Ge show that two different flow models, when discretised, approximate the continuous solution of the LWR model. It is shown that the size of the segments has a clear effect on the performance of the models, and that smaller segment sizes result in a solution that is closer to the LWR model.

However, in the case of Daganzo and Carey & Ge, the models use a continuous representation of flow (and density), which allows arbitrarily small segment sizes. In the case of Mezzo the vehicles themselves are represented (with their individual lengths) and therefore the smallest segment size is limited. Smaller segments mean less space for the vehicles, and therefore, with increasingly smaller segments our speed-density function becomes a step function with increasingly large steps. (When the segment size is smaller than or equal to the vehicle length, the Mezzo model has turned into a cellular automaton, where a segment is either occupied or free, with no states in between).

Further research into the effects of different segment lengths on the performance of Mezzo should give a better insight into what segment length would be optimal.

3.7 Traffic generation

The demand is represented by a time-sliced Origin/Destination matrix, which specifies for each time slice the trips for all vehicle classes. The mix of different vehicle types (trucks, normal autos, buses etc.) is specified separately. This mix specifies each vehicle type, its length, the percentage of the total vehicle fleet and the capability of receiving information en-route.

The generation of traffic is done by a separate process for each Origin/Destination pair (OD-pair), and the inter-arrival times follow a negative exponential distribution:

$$P(h \geq t) = e^{-lt} \qquad (8)$$

Where
h = time headway
t = time headway parameter
$l = 1/\bar{t}$ = distribution parameter (mean and standard deviation)
$\bar{t}$ = mean time headway

So for each origin a number of OD-pairs generate the arrivals of vehicles independently, and enter the vehicles directly on the outgoing links. When a new vehicle is generated, the origin and destination are already determined. Then first its type is determined by a random draw according to the vehicle mix that is specified and then a route is chosen from the set of known routes from the origin to the destination.

The changes of demand over time for different OD-pairs are treated as events. This means that in the demand input file, the time-slices for the demand can vary in size, and can contain only those OD-pairs for which the demand changes. This allows for a flexible representation of the demand.

3.8 Route choice

The route choice in Mezzo exists of two main mechanisms. The pre-trip route choice, which selects the drivers' 'habitual' path (the terms route and path are used interchangeably), and a route-switching mechanism, that allows drivers to divert en-route from their current path. The next section will discuss the generation of historical travel times and habitual paths, which are presupposed by the pre-trip route-choice.

3.8.1 Pre-trip route choice

The route choice in the Mezzo model is based on a set of known routes and historical link travel times for all links.. These travel times are time dependent (meaning that the travel time on a link at say 8:00 differs from the travel time at 8:30, etc.). When a vehicle is created at an origin, it makes a pre-trip route choice according to a multinomial Logit function (MNL) (Ben-Akiva, M.E. & Lerman, S.R. 1985) from the set of known routes that connect its origin to its destination according to a certain probability.

The following function shows the MNL probability that a driver will choose alternative i, given the set of alternatives S and the utility function U$_i$.

$$P_i(t) = \frac{e^{U_i(t)}}{\sum_{j \in S} e^{U_j(t)}} \tag{9}$$

Where

P$_i$(t) = The probability of alternative i being chosen, given departure time t.

U$_i$(t) = The utility of alternative i, given departure time t.

S = The set of all eligible routes between the given origin and destination.

At the moment the utility function that is used in the route choice is only a function of the time-dependent travel time on the route. In the future more complicated utility functions can be implemented, taking into account factors such as the number of stops, the distance, the average speed on a route, the number of intersections, etc. Also, the set S of eligible routes can be made time-dependent (S(t)).

Another issue that should be implemented in the future is a correction for the overlap between alternative paths, such as C-Logit (Cascetta, E. et al. 1996) or Path Size Logit (Ramming, M.S. 2002). When using simple multinomial Logit, it is important that the utilities of alternatives in the choice set are independent (IIA property), which is obviously not the case if alternative routes may consist of (for instance) 90% the same links. C-Logit or Path Size Logit correct for this violation of the IIA property.

3.8.2 En-route switching

During the generation process of the historical travel times and routes database (see next section), the en-route route-switching mechanism is inactive. When the historical travel times and routes database has been generated, scenario's can be tested that include switching of routes en-route. Cases where this would happen are when information about an incident (see description of incidents further on) is received, a road blockage, or when the driver experiences excessive delay (compared to the expected travel time).

The route switching mechanism that is implemented models currently the reaction of drivers to broadcast information about an incident. The information consists of the link on which the incident happened, the time it started, the expected duration, and the expected delay on that link. This delay includes expected delays on upstream links.

When such information is broadcast, a certain percentage of the drivers receives the information and determines whether it is relevant to them. Currently this means simply checking if the information concerns a link that is part of the driver's current route. If this is the case, the driver compares the expected travel time on the remainder of his current route (with the expected delay added) to the shortest time alternative route from his current link to the destination (again taking into account the expected delay on the incident link). If the 'alternative' route found differs from the current route, the driver makes a choice using the Logit model described in (9) between the current route and the alternative, if the alternative is estimated to be significantly better than the current route (the difference should be larger than a pre-specified threshold).

3.8.3 Historical travel times and Path set generation

For the pre-trip route choice to function, a set of candidate paths needs to be present for each OD pair, as well as correct time-dependent historical travel times for the links. The generation of a path set that includes all realistic paths, but no other (spurious) paths is not trivial. Nevertheless, this area of research has not received much attention, compared to the area of route-choice. A recent contribution to this area, and a good overview of existing techniques can be found in (Ramming, M.S. 2002).

The most trivial way of generating a path set is by including only the shortest (time/distance) path and assigning all OD traffic to that path. This method, also known as all-or-nothing assignment, eliminates the need for pre-trip route choice. (for a review of shortest path finding methods, see (Cherkassky, B.V. et al. 1993))

However this method has a number of obvious drawbacks. First of all, in practice travellers tend to take many other paths, and (Ramming, M.S. 2002) reports that only a small portion of travellers actually travel along the shortest (travel time) path. Moreover, for OD pairs with high demand, often one single path cannot accommodate all travellers, as the demand is higher than its capacity.

Another method that has been commonly applied is the K-Shortest path algorithm. Usually variants of this method will add a penalty to all links in the shortest path and search for a new shortest path. Then a penalty is added to the links in the new shortest path, and yet another shortest path search is performed, until k alternative paths are found. Other variants eliminate links from the shortest path instead of adding a penalty, or use different link attributes such as distance, number of stops and travel time to create different paths. However a common problem is that the k different paths often differ by very little, and in practice often do not cover paths that are known to be used (Ramming, M.S. 2002). Other methods for path set generation often use elimination to exclude unrealistic paths. However this presupposes the creation of a set of all paths beforehand, which can be very time consuming.

In addition, (Ramming, M.S. 2002) introduces an approach where a random 'perception noise' is added to the link costs, after which the shortest 'perceived' path is generated. This process is repeated with different draws of the random noise. This last approach is shown to have a good overlap with routes chosen by real travellers in a test area in Boston.

In (Mahut, M. et al. 2003) an approach is presented, where the path set and travel times are generated using iteration. In the initial N iterations, for each OD pair and time period new shortest paths are calculated, based on the travel times from the previous iteration. After N iterations no new paths are added to the choice set. Instead of an explicit route choice modelling, the vehicles are distributed equally over the known paths for the first N iterations. In the following iterations the shortest (minimal travel time) path(s) (among the set of known paths) from the previous iteration will be assigned increasingly larger proportions of the demand. This algorithm stops when the total travel time (sum of travel times of all vehicles) approaches the 'optimal' total travel time (the sum of vehicle travel times if they would have taken the shortest paths).

The approach proposed is the use of an iterative dynamic traffic assignment procedure to generate both the path set and the historical link travel times. With regard to path set generation this approach is similar to that of (Mahut, M. et al. 2003), but there are differences as well. In our case the route choice is modelled explicitly and the travel times for each new iteration are averaged from the input and output travel times of the previous iteration. In addition, the stopping criterion is different, and the finding of new paths is not restricted to the first N iterations. In figure 15 the overall scheme of generation of the base route and travel time database is depicted.

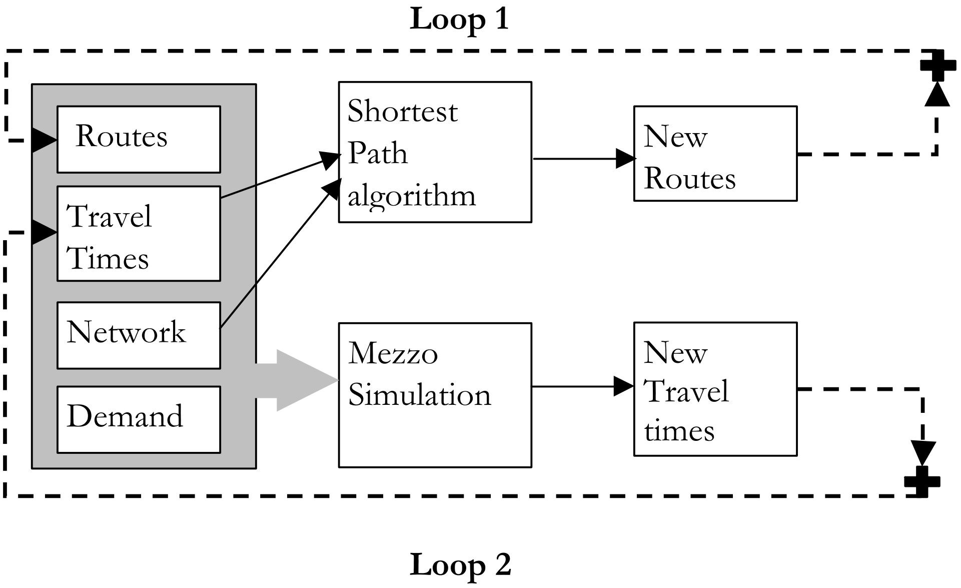


Figure 15. Generation of historical travel times and routes database.

Starting with the first iterative loop, loop 1, it can be observed that the shortest path algorithm starts with the network description (the links and nodes that make up the road network) and the time-dependent travel times for all links in the network. When the travel times are unknown (for instance in the first iteration of the generation process), surrogate travel times are calculated from the free-flow speeds on all the links. The shortest path algorithm now finds the shortest paths for all origins and destinations, for each departure time period. Then all paths that are not yet in the route database, are added.

Then loop 2 becomes active. The whole grey box is input to the Mezzo simulation model, including the newly updated route set. In Mezzo drivers will choose their (pre-trip) routes according to the "historical" travel times (sometimes called "experienced" or "expected" travel times). Again, if these times are not known, the travel times are estimated from the free-flow speeds on the links. One of the simulation outputs from Mezzo are the new travel times that were experienced by the drivers, as a result of their route choice and interaction with other drivers (the busier the road, the higher the travel time). These new travel times are averaged with the "historical" travel times that were input to the simulation:

$$tt_i^{n+1}(t) = a\, TT_i^{n}(t) + (1-a)\, tt_i^{n+1}(t) \tag{10}$$

Where,

$tt_i^{n+1}(t)$ = historical travel time on link *i* for iteration *n+1*, when entering at time *t*

$tt_i^n(t)$ = historical travel time on link *i* for iteration *n*, when entering at time *t*

$TT_i^n(t)$ = Simulated (output) travel time on link *i* for iteration *n*, when entering at time *t*

a = moving average parameter, $\in [0,1]$

This method (also known as single exponential smoothing) ensures stability in the iteration process, which may exhibit oscillatory behaviour if the new travel times simply replace the old ones (in formula (10) if a = 1, this is the case, and if a = 0 the new travel times are discarded)[3]

The updated travel times are then used in the next iteration of Loop 1. This time, the shortest path algorithm may find new shortest paths, since the travel times may have changed from the previous iteration. These new paths are added to the route set, and a new iteration of Loop 2 is performed. This way Loop 1 and Loop 2 are iterated alternately until Loop 1 does not find any new routes and Loop 2 produces travel times that are sufficiently close to the historical travel times it used as input.

This heuristic process of dynamic traffic assignment does not guarantee convergence to equilibrium, but has in practice shown to converge to a local equilibrium. The speed of convergence is also a function of the complexity of the network, the definition of “sufficiently close” and the level of saturation of the network. Another issue is the fact that the model used is stochastic, and therefore multiple simulation runs should be made to gain a good estimate of the network performance (See (Burghout, W. 2004),(Toledo, T. et al. 2003) for a good way to calculate the number of simulation runs needed, based on the variance of the MOEs). Loop 2 can be modified so that the average of link travel times of a number of runs is used in (10).

[3] More sophisticated methods are possible, for instance to correct for trends in the successive forecasts. Examples are Method of Successive Averages, where $a_i = 1/i$, or Adaptive Response Rate Exponential Smoothing, where a_i is varied according to the forecast errors in each period.

The path set generation procedure has been shown to produce realistic paths, where OD pairs that are connected by paths with little traffic will only have the shortest path as a choice, and OD pairs with paths through heavily trafficked parts of the network may have a larger number of paths in the set (in practice rarely more than 5).

The most important test for a path set generation algorithm is a validation where it is applied to a certain (large enough) traffic network, and the resulting paths are compared with a survey of the paths that people actually use in the study area, such as in Ramming (Ramming, M.S. 2002). However such a study is outside the scope of this thesis and we have to content ourselves with the observation that the paths generated by the combined dynamic assignment and path generation procedure are sufficiently different from each other and that OD pairs with routes through highly congested areas tend to get a larger number of routes.

3.9 Operational Issues

3.9.1 Event based simulation

In most traffic simulation models the state of the traffic is computed at fixed time intervals. This is called discrete-time simulation. From each time step to the next the changes in the traffic state is calculated. In the case of microscopic simulation models this means for instance that the vehicles make little “hops” according to the distance they have travelled in the time interval. Another way of simulating the changes in traffic state is computing the changes “every time something happens”. This is called event-based simulation. In the case of Mezzo these events are for instance the entrance of a vehicle into a link, the generation of a new vehicle, the transfer of a vehicle from one link to another via a turning movement. See figure 16 for a graphical explanation of event-based versus discrete-time simulation.

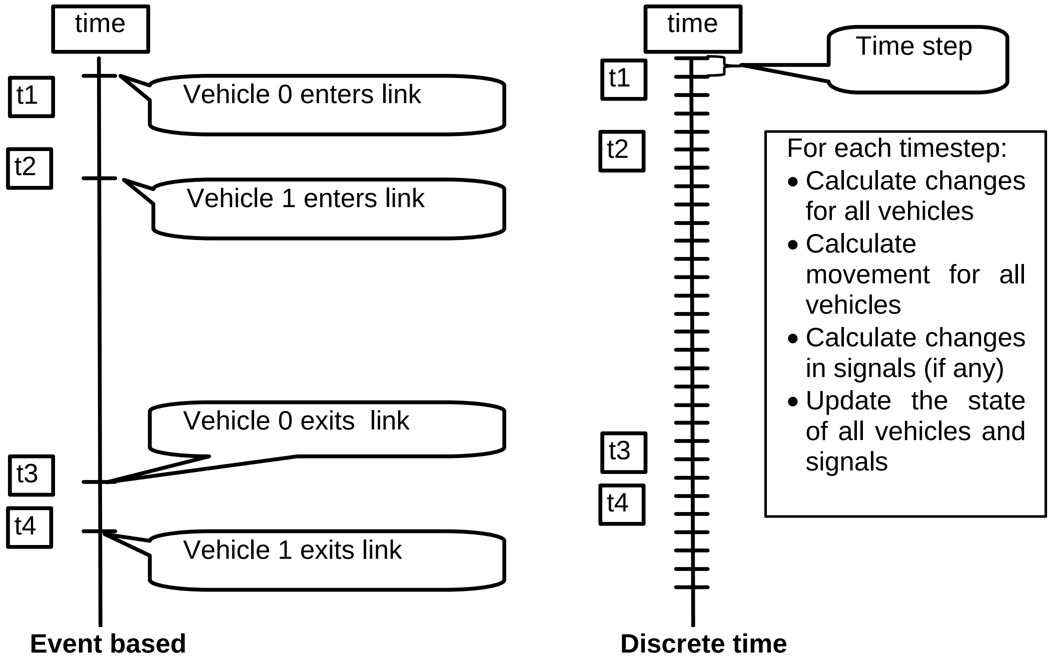

time
t1
Vehicle 0 enters link
t2
Vehicle 1 enters link
Vehicle 0 exits link
t3
t4
Vehicle 1 exits link
Event based
time
Time step
t1
t2
For each timestep:
• Calculate changes for all vehicles
• Calculate movement for all vehicles
• Calculate changes in signals (if any)
• Update the state of all vehicles and signals
t3
t4
Discrete time

Figure 16. Event based versus time based simulation

In the case of a discrete time simulation, a lot of computation time can be spent updating all individual objects (vehicles, traffic signals, detectors) each time step. Especially if many objects do not change state during many time steps (such as vehicles that are stopped in a queue), this can waste a lot of computation time. On the other hand, not all types of simulation are equally suited to event based processing. The reason why (to the author's knowledge) microscopic simulation models are usually discrete time, is that it is very difficult to predict for each vehicle when the next "event" is going to take place, since it depends very much on what the vehicles in the vicinity are 'doing'. In particular, in the case of many such cause-effect relations between objects, the risk exists that a deadlock occurs. An example of a deadlock is when object A is dependent on object B, which in turn is dependent on what object A does. As a result the simulation logic may be 'locked' forever. Another difficulty may be the fact that the causal order of events may be distorted, due to simulation simplifications and deadlock-correction mechanisms, resulting in events taking place 'in the wrong order'.

In addition, discrete-time simulation enables smoother graphical animation of the movement of vehicles during the simulation, which is an additional concern for microscopic simulation models.

For mesoscopic models like Mezzo, the event based approach is ideal, since the particular movements of the vehicles on the link are not traced, only the entries and exits. Therefore there are few causal interactions between objects, and virtually no risk for deadlock situations or causality violations.

The structure of event processing in Mezzo is depicted in figure 17. The events in the Mezzo simulation are ordered in an *event list*. Each process that generates events is called an Action. There are different types of actions, and four are depicted in figure 17. An ODAction is associated with an Origin Destination pair, and generates the events for vehicle generation. The Daction is associated with a Destination and generates the events that take away vehicles from the network. The TurnAction generates the events for processing the vehicles at turning movements. The MatrixAction marks a change in the demand for one or more OD pairs.

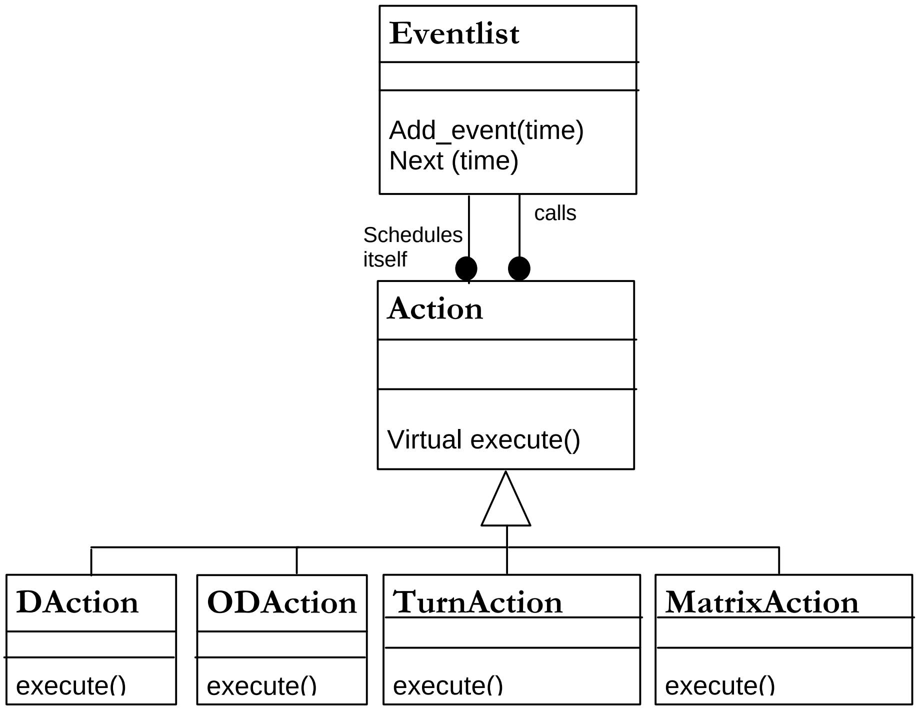


Figure 17. Event processing in Mezzo.

Each Action performs certain changes to the state of the system (adds a vehicle to a link, removes a vehicle from the network or transfers a vehicle from one link to another) and then 'books' itself into the event list with a new time.

The event list simply calls each Action as their event becomes the first in the list. The new time for the next event is generated stochastically, depending on the kind of action, by a 'server'. This server draws random time intervals given a certain mean and standard deviation, and the next time the Action is called is the current time plus the generated time interval. Depending on their function (for traffic generation, turning movements, or other) servers can draw time intervals according to user-specified distributions such as Normal, Negative Exponential, etc.

This structure allows for easy extension for new action types, such as Signal Controller actions for signal plan changes, actions for communications with other (micro) models (see Chapter 4), etc. In these cases the times for new events need not be determined by stochastic servers, but are determined by the internal logic of the controller or communication module.

3.9.2 Simulation inputs

The Mezzo simulation takes the following inputs (which are defined in a master file):

- A network description. This file contains a description of the
  - Speed/density functions: the parameters a and b, free speed and min speed, max and min density.
  - Servers: the type of distribution (normal, exponential, etc.), mean and standard deviation.
  - Nodes: the coordinates and type: Origin, Destination and Junction. For Destination also which server is used.
  - Links: the start and end node, number of lanes, length and which speed-density function is used.
  - Turning movements: Node, incoming and outgoing link, which server is used, the queue look back limit, and (optionally) added delay. The delay is useful when the movement through the intersection incurs a certain (average) delay.
- Routes: origin, destination, list of successive links
- Travel Times: For each time period the travel time per link
- Demand: Origin/Destination (OD) matrix for each time period (slice). Time periods do not have to be of equal duration. Every slice contains the time it becomes active and those OD-pairs for which the demand changes.
- Vehicle mix: For each vehicle type the length, percentage of total fleet, capability to receive information.
- Server rates. Optional file in which changes over time in the means and standard deviation of the servers are specified. This is used for instance to temporarily decrease the throughput of a node in case of an incident. It is also useful to decrease the exit capacity at a destination, if this is known to

change over time (to model the effects of congestion outside the boundaries of the studied network).

Turning movements can be generated automatically when none are present in the input files, and a default server is assigned to them. This server usually needs to be replaced, since different turning movements normally have different saturation flows.
If no routes are specified, they are generated by the shortest path module, given the input travel times. If no travel times are known, the free-flow travel times on the links are used as surrogate.

The network and demand definitions can be calculated from Emme/2 (INRO-Consultants 1996) or Contram 8 (Leonard, D.R. et al. 1989), (TRL 2004) networks and demand files. The links and nodes are mapped directly, and a speed-density function is assigned automatically to the links, depending on their speed limit. In the case of Contram the server type and rates for the turning movements can be calculated from percentage effective green time (in the case of signalised intersections) and saturation flows. In the case of Emme/2 networks they need to be specified manually (replacing the default servers).

3.9.3 Simulation outputs

- MOEs : For each link and each aggregation period (e.g. 60 seconds) all Measures Of Effectiveness, which are the following: Avg. Speed, Avg. Density, Avg. Inflow, Avg. Outflow and Avg. Queue length.
- Link times: The time-dependent link travel times. If α=1 these are the pure output travel times of the simulation. If 0< α < 1, these are the weighted average of the input travel times and the simulated travel times. If α=0 these are equal to the input travel times. (See explanation above)
- Vehicle trips: This file contains the path travel times for each OD pair, for each vehicle trip.

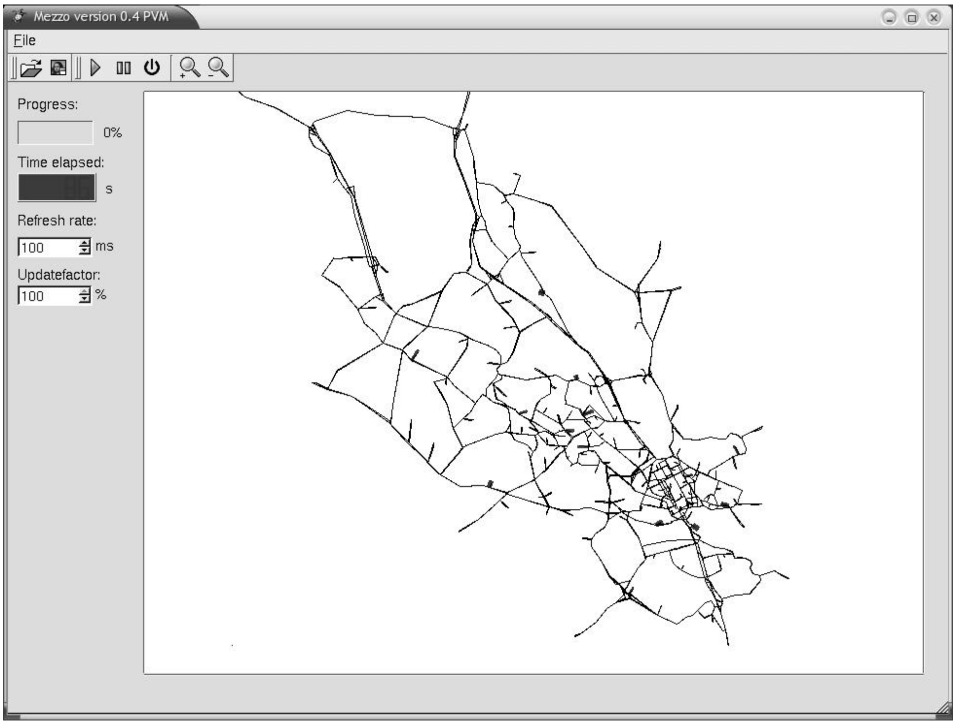
3.10 Implementation

3.10.1 General

The Mezzo model has been designed using the Unified Modelling Language (UML) (Booch, G. 1994), (Rumbaugh, J. et al. 1991) methodology, and is therefore strictly object oriented, both in design and implementation. The implementation is done in C++ with heavy use of the Standard Template Library (STL) for storing and retrieving objects. For the Graphical User Interface the QT cross-platform libraries are used (Trolltech), ensuring that the model can be compiled and run on MS Windows, Linux and MacOS platforms without modification. There are no limits to the size of the networks that can be simulated, nor to the amount of vehicles that can be simulated concurrently. While little effort has yet been made to optimise the speed of execution of the model, a network with 1000+ nodes and 2000+ links can be simulated three times faster than real time on a Pentium 1 GHz with 256 MB ram under Linux.

3.10.2 Graphical User Interface

The graphical user interface allows the user to observe the state of the network over time, during the simulation. Different values can be shown per link, but most commonly the queue length and density is shown graphically for each link, using bars that change in colour and size. An example is displayed in figure 18.



Progress: 0%

Time elapsed: s

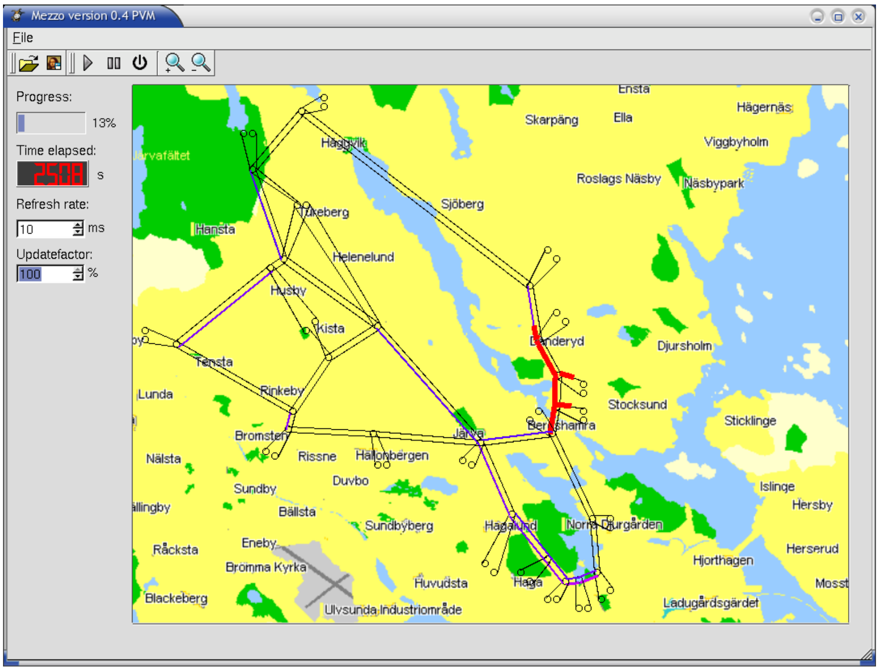
Refresh rate: 100 ms

Updatefactor: 100 %

Figure 18. Mezzo screenshot

In figure 18 a screenshot is shown of the Mezzo model running a network of 2000+ (2173) links and 1000+ (1181) nodes, containing the main roads of central and north Stockholm. Each node is represented by a small circle and each link is represented by a black line. If there is a queue of vehicles at the downstream node waiting to exit, this is shown by a red bar extending from the downstream node in the upstream direction. Its length indicates the length of the queue (or, more precisely, the amount of storage space occupied by the queued vehicles, as a percentage of the total available space on the link). In addition, the density on the link is shown for the running part (that part of the link that is not occupied by the queue) by a bar extending from the end of the queue bar to the upstream node. This density bar changes colour from blue to red, and gets wider, when the density increases. When the maximum density is reached, the colour is red, and the width the same as the queue bar.

In figure 19 a smaller network is shown, containing the major roads in the north of Stockholm. An incident on the north-west side is causing a queue build-up, shown by the red bars on the congested links.



Progress: 13%

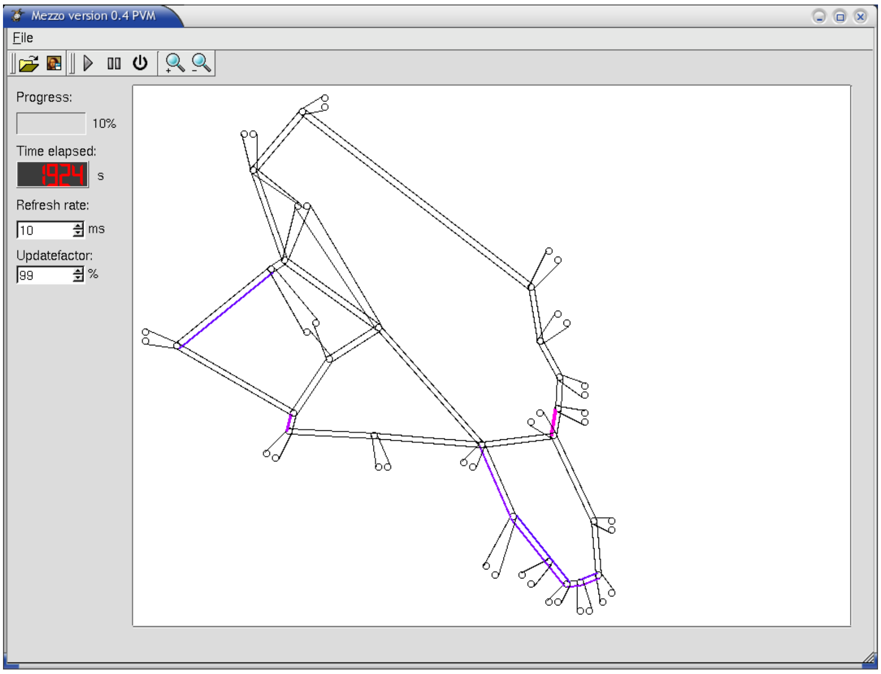
Time elapsed: 2508 s

Refresh rate: 10 ms

Updatefactor: 100 %

Figure 19. Queue build-up (red) in north Stockholm network.

In figure 20 the same network is shown, but without queues (and without background). This illustrates the display of density by colouring the links. A **black** link has little density, **blue** has moderate density and **pink** has a high density. **Red** links are completely congested (jam density). Instead of density other quantities may be shown, such as speed, inflow or outflow.



Progress: 10%

Time elapsed: 1924 s

Refresh rate: 10 ms

Updatefactor: 99 %

Figure 20. North Stockholm network without queues.

3.10.3 Compatibility

The input and output files are based on the XML format, which allows for easy data communication with databases (possibly over the internet) and other simulation models. An additional advantage of the XML format is the fact that the structure of the files is determined by a published Document Type Definition (DTD). This allows other programs (and humans) to read and interpret, as well as write the information correctly, and makes it possible to transform the input files to and from other formats (such as those needed in a GIS). Import filters are being developed that convert network coding and OD demand files from Contram 8 format and Emme/2 format (with augmentation of turning server specifications). A Java-based network editor is being developed for the editing and creation of Mezzo networks.

Chapter 4. MiMe: A hybrid Micro-Meso model

4.1 Introduction

In this chapter a new hybrid mesoscopic-microscopic simulation model is presented. An important aspect in developing a hybrid meso-micro traffic simulation model is the definition and implementation of conditions for consistent interfaces between the meso and micro simulation components of the simulation model. These conditions range from structural compatibility issues in terms of modelling traffic flows in the two models, to model synchronisation and compatibility of route choice.

In chapter 2 it was observed that most of the existing hybrid models involve microscopic and macroscopic models. Their approaches focus on the compatibility of the micro and macro models involved. Several approaches attempted to derive a micro model from the macro model. However these approaches had some limitations that were reported to be difficult to overcome. Most importantly, the micro (car-following) models that were derived from the macroscopic flow theory were difficult to extend to include important components such as lane-changing, gap acceptance and stochastic driver behaviour. It was also clear that some aspects of model integration have not yet been studied, such as the integration of network representation and route choice. On the other hand, a number of important issues were pointed out (and treated) in the literature, such as the proper propagation of traffic conditions across the inter-model boundaries.

Based on these considerations, the approach adopted here is to establish an integration framework of universal integration issues and conditions, and to develop a functional prototype hybrid model in which they can be evaluated and tested. Moreover, rather than developing a (simplified) microscopic model from a macroscopic or mesoscopic model, a new mesoscopic model was presented in chapter 3 that is capable of being integrated with a microscopic model. The micro-meso model equivalency that is sought is that of the traffic performance of the micro and meso models as a whole.

The first section discusses the requirements that an integrated hybrid model should meet in order to be consistent, and in the second section an integration architecture is presented that is capable of meeting the stated requirements. In the final section the implementation of the hybrid model is described.

4.2 Requirements for integration

In this section describes the different requirements that need to be met for the formulation of a consistent hybrid meso-micro model.

4.2.1 Consistency in route choice and network representation

One of the most basic conditions is general consistency of the two models in their representation of the road network, paths, and route choice. The route choice needs to be consistent to ensure that vehicles will make the same decision given the same choice situation, regardless if they are in the micro or the meso model. This means also that the representation of the road network needs to be made consistent throughout the hybrid model, as well as the travel times (link costs). As is discussed later on, this is not trivial to achieve.

4.2.2 Consistency of traffic dynamics at meso-micro boundaries

Besides the consistency of network representation, the consistency of traffic dynamics at the boundaries from the meso to the micro submodels and vice versa needs to be ensured. This means that traffic dynamics upstream and downstream of the boundaries need to be consistent. For instance, when a queue is forming downstream of the boundary point, and grows until it reaches and passes the boundary, it should continue in the other submodel, upstream of the boundary, in a similar way as it would if the boundary had not been there.

4.2.3 Consistency in traffic performance for meso and micro submodels

The two submodels need to be consistent with each other with regard to the results they produce. Ideally, given the same network and demand data, one would like the two submodels to produce the same results (such as travel times, link speed, flow and density). However, the reason for combining the meso and micro models is the fact that each model type has its own strengths and weaknesses, and thus that the models may produce different results given the same network. In addition, the micro model produces results (such as vehicle trajectories) that cannot be produced by the meso model, and thus not compared. Therefore the following condition on submodel similarity is defined:

For those facilities that can be simulated sufficiently well by both models, they need to produce (nearly) the same results on the meso level of aggregation.

Meso models produce aggregated results such as link travel times, flows, speeds and densities. Thus, the models should be compatible with respect to these measurements, at least on facilities that do not require the specific detail found in micro models. This implies the need for consistent calibration of the two submodels.

4.2.4 Transparent communication and data exchanges

The submodels exchange large amounts of data conveying vehicle characteristics, and downstream traffic conditions. This requires an efficient synchronization and communication paradigm, and a design that minimizes the amount and frequency of data exchange. Otherwise the communication overhead may become very large. On the other hand, aggregation and disaggregation of information at the boundaries may introduce complications, and should thus be avoided (see (Magne, L. et al. 2000), (Bourrel, E. & Lesort, J.-B. 2003), (Bourrel, E. 2003), (Helbing, D. et al. 2002)).

4.3 Integration framework

In this section an integration architecture is proposed that facilitates the consistency between the submodels and alternative modelling approaches are discussed to meet the requirements identified in Section 4.2.

4.3.1 Integration Architecture

Simulation models are in general a synthesis of a number of supply and demand models. Meso and micro models use different approaches to model traffic dynamics and different levels of aggregation for network representation. However, since they both use an individual vehicle-based representation of flow, they usually employ similar travel behaviour models (e.g. route choice models). Hence, the route choice component should be shared by both models, and operate on the complete network graph, including both the micro and meso areas. The paths should be defined over this common network graph, and the link travel time database should cover the entire network, and also be shared by both models.

The proposed architecture consists of a module, outside the meso and micro models, that contains the common elements: a database with the network graph, the travel time tables, the set of paths and the origin-destination (OD) flows, as well as a travel behaviour component with route choice models and path generation algorithm.

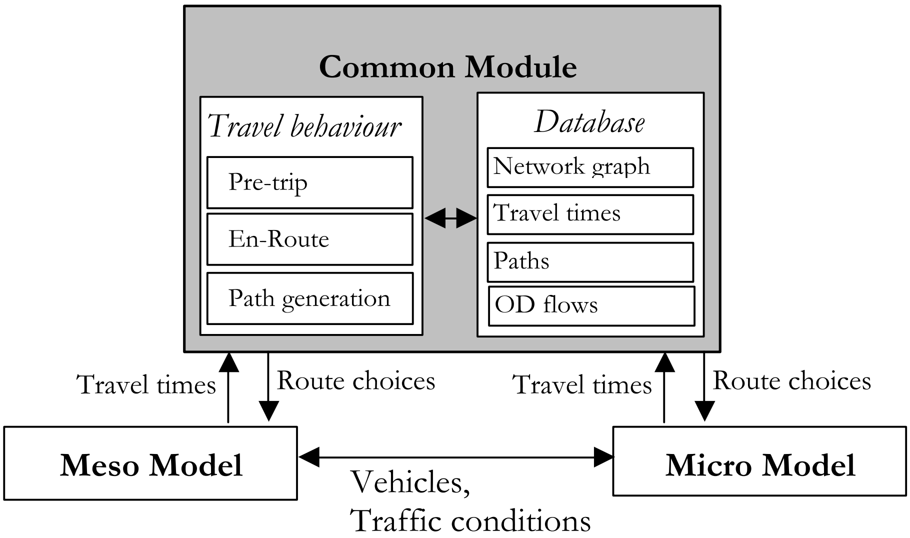


Figure 1. Integration Architecture

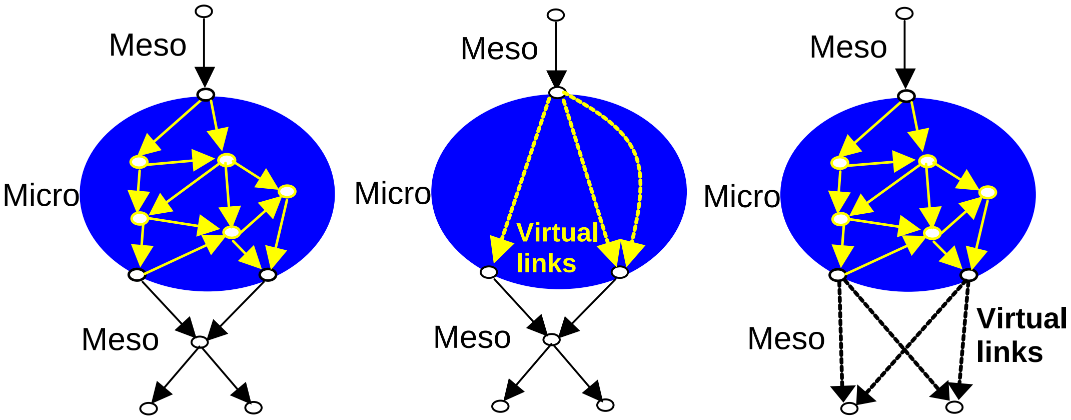
Figure 1 presents the proposed integration architecture. The travel behaviour component in the common module uses the database that contains the entire network graph, link travel times and known paths for the entire network. Both the micro and meso models supply descriptions of their subnetworks, from which the network graph is constructed. Each time a vehicle makes a route choice (whether in micro or meso), the common module is consulted. For the common route choice module to operate properly, the meso and micro models need to update the travel time database regularly with the link travel times in their subnetwork.

The above architecture is applicable in the case where the two simulation models to be integrated are new models, developed with integration in mind. In integrating existing models, the implementation of the integration framework needs to be adjusted to minimize inter-model communication, and use functionalities that are implemented separately in each model in a consistent way. For example, the simulation models may have their own route choice models. In this case, consistent route choice behaviour across submodels requires two conditions:

a) Route choice models with the same structure and parameters in the two submodels
b) Consistent path choice set (defined over the entire network).

To maintain the required consistency in this case, the micro and meso networks are enhanced with the addition of virtual links that correspond to the paths in the remaining network (see Figure 2).

More specifically, the meso network includes virtual links for each path connecting boundary nodes in the micro network (Figure 2b). This representation guarantees that each relevant path through the micro model is represented correctly in the meso route choice. The meso model collects travel times for the virtual links, and uses them in the route choice like any other link in the network.



(a) The entire network *(b) Meso virtual links* *(c) Micro virtual links*

Figure 2. Virtual Links in Meso and Micro

Since each relevant subpath in the micro area needs to be enumerated (as virtual link in the meso network), these paths need to be determined prior to the simulation. This can be done manually or automatically, for instance using the path generation algorithm outlined in section 3.8.3. When these subpaths are found, the micro pre-trip route choice will be disabled, since each vehicle that enters the micro area now already has a subpath that corresponds to the virtual link in meso.

Similarly the micro network is expanded with the addition of *micro virtual links* to the micro subnetwork (Figure 2c). The virtual links allow the micro model to deal with en-route choice in the micro subnetwork. Since each path from an exit point in the micro network to a destination in the meso network is represented by a virtual link in the micro model a change of route for a vehicle in the micro sub-network can effectively mean a different exit point into the meso network.

The common module also includes the OD matrix for the entire network. To simplify the information exchange and facilitate a transparent data input interface, the current implementation assumes that all origin and destination nodes in the network belong to the meso subnetwork. This assumption is by no means restrictive, since an origin or destination node in the micro area can always be designated as a boundary node in meso connected directly to the micro subnetwork.

The above considerations simplify the initial architecture considerably in the case where integration of existing simulation models takes place (Figure 3). Under this architecture, and based on the OD matrix representation discussed earlier, the meso model is solely responsible for all pre-trip decisions, while en-route decisions are the responsibility of the respective subnetwork. This architecture has a number of practical advantages compared to the initial one. While the initial architecture would be preferable in the case of a new meso-micro model, it would require large amounts of communication overhead when combining two existing models. With the above modifications this overhead is avoided.

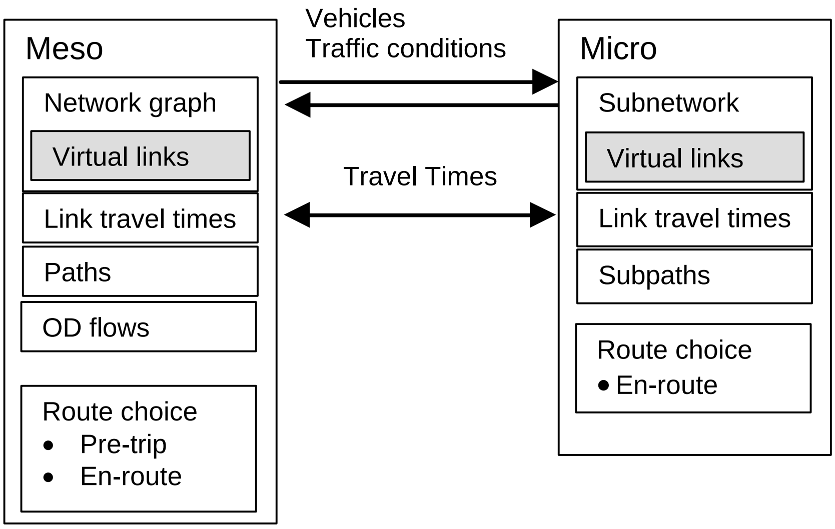


Figure 3. Simplified Integration Architecture

In the case of small micro areas, changes of subpath and exit node are often not possible. In this case the virtual links in micro and en-route choice module in micro can be eliminated, which further reduces the communication needs.

4.3.2 Modelling for Consistency

Assuming a common network graph, the architecture proposed in section 4.3.1 ensures route choice consistency. However, several aspects still need to be addressed at the modelling level, especially at the interface between the micro and macro areas.

Consistent network representation

An important step in the consistent route choice is the formation of a consistent network representation from the meso and micro subnetworks. Problems may arise from the level of detail in which micro and meso subnetworks are coded. Networks for micro simulation include usually (almost) all roads in the area of study. On the other hand, meso networks are usually much larger and include only a subset of the roads in the study area, excluding minor roads. When these networks are combined into a general network graph there are two problems:

a) Connectivity
b) Capacity

First there is a problem in connectivity (Figure 4). Each road that crosses the boundary between the meso and micro area, needs to be present in both (sub)network representations. How should one connect the many small roads in the micro area to the few large roads that are in the meso area? This also affects the definition of paths across the entire network.

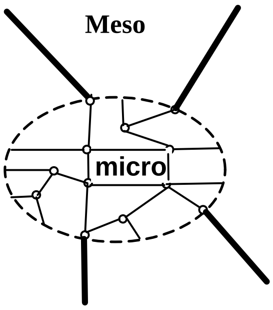


Figure 4. Problem connecting many micro to a few meso links

A practical solution to this problem may be to increase the level of detail in the meso subnetwork where it connects to the micro subnetwork, and maybe decrease the level of detail in the micro subnetwork, near the boundaries with meso.

The second problem in consistent network representation is that of the representation of capacity that is inherent in the detail of network representation. In the meso network representation, often *only* the larger roads are present, whereas the micro network – when applied to the same area – would include *both* the larger and the minor roads (Figure 5). This means that the capacity that is represented by both networks is different. When a certain area inside a larger meso network is replaced with a micro representation (which will include an additional set of minor roads), one has effectively increased the capacity of this area, relative to the remaining meso network. In terms of assignment, this will mean that the number of routes through the area that is now micro will increase compared to the all-meso case. This, in turn, will *increase* the amount of traffic that is assigned on routes through the micro-area, and *decrease* the amount of traffic assigned to routes that go outside of this area.

So in general, adding more detail to only a certain part of the network will result in a skewed traffic assignment.[4].

[4] This is also true for single-model traffic simulations when the area of interest is coded in more detail than the surrounding (buffer)network

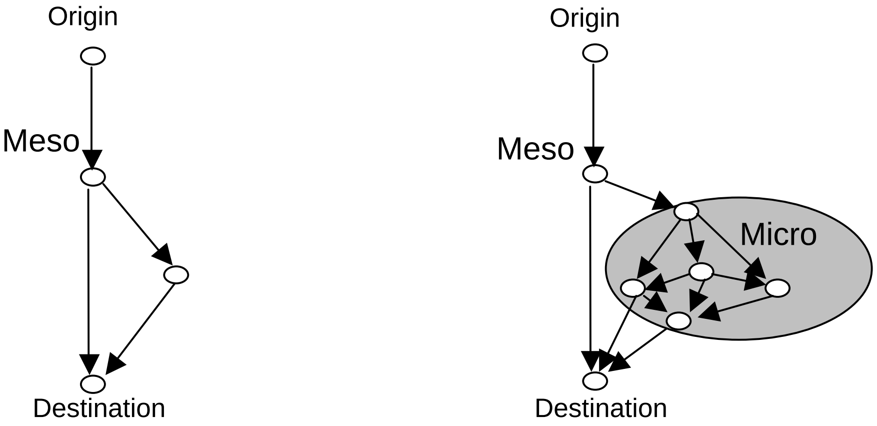


Figure 5. Level of detail in network representation. Left: meso only. Right: meso with micro sub-area.

For this reason it is important that both the meso and micro subnetworks include all relevant links, including minor roads. On the other hand, if the omitted minor roads are few, and carry little traffic, the shift in capacity may be negligible. In that case it may suffice to increase the level of detail in the meso subnetwork at the boundary locations.

Modelling traffic dynamics at meso-micro boundaries

The aspect that has received the most attention in the limited amount of literature available on hybrid simulation models is the way the two submodels are interfaced. In most cases the discussion focussed on aggregation / disaggregation issues between the macro and micro representations of traffic (Helbing, D. et al. 2002),(Magne, L. et al. 2000),(Bourrel, E. & Lesort, J.-B. 2003),(Kates, R. & Poschinger, A. 2000). In our case these issues are avoided by integrating two models that both have a vehicle-representation of traffic flow.

However, other issues concerning the interfaces between the models remain. In principle, the main sources of potential inconsistencies at the interface between the meso and micro models are:

a) Location of boundaries
b) Queue formation and representation
c) Vehicle variables (i.e. determination of speeds and accelerations of vehicles crossing the micro/meso boundaries)

Location of boundaries

There are a number of possibilities for placing the boundary between the micro and meso representations:

- On existing nodes
- On links using dummy nodes
- On links using transition zone

One possibility is placing the boundary at an existing node (such as an intersection) in the network. This would mean that some of the incoming and outgoing links would be micro links and some would be meso links. The difficulty that arises is how to deal with microscopic vehicles that need gaps to accept or reject and detailed traffic control on one hand, and the mesoscopic representation without the exact location and position of vehicles on the other hand.

Alternatively, one could place a dummy node on existing links, thereby cutting the links in two parts: one micro and one meso link. This clearly simplifies the task, since the correct boundary dynamics have to be assured for only one connection. However the dynamics still have to be translated immediately (from micro to meso or from meso to micro) at the boundary.

In case this immediate translation proves too difficult, another possibility is the creation of a transition zone, such as is used in (Bourrel, E. 2003),(Bourrel, E. & Lesort, J.-B. 2003). The transition zone is an area where both the mesoscopic and microscopic simulations control the movements of the vehicles. This would mean in case of the meso to micro transition, that vehicles that are coming from the mesoscopic area are projected *before they have actually exited the meso model,* in front of the microscopic continuation of the link. They would enter car-following behaviour, but at the same time corrections in their trajectories need to be made to make them match the exit times as generated by the meso model. However, this method makes the interfacing much more complicated due to the corrections that are needed to maintain a 'double control' over the vehicles. It seems that this option should only be used when it proves impossible to translate the dynamics directly at a dummy node.

In conclusion, the method adopted is to place dummy nodes on existing links, and to instantly translate the traffic dynamics from meso to micro and vice versa.

Queue formation and representation

The placement of boundaries is also dependent on the homogeneity of traffic conditions. Typically micro models, due to their level of detail, represent lane-specific queues. Meso models on the other hand, usually do not have lanes. If this is the case, a queue on one lane in micro, even if the other lanes remain open, will block the boundary node completely, unless the exchange of vehicles at the boundary is taking place properly. The virtual links in the meso subnetwork play an important role in solving the problem of inconsistent queue representation. Virtual links that represent paths that can only use the blocked lanes will be blocked, while virtual links representing paths that use open lanes will have available capacity and allow the proper advancement of the corresponding vehicles.

Crossing vehicle attributes

The attributes of the vehicles as they cross the boundaries need to be properly determined in both directions (micro → meso) and (meso → micro). Attributes such as speeds, accelerations, headways, etc should be consistent with the prevailing conditions in the new segment the vehicle is moving to, otherwise unnecessary shockwaves may propagate upstream.

Boundaries from Meso to Micro

On the boundary from the meso to the micro submodels, a number of conditions need to be met. Information is flowing in both directions: from meso to micro information about vehicles needs to be communicated; from micro to meso information about blocking of boundaries and downstream density and flow needs to be communicated. If the entry to the micro link (downstream of the boundary point) is blocked, for instance because a queue has backed up to the boundary, the meso needs to stop vehicles from exiting the upstream meso link. The micro model informs also when the blockage is removed and vehicles can start flowing (over that specific boundary) from meso to micro again. When a blockage is removed (for instance when the queue front has reached the boundary), the density and flow in the vicinity of the boundary is communicated to meso, where they are used to calculate the propagation speed of the shockwave that continues in the meso.

A more complicated issue is the generation of information that is needed in the micro representation of traffic, but missing in the meso one. While vehicles are represented individually by the meso model, they do not have the detailed characteristics needed in the micro model. Therefore these characteristics need to be created at the meso to micro boundaries.

The micro characteristics that need to be generated at the entry to the micro model can be divided into vehicle/driver *attributes* and *model variables*. Attributes such as desired speed and the critical gaps for lane-changing can be generated independently from the traffic situation upstream and downstream of the boundaries. However, model variables, such as the vehicle's speed, acceleration and time headway to the vehicle in front need to be *in accordance with the traffic situation upstream and downstream of the boundary*.

The variables that need to be assigned values at the entry of a vehicle from the meso into the micro models are usually: lane, time-headway to the vehicle in front, speed and acceleration. From the meso model the time-headway is determined, but not the speed, acceleration or lane. Based on the type of vehicle and the continuation of the vehicle's route, a set of candidate lanes is constructed, where all lanes that are disallowed for the vehicle (e.g. bus-lanes) or not connected to the continued path (so that an immediate lane change would be required), are taken out. From the candidate lanes the lane with the largest headway (space) is selected.
At this point the time headway and lane are known, but not the speed and acceleration. In section 4.3.4 a new method for the generation of initial speeds is presented in detail. The method assigns the initial speed to a vehicle that is to enter the microscopic network, based on the time headway to the vehicle in front, on the same lane. If the time headway is larger than a certain threshold, the vehicle to be loaded is assigned its desired speed. Otherwise, if the time headway is very small (below a certain threshold), and the vehicle in front has a speed lower than the desired speed of the vehicle to be loaded, the vehicle is assigned exactly the same speed as the vehicle in front. If the time headway is between the lower and higher threshold, the initial speed is a linear combination of the desired speed and the speed of the vehicle in front.

In almost all micro models, an initial acceleration rate of 0 m/s^2 is assigned to the vehicles that enter the network. While it may be possible to assign different values, it is difficult to obtain experimental data of accelerations over a specific sensor location with standard equipment (as opposed to time headways and speeds). In the future this kind of data may be available (as point-data, together with the time headways and speeds), and an equivalent algorithm may be developed that assigns non-zero accelerations based on the time headway and speed of the vehicle in front. At this moment, with a sufficiently good algorithm to assign initial speeds, the assignment of zero acceleration seems to suffice for our purposes.

Boundaries from Micro to Meso

In the interface from the micro to the meso model, the same kind of conditions need to be met as mentioned in the meso to micro case. In the first instance the meso needs to inform the micro each time the downstream meso link becomes blocked or unblocked, so that the micro model can stop / start sending vehicles at the right moments. In addition to the blocking, the vehicles in micro that move towards the exit to meso, need to react to the downstream traffic conditions, as they would if the downstream link were micro as well. In that case, the vehicles would react to vehicles in front, using their car-following logic, and those vehicles would react to vehicles in front of them, etc. In the meso model, however, the position and detailed behaviour of vehicles is not modelled. But it is known when the latest arrival of a vehicle was, and the (average) speed it was assigned. Using this information a *virtual vehicle* is projected in the continuation of the micro link, and vehicles in micro that are near the exit *react to the virtual vehicle*, as if it were a normal vehicle in front of them (Figure 6). This idea can also be found (in a slightly different form) in (Bourrel, E. & Lesort, J.-B. 2003)and (Magne, L. et al. 2000). An obvious refinement is to have each lane have its own virtual vehicle.

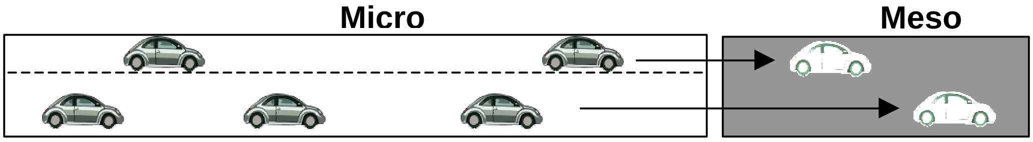


Figure 6. Exiting vehicles in micro follow virtual vehicle

In short, every vehicle in the micro area that has no leading vehicle in front of it (provided the next link is a meso link) will follow a virtual vehicle whose position and speed is determined as follows:

1. The current time $t_{current}$ and the time the last vehicle (on that lane) exited t_{exit}
2. The speed that was assigned in meso to the exited vehicle $v_{virtual}$ (as a function of the density on the meso link)
3. The extrapolated position: end of micro link + $v_{virtual}$ * $(t_{current}-t_{exit})$

This procedure requires that meso communicates to micro the speeds of vehicles that enter, and that micro keeps track of the last time a vehicle exits on each lane of outgoing links.

A possible source of problems in the transition for vehicles in micro from following a real vehicle to following a virtual vehicle lies in possible sudden changes in speed. The speed of the lead vehicle in micro (that is about to exit and become a virtual vehicle) is determined by many factors, including desired speed, and possibly the speed of a virtual vehicle in front. But as soon as it exits, its speed is determined exclusively by the density in the meso link downstream.
However, in those cases where the virtual vehicle is relevant for the following vehicle, i.e. short time headways, and a speed equal to or lower than the following vehicle, it is likely that the speed of the (now virtual) vehicle in front was already 'determined' by the density in the meso link before it exited the micro area: it was probably also following a virtual vehicle before it exited. In other words, the changes in traffic conditions are expected to move smoothly over the boundaries.

4.3.3 A new method for generation of initial speeds

In micro simulation models the generation of initial speeds (and time headways) is usually done using either deterministic speeds, or using a stochastic process. Unfortunately, the process of initial assignment of time headways and speeds is seldom discussed in the literature on particular simulation models5. Often (Mitsim (Yang, Q. 1997), Vissim (PTV 2003)) the desired speeds of vehicles are drawn from a certain (measured / estimated) distribution and the vehicles are assigned a speed upon entry that is based on this desired speed. Deterministic speeds are undesirable from the point of view of a realistic representation of the traffic process, but the stochastic generation of speeds presents us with problems when speeds are generated that are independent of the time headway and the speed of the vehicle in front. Due to this independence, some of the vehicles will start to accelerate upon entry, and some of the vehicles will need to decelerate in order to create a sufficient (time) headway to the vehicle in front.

In order to study the effects of initial speed generation on the initial behaviour of entering vehicles and capacity of the entry links, some experiments in MiMe (Mezzo and Mitsim) are performed. The network consisted of three 1km long 2-lane links; one link in Mezzo, followed by one in Mitsim, followed by one in

[5] An exception is (Bladh, P. 1996) where a general traffic generation model is presented that combines lane-dependent time-headway generation, lane-distribution and headway- and flow-dependent speed generation. The model components are rather rich in parameters, which have been estimated using data from Swedish two-lane motorways. It explicitly considers left-right lane dependencies.

Mezzo. The speed limit was 55 mph and a base free flow speed of 60 mph. The demand was varied from 3000 veh/h (1500 veh/h/lane) in the first five minutes of simulation, gradually increasing to 5000 veh/h at t=30 min. From t=40 min. the demand drops to 4000 veh/h and at t=50 min to 3000 veh/h.

The initial accelerations of the vehicles immediately upon entry are studied, and the flow and speed over time at two points: sensor 0 exactly at the entry of the micro link, and sensor 1 at 500m. downstream from the entry.

The original method for generation of speeds in Mitsim works as follows. Each vehicle is assigned a desired speed based on the driver/vehicle type and the associated distribution. This desired speed is adapted to the local conditions on the link, taking into consideration the speed limit and the level of compliance of that driver (which is also drawn from a random distribution). Based on the level of queuing on the link the vehicle is loaded with a speed varying from its (modified) desired speed in case of no queues, to the segment average speed in case of a large queue.

The problem of initial disturbances in the accelerations manifests itself clearly in figure 7, where the acceleration rate in the first 20 seconds (after entering the micro link) is shown, averaged over all vehicles in the 1 hour simulation period.

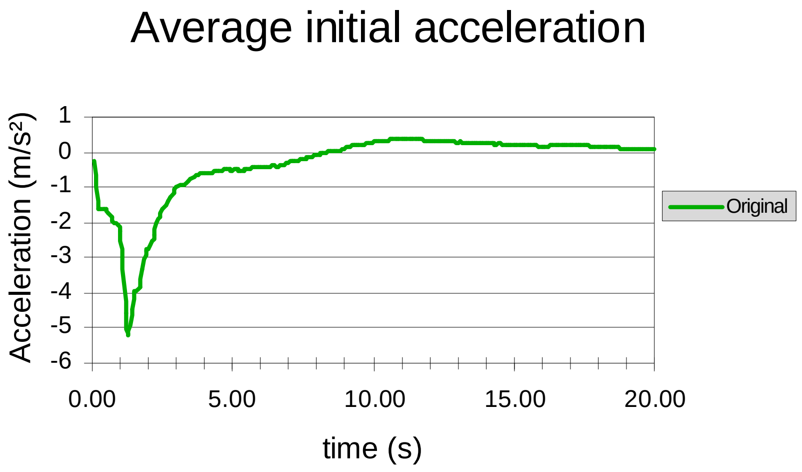


Figure 7. Average initial acceleration in Mitsim using the original stochastic speed generation method.

As can be seen in Figure 7, there is a clear tendency for vehicles to break hard in the first 2 seconds upon entering the micro link. This means that many vehicles enter the link with a speed that is too high given their time headway and the speed

of the vehicle in front. This behaviour has been recognized before, but it has been assumed to occur only in the first section and therefore not have a large effect on the simulation results, provided that the entry links are long enough so that any disturbances are evened out by the time the vehicles exit these links.
First of all there is no reason to assume that these disturbances will even out and create the dynamics found in real traffic streams. More important, however, is the fact that the disturbances at the entries cause the capacity of the entry links to be artificially limited, as can be seen from the flow and speed over time (see figure 8). This limited capacity will cause a capacity breakdown and queues to build up at flows well below the usual capacity. As a benchmark (upper limit) it is noted that under deterministic time headways and speeds, the capacity in Mitsim exceeds 2600 veh/h/lane (for 5 min flows).

In figure 8 a maximum flow of around 2000 veh/lane is observed for both sensors. Sensor 0 is located directly at the entry, while sensor 1 is located 500 meters downstream. From the speed diagram one can observe immediately that the average speed at the entry (sensor 0) is much higher than the speed 200m downstream (sensor 1). Moreover, it can be observed that the speed at sensor 0 drops when the input flow (demand) increases and stays low until the end of the simulation. Note that the flows over the two sensors are virtually the same.

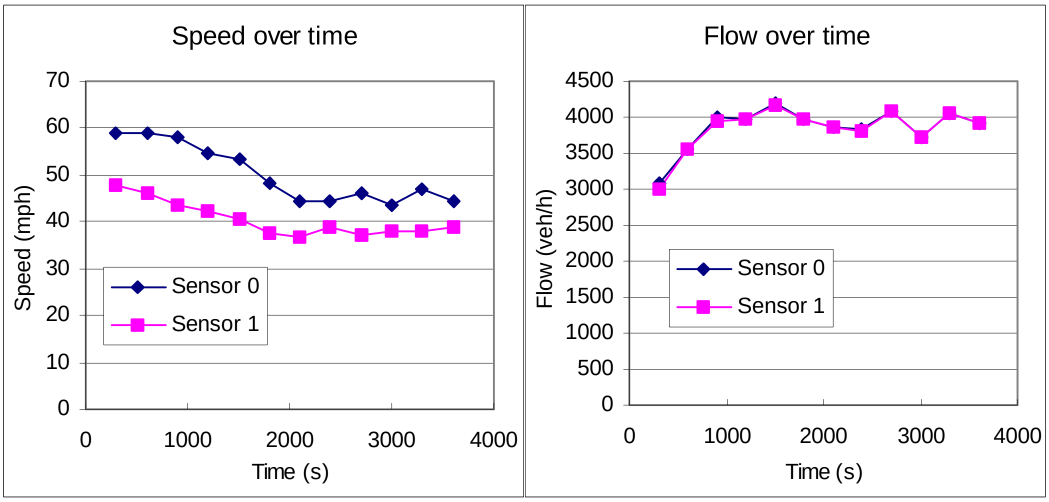


Figure 8. Speed over time (left) and Flow over time (right) for sensors 0 and 1.

Based on these considerations, four candidate methods for generation of initial speeds were tested. They all take into consideration the dependency of a vehicle's speed on the traffic situation ahead. The candidate methods tested are:

1. Mezzo speed/density function applied on lane density. Using the density in the lane where the vehicle will enter, a speed/density function is used to determine the initial speed.
2. Lane average speed. Assign the average of the speeds of all vehicles in the selected lane.
3. Same speed as vehicle in front.
4. 3-regime method[6]
   a) Regime 1 (bound traffic): $t_1 < t_h <= t_2$
      $V = V_{front}$
   b) Regime 2 (partially bound traffic): $t_2 < t_h <= t_3$
      $\alpha = (t_h - t_2)/(t_3 - t_2)$
      $V = \alpha * V_{desired} + (1-\alpha) * V_{front}$
   c) Regime 3 (unbound traffic): $t_h > t_3$
      $V = V_{desired}$

Where,

t_i	= threshold, to be calibrated
t_h	= time headway (s)
V	= speed (km/h)
V_{front}	= speed of vehicle in front (km/h)
$V_{desired}$	= desired speed (km/h)

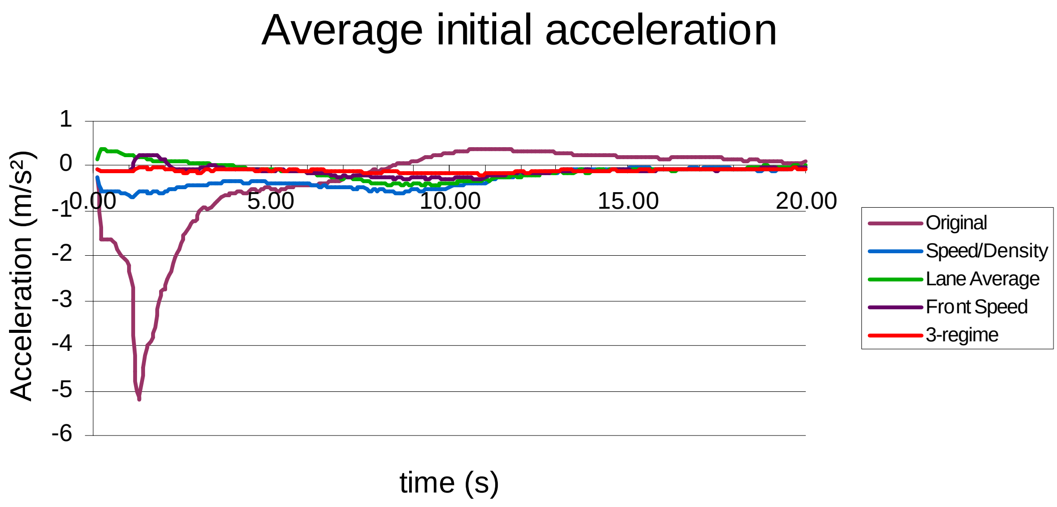
In all four methods the actual loading speed is min $(V, V_{desired})$ so that no vehicle is loaded with a speed higher than its desired speed.

The 3-regime method was inspired by the "known" correlation of speeds of close-following vehicles on one hand, and the decreasing degree of correlation with increasing time-headways on the other hand. The time-headway to the vehicle in front is used to determine whether the vehicle's speed is bound by the vehicle in front (regime 1), partially bound by the vehicle in front (regime 2) or not bound by the vehicle in front (regime 3). The limits that separate the three regimes are time headways of $t_2 = 2.5$ and $t_3 = 7.5$ seconds, which have been deduced from observations of time headways and speeds of vehicles in the same lane (as reported later on), as well as the minimum headway $t_1 = 0.5$ seconds.

[6] In (Bladh, P. 1996) the speed generation is also dependent on the speed of the front vehicle and the time headway (as well as the flow and for the right lane also the speed of the previous vehicle in the left lane). For time headways smaller than 1 second the speed of the front vehicle is assigned, for headways equal or larger than 1 second, a function is used that depends on the flow, time-headway, the desired speed and the speed of the front vehicle.

As can be seen from figure 9 (top), all of the proposed speed generation methods improve the loading of the vehicles considerably. Where the minimum average acceleration of entering vehicles was around –5 m/s² using the original method, is the minimum now –0.7 m/s². From the figure 9 (bottom) it shows, however, that there are clear differences in the performance of the four new methods tested.

The speed/density method (assigning a speed based on the segment density downstream) still seems to load the vehicles at a speed that is (on average) too high, still leading to an average deceleration in the first seconds after entering the network. Probably the reaction of density to local conditions directly downstream of the loading point (such as a spontaneous capacity breakdown during the period of highest inflow) is too slow.



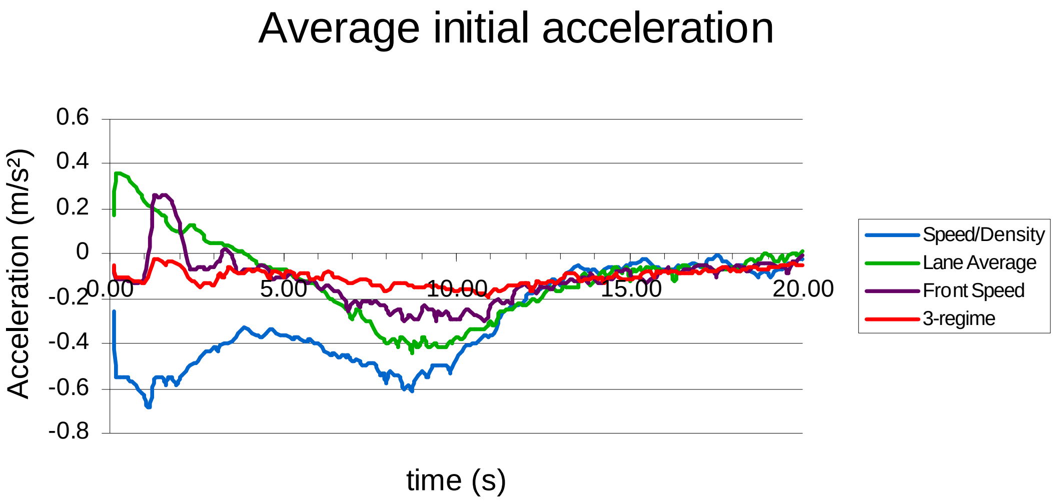


Figure 9. Average initial accelerations of vehicles upon entering the simulation.(above) inclusive the original speed generation method, (below) exclusive the original method.

The second method (lane average) assigned each vehicle the average speed of all vehicles (downstream) in the lane in which it was entered. This resulted (on average) in an acceleration followed by a deceleration with a minimum reached at 10 seconds from entering the link. Here it seems that the initial speed is on average too low leading to initial accelerations followed by modest decelerations later on. Also this method is hypothesised to react too slowly to local disturbances in the traffic flow.

Therefore the third method was tested (front speed). This consisted of simply assigning the speed of the vehicle in front (in the selected lane). The average acceleration pattern produced by this method shows a marked acceleration about one second after entering the network, and a modest deceleration afterwards, with a minimum around 10 seconds after entry.

The fourth method tested (3-regime) was a refinement of the third, by making the initial speed a function of both time-headway and the speed of the vehicle in front. This method shows very little average accelerations compared to the other methods, varying from 0.0 to –0.2 m/s^2.

In figure 10 the (5 minute) average flows are shown over sensor 0 (directly at the entry of the link) and sensor 1 (at 500m from the entry point) for the one hour simulation period. Observing the Original method, it can be noticed that after t=1600 the flow stagnates at around 2000 veh/h/lane, and never recovers during the simulation period. Remember that the demand pattern has a peak at t=1800 – 2400 seconds (5000 veh/h) and then decreases gradually to 3000 veh/h in the last 10 min. period. The other methods follow the peak and decrease in the demand much better, with the 3-regimes showing the highest 5 minute flow at t=2400. This demonstrates the effect of the speed generation method on the implicit capacity of the entry link

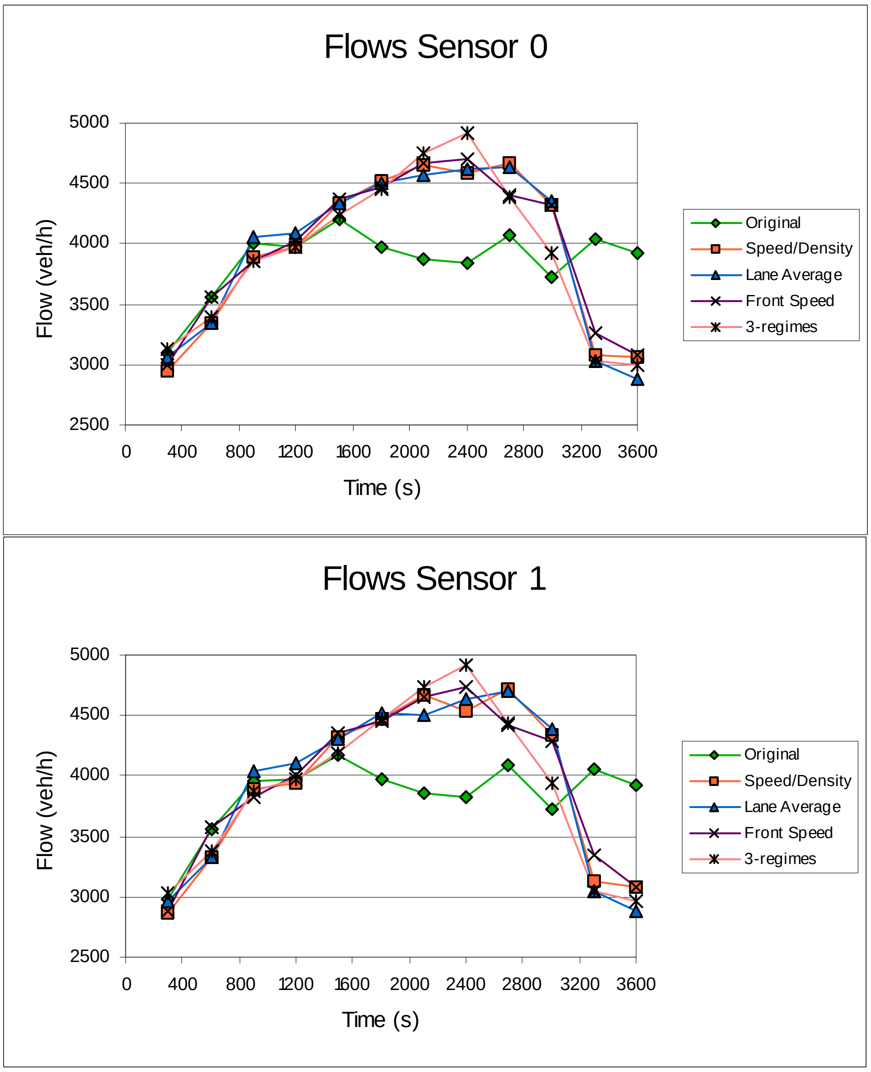


Figure 10. Average flows over time for sensor 0 (entry of link) and sensor 1 (500m downstream of entry).

In figure 11 the (5 minute) average speed is shown for sensors 0 and 1. Both the original method and the speed/density method show initially high speeds over sensor 0. For the same time periods the speeds over sensor 1 are much lower (this difference is larger for the original method). This strengthens the hypothesis that the original loading speeds are too high using these methods. The difference between these two methods is shown at the end of the simulation period, where the speeds seem to recover in the case of the speed/density method, but not in the original methods case.

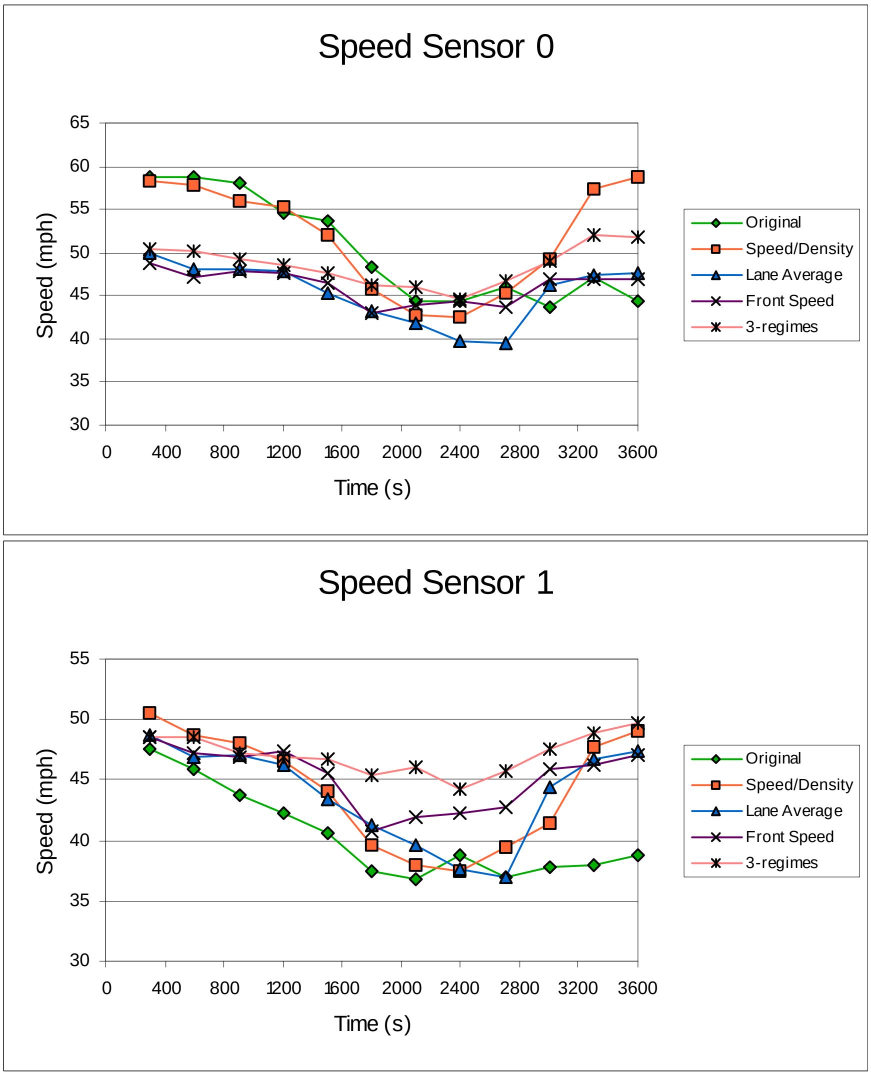


Figure 11. Average speed over time for sensor 0 and sensor 1.

The other methods show a much lower speed over sensor 0 and the difference between the speeds over sensor 0 and 1 is much smaller. However here we still observe a secondary effect in case of the lane average method, where the drop in speed during the high-flow period (t=1800-2400s) is most pronounced, and the drop is larger over sensor 1 than over sensor 0.

The greatest stability is shown in the case of the 3-regime method, where the speeds over both sensors follow the same pattern, and the drop in speed at high flow is minimal (5 mph).

Comparison with freeway measurements

The 3-regime algorithm was also compared to measurements of speeds and time-headways of vehicles on the same lane on an urban freeway in Stockholm[7]. The data consisted of high-fidelity measurements of time-headways and speeds (with a reported variance of 0.005 seconds for time headways and 0.1 km/h for speeds), during and after the morning peak (7:30 – 11:30). The average flow during the period was around 1500 veh/h/lane, and the average speed around 90 km/h.

First, the data is grouped into classes (class-size 1 second) according to the time-headway between the vehicles. Then the correlation of a vehicle's speed with that of the vehicle in front is calculated for each time-headway class[8]. In figure 12 two examples are given of scatter-plots of the speed of a vehicle against the speed of the vehicle in front, for the time headway class <1.0,2.0] (left) seconds and for class <7.0, 8.0] seconds. From these figures it is obvious that small time headways result in a much higher correlation in (following) speeds than higher time headways.

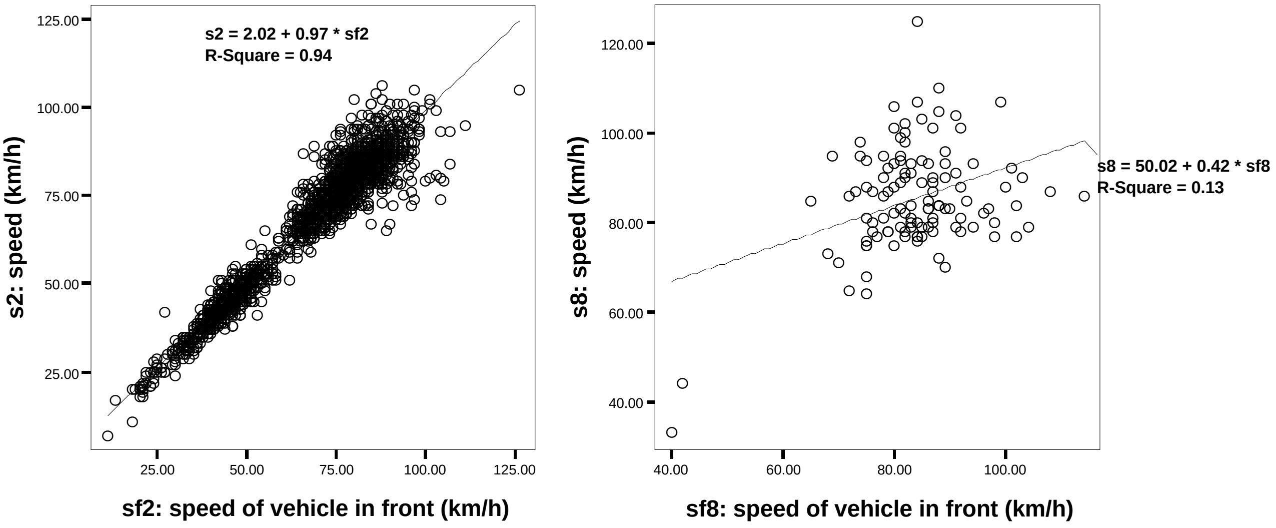


Figure 12. Speed of a vehicle against the speed of the vehicle in front (in the same lane). (left) time-headway class 1-2 seconds. (right) time-headway class 7-8 seconds.

[7] The author wishes to thank Allogg AB and the Swedish National Road Administration for providing this high-quality data.

[8] In (Vogel, K. 2002) the correlation of following vehicle speeds is also plotted against the time-headway of the following vehicle, in order to determine one cut-off point between free and non-free vehicles in urban situations.

The correlations of the speeds with speeds of vehicles in front were calculated for all time-headway classes (and were all significant at the 0.01 level) and are plotted in figure 13.

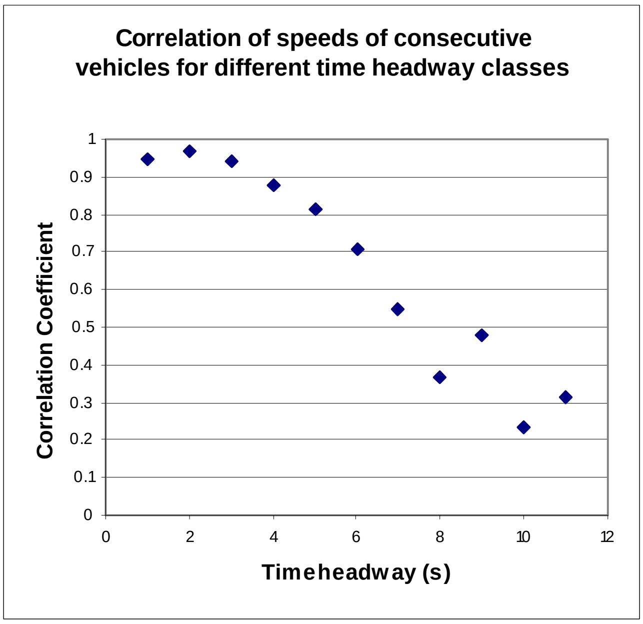


Figure 13. Correlation of speeds of consecutive vehicles for each time headway class (class size = 1 second)

The measurements show high correlation of speeds between consecutive vehicles on the same lane, in the case of small headways. This correlation decreases with increased time headways, and beyond (approx.) 7.5 seconds remains at a low level. This supports the relevance of the method, and suggests that the break-off points of 2.5 and 7.5 seconds between the three regimes which had been observed in initial tests in Mitsim to provide the best initial acceleration behaviour. As can be seen from figure 13, the correlation seems to go up at the 8-9 seconds time headway class. This may be because the number of observations per class varies greatly, with the majority of measurements being in the 1-2 second time headway class (2132 observations) and the number of observations rapidly decreasing to 115 observations in the 8-9 seconds headway class.

4.4 Implementation of the hybrid model

The hybrid model framework that is introduced in this chapter has been implemented using the Mezzo mesoscopic model and the MITSIMLab (Mitsim)

(Ben-Akiva, M. et al. 1997) microscopic model. Both models needed modifications and additions to provide for the communication and synchronisation mechanisms, as well as for the generation of information needed by the other.

4.4.1 Communication and synchronisation

The Mezzo and MITSIMLab (Mitsim) models need to communicate information on vehicles, travel times, traffic conditions, etc. In addition, they need to synchronize to avoid one model running faster than the other.

There exist a number of technologies for distributed computing that can be applied. The criteria for selection of the technology are in this case: speed of communication, message encoding/decoding overhead, platform independence and memory and CPU overhead. The most well-known method for platform independent distributed computing is CORBA (OMG 2004), which provides much more than our needs of message passing and synchronization (such as object sharing etc.). However, this is a rather large and complex system that introduces a serious overhead on CPU, memory and communication. Microsoft's DCOM (Microsoft 2004) is also available on both Linux and Windows platforms, but suffers from instability under Linux and relatively large communication overhead on both platforms. Parallel Virtual Machine (PVM) (Alexandrov, V. & Dongarra, J. 1998) and Message Passing Interface (MPI) (MPI 2004) are technologies that were especially developed for parallel computing on heterogeneous sets of computers. This means that the communication overhead and memory and CPU footprint of these systems is small and that it lets Linux and Windows computers interoperate in the distributed computing task. Given the fact MITSIMLab already has an implementation of PVM, this was the system adopted for this project, but there is no reason to assume PVM to be superior to MPI for our task.

For both communication and synchronization tasks the PVM is used, which provides a mechanism to pass messages between processes in the same or different physical computers. In this case the Mezzo and MITSIMLab run as two processes within the virtual machine, exchanging messages. This message passing is used to synchronize the models, as well as to communicate the necessary information (see figure 14).

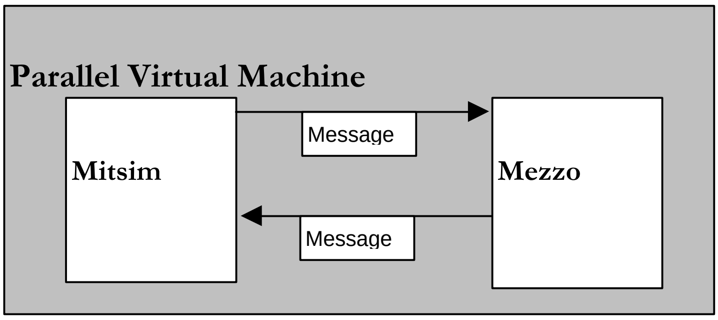


Figure 14. Message passing in the virtual machine

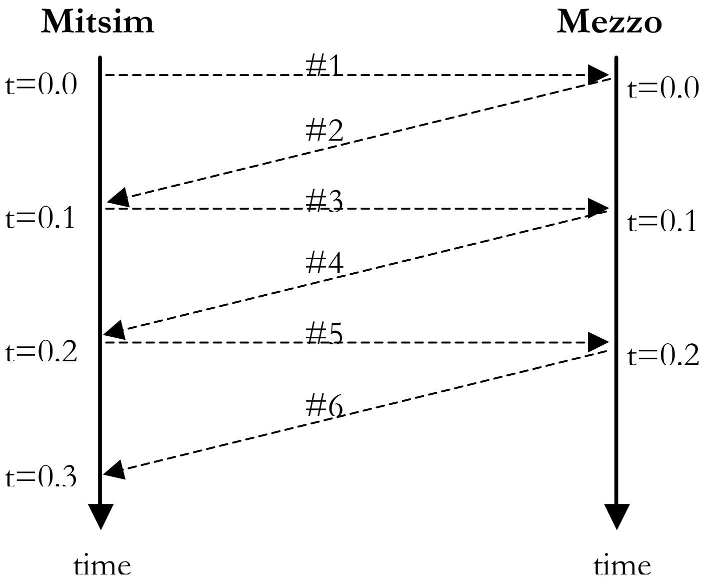


Figure 15. Event diagram of message passing between Mezzo and Mitsim

As can be seen from figure 14 and 15, the messages are passed between Mitsim and Mezzo through the virtual machine. Each microscopic time-step (normally 0.1 seconds) Mitsim sends a message with the relevant information for Mezzo. Mezzo receives this message, decodes it and encodes and sends its own information *immediately* back in a message to Mitsim. In the example in figure 15 the transmission time for messages is assumed to be negligible.

The synchronization works as follows: Each Mitsim time step (except for the first one) Mitsim waits for a message from Mezzo, sends a new message back and continues to the next time step. Mezzo waits for a message from Mitsim, sends back a reply, and books an event at (t+0.1) in the event list to look again for a message from Mitsim. Mezzo then continues the simulation by processing the events in the event list until the booked communication event comes up.

Message structure

The messages that are sent between Mezzo and Mitsim and vice versa are structured in a specific way. All boundary-out nodes collect the vehicles that want to cross the inter-model boundary during the time interval (0.1 second). For each vehicle that crosses the boundary a *signature* is created, containing the information that is essential for the other model, as well as the information that is needed to keep track of the vehicle throughout the model In figure 16 the Signature Class is shown.

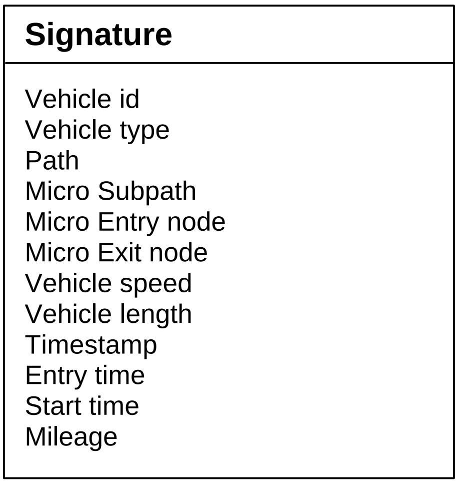


Figure 16. Signature class for communication of vehicle information

Both models have the same Signature class included, which ensures that information is communicated correctly between them. This also allows for easy extension or modification of the attribute list in the future.

The vehicle id is unique for each vehicle throughout the model. The vehicle type indicates if the vehicle is a normal car, a bus, truck etc. The path indicates the id of the vehicle's overall path through the network, including virtual links. Each virtual link in a vehicle's path is converted into a micro subpath when the vehicle reaches the boundary node. The boundary-out and boundary-in nodes in the meso model are converted into micro entry node and micro exit node. The meso vehicle speed is communicated, but (at this moment) not used, as the microscopic model has its own loading mechanism, and the mesoscopic model uses speed/density functions to assign speeds upon entry of a vehicle. The vehicle length is communicated explicitly to ensure that the density of a specific traffic stream remains the same whether it is in micro or mesoscopic representation. (different vehicle lengths result in different densities). The timestamp indicates the exact moment the vehicle arrived at the boundary node (in Mezzo time is continuous) within the time step of communication. The start time is the time the vehicle entered the network, and the mileage the amount of distance travelled since entering the network. Both are kept for data collection at the destinations, where vehicle travel times and mileage are collected for all vehicles. The entry time is the time the vehicle entered the current link.

Apart from the signatures for all vehicles that cross the boundaries, each message contains specific information for the other model to ensure the correct propagation of traffic conditions. In the case of Mezzo, each boundary-in node will send the information of the virtual vehicle (speed and distance from entry node), as well as whether the downstream link is full or not (block/unblock). Mitsim sends in each message for all boundary-in nodes the flow and density on the downstream segment (which are used in Mezzo to calculate the speed of shockwaves), as well as whether the downstream segment is full or not.

When a model receives the information about blockage of the downstream segment, it pauses the sending of vehicle signatures (for that boundary node) until information is received that there is room again for new vehicles.

4.4.2 Substituting Mitsim for another microscopic model

The implementation described in this section can be used to integrate other mesoscopic and microscopic models. While there are currently no mesoscopic models other than Mezzo that would be easily integrated with a microscopic model, the Mitsim micro model could be substituted by another model.

For such a substitution to be possible, some degree of flexibility is needed on the part of the microscopic model to be integrated. First of all, the necessary communication (for instance using the PVM, but other mechanisms are possible) needs to be added to the operation of the model. This requires either access to the program source code, or an API (Application Programming Interface) that is powerful enough to add a module to the simulation model that can do the following:

1. Send and receive vehicle signatures from/to Mezzo.
2. Create vehicles from the signatures and enter them into the simulation
3. Extract vehicles that exit the micro area, convert them to signatures and send these signatures to Mezzo

These three points are essential for the possibility of integration. In addition, a number of other functions need to be provided by the micro model to achieve correct hybrid functionality:

4. Routing along a pre-specified path that corresponds to the virtual link
5. Attachment of a signature to the vehicles, so that no new signatures need to be created when the vehicle reaches an exit node.
6. Addition of a virtual vehicle that functions as the lead vehicle for vehicles approaching the micro-meso boundary.
7. Communication of density and flow in the first segments of entry links.
8. Specification of vehicle types and their lengths (this is satisfied by almost all existing micro models)

Most likely the vehicle-loading algorithm (the way the vehicles obtain their initial speed, lane and acceleration) needs to be adapted as well, for instance using the method outlined in this chapter.

If the microscopic virtual links are required for en-route changes inside the micro model, the route choice model and network representation within the micro model needs to be adapted.

Chapter 5. Evaluation of Mezzo and MiMe

5.1 Introduction

In this chapter the MiMe hybrid meso-micro model and the Mezzo mesoscopic model are evaluated using both laboratory tests and tests with field data. The microscopic model in MiMe, MITSIMLab (Mitsim), has already been validated in a number of studies (Ben-Akiva, M. et al. 1997), (Ben-Akiva, M.E. et al. 2000) and (Toledo, T. et al. 2003), (Burghout, W. 2001).

First a number of goodness-of-fit measures are introduced in section 5.2. In the following section Mezzo is evaluated in three steps. First the fundamental speed/flow/density diagrams are examined for basic correctness, then the model is applied and calibrated on a small network, using the measured speeds (and approximated densities), while the flows are compared against measurements. Then the model is tested on a small/medium-sized network, and besides the traffic performance, the behaviour of the dynamic traffic assignment and route choice is studied as well.

In the fourth section of this chapter the MiMe hybrid model is evaluated in a number of steps. First the meso-micro and micro-meso boundary consistency is checked using laboratory tests. An incident is created to study the propagation of traffic dynamics such as congestion and queue dissipation over the submodel boundaries. Then the model is applied to a medium sized network north of Stockholm (Brunnsviken). The chapter concludes with a comparison of Mezzo, Mime and Mitsim on the Brunnsviken network.

5.2 Goodness-of-fit Measures

There are a number of measures that are commonly used to test the goodness-of-fit of a model to measured data. One of the most common is the root mean squared error (RMSE).

$$RMSE = \sqrt{\tfrac{1}{N}\sum_{i=1}^{N}\left(m_i^{sim} - m_i^{obs}\right)^2} \qquad (1)$$

Where,

m_i^{sim} = simulated measurement for period i

m_i^{obs} = observed measurement for period i

N = number of time periods

While the RMSE is a standard measurement for the goodness of fit, it suffers from a drawback. Because the absolute differences between observed and simulated values are used, small deviations for high flows contribute as much to the RMSE value as the same values for low flows. If we want to account for the fact that a difference of (say) 100 veh/h for flows of 500 veh/h indicates a larger deviation than the same difference for flows of 2000 veh/h, it is better to use the root mean square normalised error (RMSNE).

$$RMSNE = \sqrt{\tfrac{1}{N}\sum_{i=1}^{N}\left(\frac{m_i^{sim} - m_i^{obs}}{m_i^{obs}}\right)^2} \tag{2}$$

A third common measure to indicate if there is a systematic bias (a consistent over- or under-prediction by the model) is the mean normalised error (MNE).

$$MNE = \tfrac{1}{N}\sum_{i=1}^{N}\frac{m_i^{sim} - m_i^{obs}}{m_i^{obs}} \tag{3}$$

Besides the RMSE, RMSNE and MNE there exists another measure of fit that has been used mainly in econometrics. Theil's U statistic (Pindyck, R. & Rubinfeld, D. 1997)[9] is defined as follows:

$$U = \frac{\sqrt{\tfrac{1}{N}\sum_{i=1}^{N}(m_i^{sim} - m_i^{obs})^2}}{\sqrt{\tfrac{1}{N}\sum_{i=1}^{N}(m_i^{sim})^2} + \sqrt{\tfrac{1}{N}\sum_{i=1}^{N}(m_i^{obs})^2}} \tag{4}$$

[9] There seem to be two definitions of the Theil U statistic. (Theil, H. 1961) defines the U statistic in the way used here, while (Theil, H. 1966) provides an updated statistic, which Theil argues is better since the nominator is now only dependent on the actual observations. The new statistic can produce values that are greater than 1. A more important issue is the fact that Theil's U (both versions) was developed for auto correlated forecasting time-series. That means it compares the *changes* from i to i+1 for observed and measured data. In this sense it is also used in (Barcelo, J. & Casas, J. 2004) where the (Theil, H. 1966) statistic is used for comparison of simulated and actual autocorrelated time-series.
However the most commonly used version considers the *differences between the simulated and observed measurements for all points i* ((Hourdakis, J. *et al.* 2003),(Pindyck, R. & Rubinfeld, D. 1997), (Ahmed, K. 1999)), and is based on the (Theil, H. 1961) measure. This is the measure used here.

Where,

m_i^{sim} = simulated measurement for period i

m_i^{obs} = observed measurement for period i

N = number of time periods

This measure will always fall between 0 and 1, where U=0 means a perfect fit. The U can be decomposed in three proportions, U^M, U^S, and U^C:

$$U^M = \frac{(\overline{m}^{sim} - \overline{m}^{obs})^2}{\frac{1}{N}\sum_{i=1}^{N}(m_i^{sim} - m_i^{obs})^2} \tag{5}$$

$$U^S = \frac{(s^{sim} - s^{obs})^2}{\frac{1}{N}\sum_{i=1}^{N}(m_i^{sim} - m_i^{obs})^2} \tag{6}$$

$$U^C = \frac{2(1-r)\, s^{sim}\, s^{obs}}{\frac{1}{N}\sum_{i=1}^{N}(m_i^{sim} - m_i^{obs})^2} \tag{7}$$

so that $U^M+U^S+U^C=1$.

Where,

$\overline{m}^{sim}$ = sample mean of the simulated measurements over all periods

$\overline{m}^{obs}$ = sample mean of the observed measurements over all periods

s^{sim} = sample standard deviation of the simulated measurements

s^{obs} = sample standard deviation of the observed measurements

r = sample correlation coefficient between the two series

U^M, U^S, and U^C are usually interpreted as the bias, variance and covariance proportions of the errors. By decomposing the error, this measure gives information about the mean and variance proportions of the error (which are the ones that modellers usually worry about) and the covariance proportion (also called unsystematic error). On the downside is the objection that the other measures (RMSE, RMSNE and MNE) may be more intuitive and therefore easier to interpret. In calibration studies the decomposition of the U measure is appreciated as a tool to obtain more specific information about the nature of the errors, which then can direct the calibration efforts.

In the following sections these measures of goodness-of-fit are used as appropriate. The Theil U is only calculated for the Brunnsviken network case studies for Mezzo and MiMe, and its decomposition is presented in the comparison of MITSIM, Mezzo and Mime in section 5.5

5.3 Evaluation of Mezzo

In this section the Mezzo model will be evaluated for correctness. First the fundamental diagrams (Speed / Flow / Density) are discussed, to show the basic correctness of the link behaviour. Then the model is applied to a small freeway corridor (2,5 km) with one on-ramp and one off-ramp, and the propagation of flow over time is compared against measurement data from sensors. The third evaluation is on a 5 x 5 km network north of Stockholm (Brunnsviken). Here the behaviour of the route generation and the dynamic traffic assignment is studied qualitatively, in addition to a comparison of flows on detector locations against measured data.

5.3.1 Fundamental diagrams

The fundamental diagrams for Mezzo were created using a simple network consisting of 15 consecutive link segments which are each 300 meter long, have two lanes and a free flow speed of 83 km/h (similar to the one used in section 3.6.4 to show shockwave propagation). An incident is introduced to produce the high density / low flows and speeds range of the diagrams. In figure 1 the network and incident location is depicted.

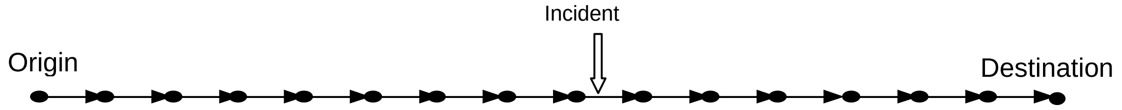
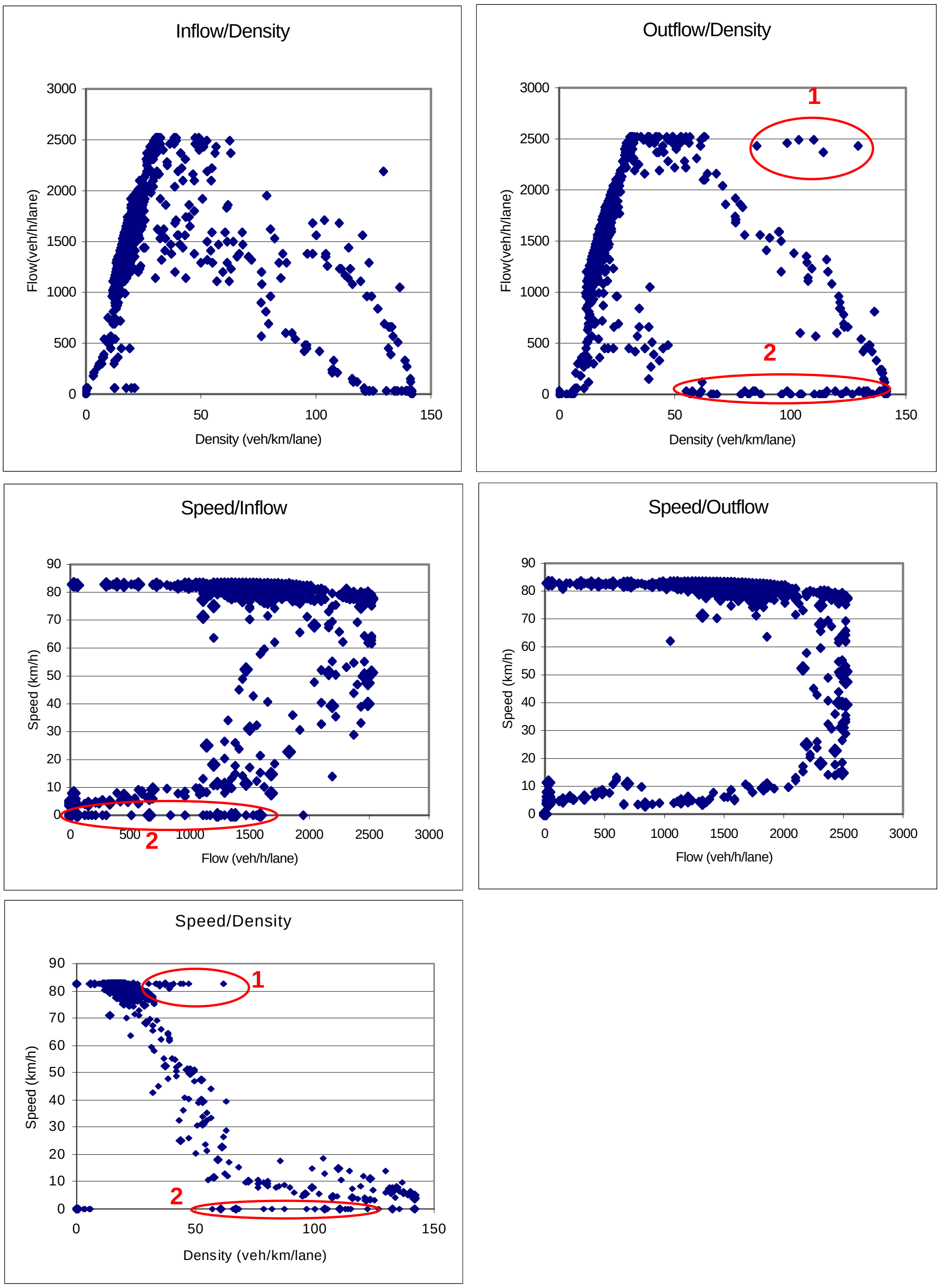


Figure 1. Network to generate data for speed/flow/density diagrams

The data in figure 2 presents 1-minute average flows, speeds and densities for all links, including the one where the incident is created. The simulation was repeated 5 times to generate a large enough dataset. The resulting fundamental diagrams show relations between flow/density, speed/flow and speed/density, which are in accordance with those usually measured. See (May, A.D. 1990) for the examples of typical diagrams. Further on in this section the fundamental diagrams of a typical freeway (in the north of Stockholm) are shown.



Inflow/Density

Outflow/Density

Speed/Inflow

Speed/Outflow

Speed/Density

Figure 2. Fundamental Diagrams for Mezzo using incident scenario

The ellipses indicated with number 1 (density / outflow and speed/density) show the data points (for all five replications) for the incident link, at the moment of queue dissipation. At that moment the density on the whole link is high, but the speed and outflow are relatively high. In normal conditions (when there is no queue dissipation) these high densities would lead to lower speeds and lower flows. Another effect of the incident that is created, is the zero speed and outflow during queue build-up (ellipses with number 2). Due to the blockage of the exit, the outflow is 0, while the density is increasing, and the speed recorded is also 0, since no vehicles exit, even though there are still vehicles flowing into the link (until it fills up completely).

One difference between measurement data and data produced by Mezzo is that the speeds reported are the space mean speeds over the link segments, rather than the point speeds obtained by loop measurements. Due to the fact that the vehicles in Mezzo are moved using speed/density relationships that give space mean speeds, and queue-server mechanisms that may add delays, the vehicles positions and instantaneous speeds on the links are not tracked, and therefore cannot be part of the model output. This is true for all (mesoscopic) models that do not track vehicles' positions and speeds on the link.

Another difference is that both the in- and out-flow on the links is reported, whereas the measurements report the flows at a specific location. These two aspects mean that the plots in figure 2 may look different from a speed flow diagram with point speeds.

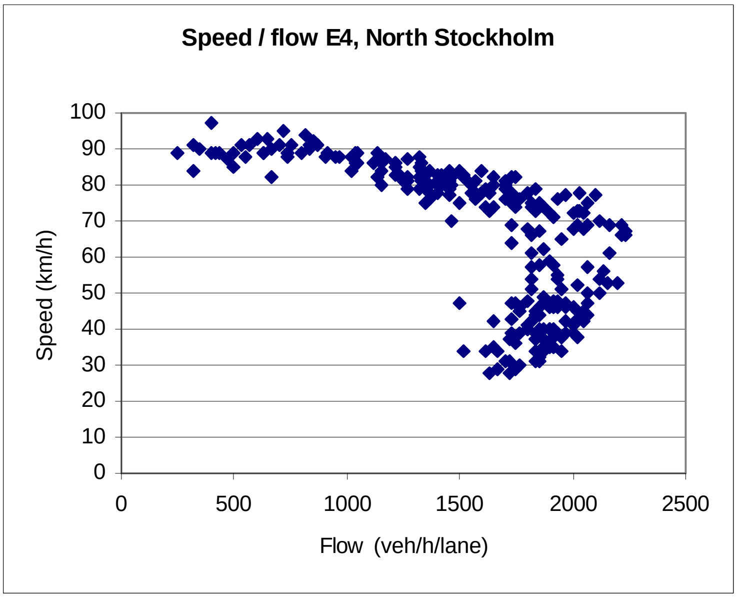


Figure 3. Typical speed /flow relationship (Freeway E4, North Stockholm)

In figure 3 a speed / flow diagram for a freeway in the north of Stockholm is shown. It is clear that the general pattern matches that of the speed / inflow and speed / outflow diagrams in figure 2.

One observation is that the pattern matches the speed/inflow diagram from Mezzo better than the speed/outflow diagram. In other words, the space mean speed on a link in Mezzo, as a function of the in-flow matches better the pattern of measured speeds and flows on the same spot, than in the case of space mean speed as a function of the link outflow. One place where the speed/outflow diagram differs from the pattern found in measurements, is the high outflow at low *space mean speeds*. One cause for that is that the space mean speed is calculated from the travel time for vehicles on the link, divided by the link's length. If a vehicle is queued for a long time on the link, when the link exit is blocked for instance, and the link blockage disappears and the outflow recovers, the space mean speed of the vehicles exiting the link is going to be low, but the flows are high. If the *point speeds* of the vehicles were measured at the exit of the link, they would have been much higher, and match the pattern observed in measurements.

In order to illustrate the propagation of the shockwaves that arise from the incident, a cumulative flow plot is shown in figure 4. This figure was already produced in chapter 3 to show the effect of the shockwave-propagation mechanism on the functioning of queue dissipation. It is reproduced here to complement the fundamental diagrams in documenting the basic traffic performance on links in Mezzo.

In the plot each line represents the cumulative outflow on one link (segment). The distance between the lines represents the density on the links. Therefore, during the incident the lines are further apart on the links upstream, indicating the high density of the queued vehicles.

Figure 4 shows how the incident that starts on the downstream node of segment 9 at t=1200 (seconds), backs up to segment 8, 7, 6, 5, 4, and 3. At t=1500 the incident clears and the consecutive links upstream clear, each with a certain delay, due to the propagation of the start-up shockwave. At around t=1800 the queue clears completely on segment 3. In addition to the fundamental diagrams that show the capability of reproducing known traffic conditions, this figure shows that changes in traffic conditions, such as queue forming and dissipation, propagate correctly over time and space.

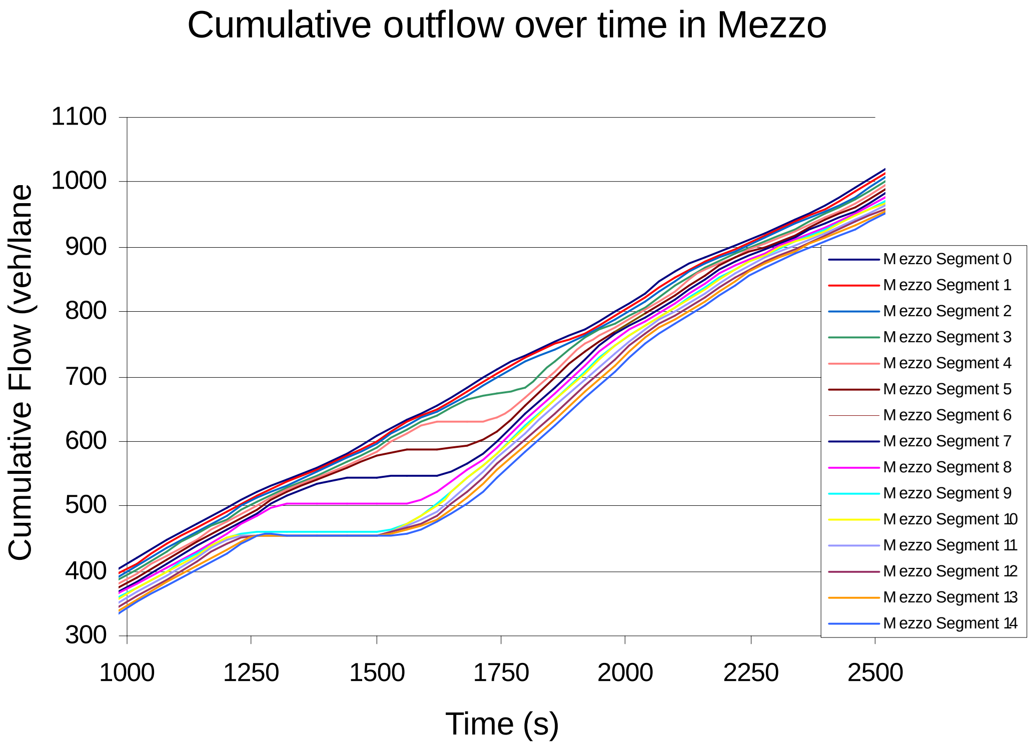


Figure 4. Cumulative flow over time for all links (segments).

5.3.2 A small freeway network

In this section the Mezzo model is applied to a small freeway network in the north of Stockholm (Järva krog). The objective is to study the propagation of flows along a number of links, as compared to data from sensor stations throughout the corridor network. The test network is shown in figure 5 and schematically in figure 6, and consists of a 2,5 km freeway stretch with one on-ramp and one off-ramp. This section is largely congested during the measurement period, and there is also congestion downstream of the modelled area that backs up into the area of interest. Moreover, the on-ramp is highly saturated and becomes congested as well during the measurement period. The weaving-in on the main road of the vehicles from the on-ramp also leads to a queue forming and backing up from the end (or nose) of the ramp, but only on the rightmost lane. The middle and left lane stay fluid. However the measurement data available consists of aggregate data over all lanes. This is a difficult facility to simulate in general, and especially so for mesoscopic models, since most of them do not model individual lanes, nor the weaving that happens where the on-ramp joins the mainline. However, I try to show that with the mechanisms in Mezzo (speed-density and queue-server) the propagation of the flows over time can be reproduced.

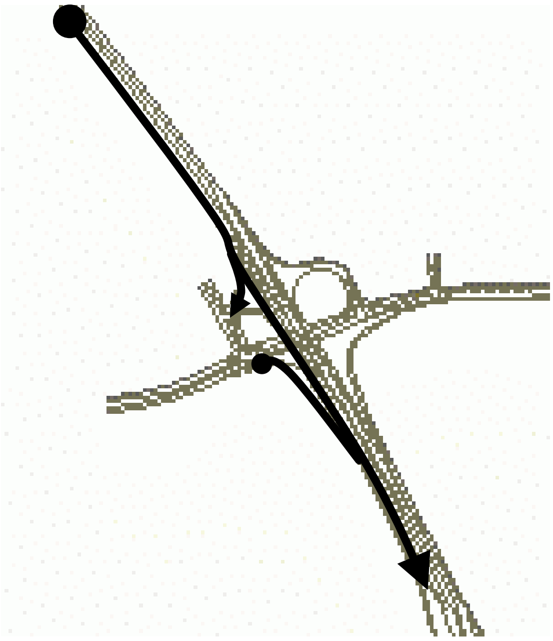


Figure 5. Järva Krog corridor on E4 North of Stockholm.

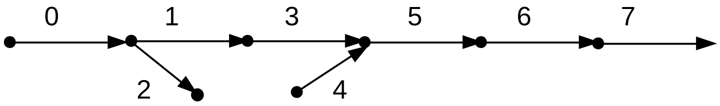


Figure 6. Schematic representation of the network.

The network consists of a freeway with two speed limits. Links 0,1 and 3 have a speed limit of 90 km/h and links 5, 6 and 7 have a limit of 70 km/h. The on- and off-ramps are modelled with a speed limit of 50 km/h. Link 0 has four lanes, followed by link 1 and 3 on the mainline, which have 3 lanes. Link 5 has four lanes (including the on-ramp merging section) and links 6 and 7 have three lanes. The off-ramp (link 2) and on-ramp (link 4) have 1 lane.

The measurement data comes from the sensors of the MCS mainline control system, and consists of 1-minute flow counts (over all lanes) and 1-minute speeds over all lanes. The speeds are a smoothed by the detector (using moving averages), which may reduce their value for comparison with the simulation data. The time of day is 5:30 – 10:15 a.m. including the morning peak build-up and decline.

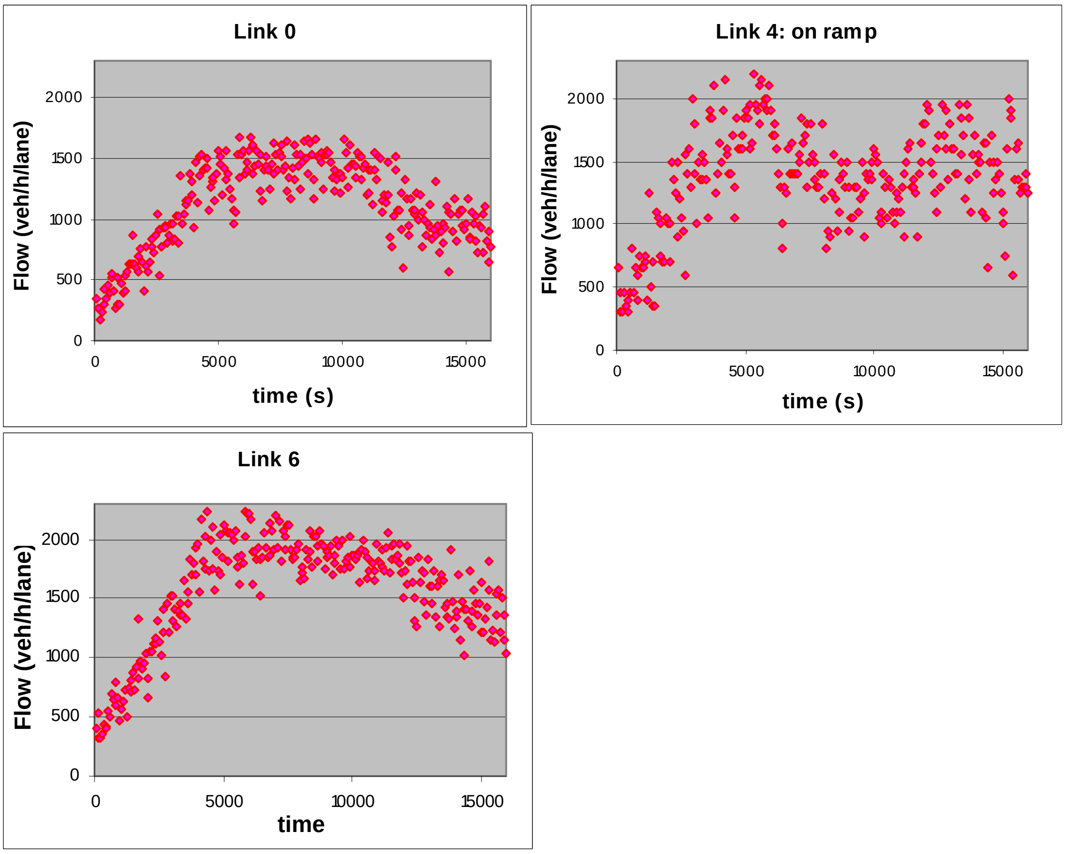


Figure 7. Flows at the origins of the network (link 0 and link 4) and the main exit (link 6).

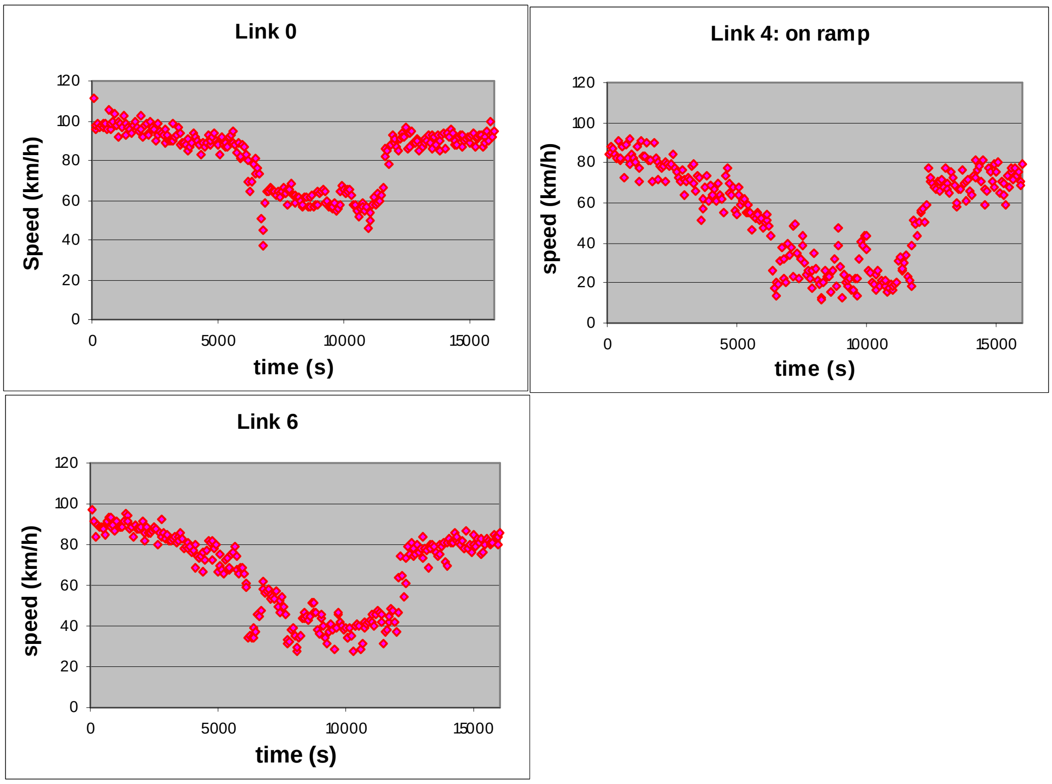


Figure 8. Speeds at origins (link 0 and 4) and main destination (link 6)

In figures 7 and 8, the main flows and speeds at the entries to the network (link 0 and on-ramp link 4) and the main exit (link 6) are shown. It is clear that around t=5000s (around 7:00 am) the flows at all sensors have reached a maximum and the speeds start to drop. Especially on the exit (link 6) and the on-ramp (link 4), the speeds drop severely. This illustrates the level of congestion in the area of study, and also indicates that the exit rate at the boundary of our network should be time-variant, to capture the effect of the congestion downstream of the limits of our network. Studying the speed/time and flow/time diagram of the sensor data at link 6, the exit capacity of the server at the exit node is limited from t=5400s (7:00 am) until t=10800s (8:30 am) (See figure 9).

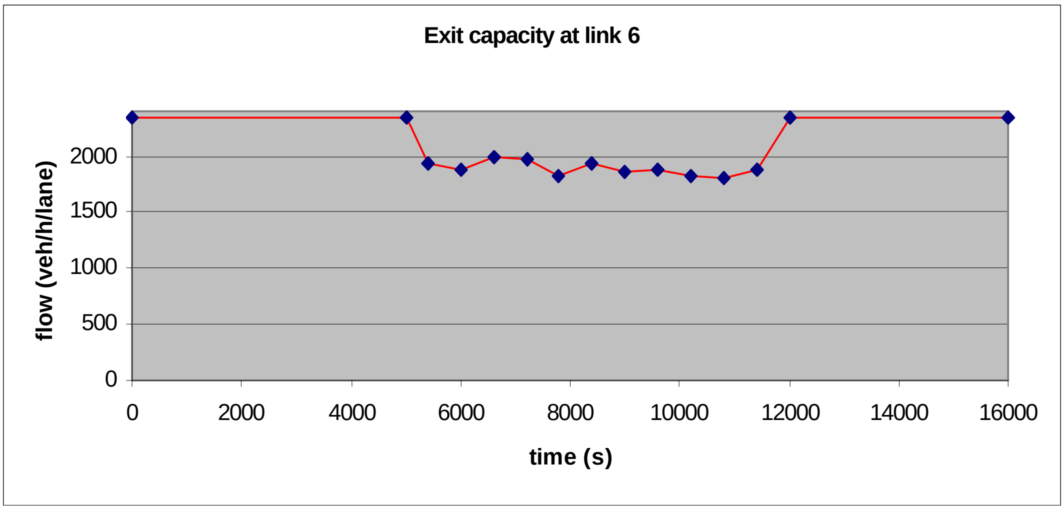


Figure 9. Time-variant exit capacity at link 6.

In Mezzo, the main mechanisms for modelling link flows are the speed/density functions and the capacity of the node servers. Using the speed and flow data, these two components were calibrated. While the speeds are smoothed, and the density is approximated (using k=Q/V, where V is the smoothed point-speed), the resulting speed/density function should be good enough for this purpose.

From chapter 3 we know that the speed/density functions used are of the form:

$$V(k)=\begin{cases} V_{free} & ,if\ k<k_{\min} \\ V_{\min}+\left(V_{free}-V_{\min}\right)\left(1-\left(\dfrac{k-k_{\min}}{k_{\max}-k_{\min}}\right)^{a}\right)^{b} & ,if\ k\in[k_{\min},k_{\max}] \\ V_{\min} & ,if\ k>k_{\max} \end{cases} \qquad (8)$$

Where,
V_{min} = minimum speed
V_{free} = free flow (maximum) speed
k_{min} = minimum density where speed is still a function of density
k_{max} = maximum density where speed is still a function of density
a, b = calibration parameters

This means that we need to set the correct values for all six parameters, based on the data from the measurements. While there are methods to calibrate these parameters automatically (for instance Box Complex method (Box, M.J. 1965)), the calibration was done manually.

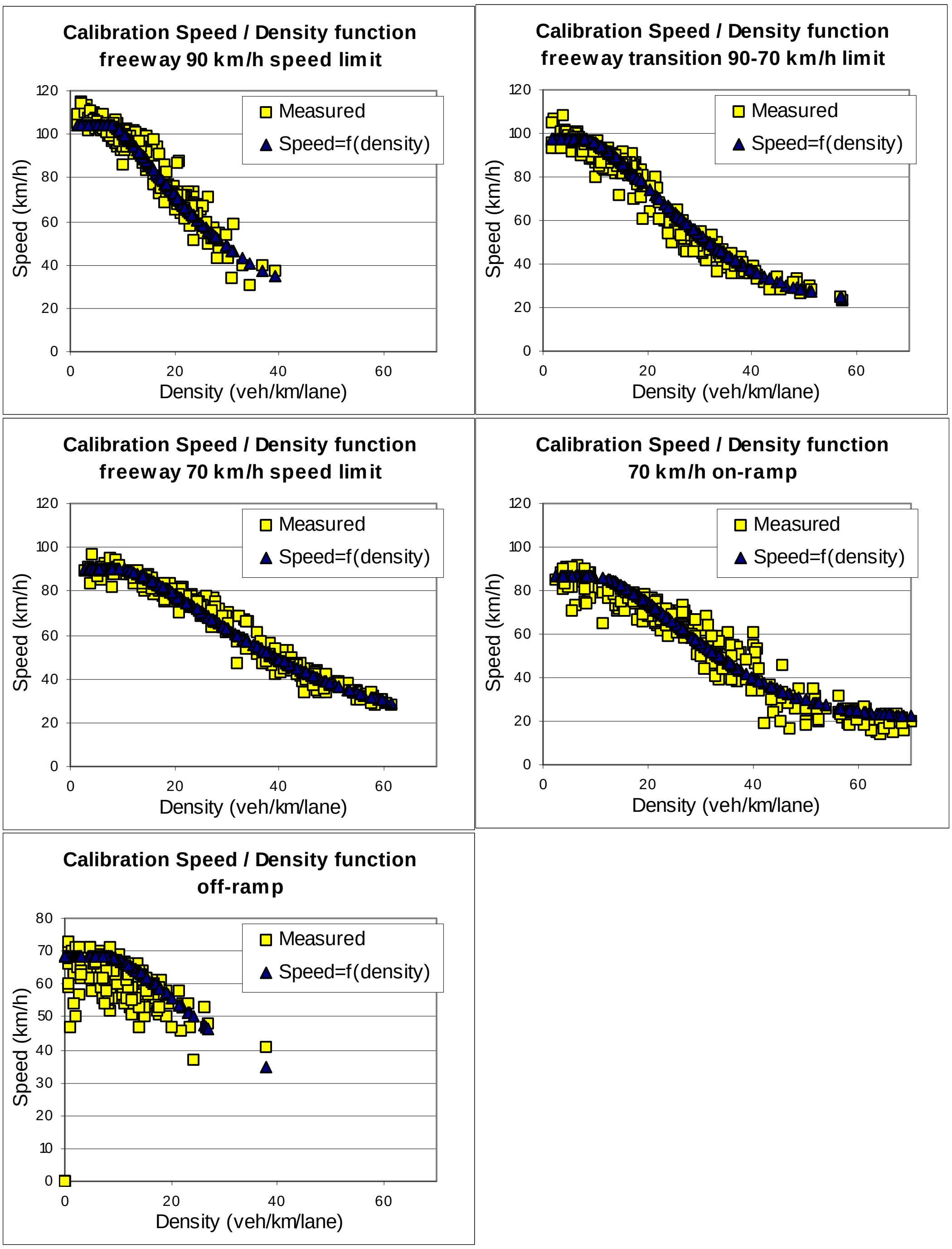
The maximum density (k_{max}) is set to 130 veh/km/lane to make sure the shockwave propagation mechanism works properly. It can be set to the maximum measured density (which is usually lower), but low values for k_{max} may lead to a rigid behaviour of the shockwave propagation. Since the density of the (downstream) segment where the queue is dissipating may be greater or equal than the k_{max}, exit speeds will be equal to V_{min}.
The free flow speed (V_{free}) is set to the highest measured speeds in the data set, and the V_{min} is usually set to 6 or 7 m/s, since it has a direct influence on the shockwave speed and exit speed at queue dissipation.

This leaves the parameters k_{min}, a and b to be calibrated. k_{min} is usually determined visually as the point in the speed/density graph where the speed starts to go down with increased density. The parameters a and b determine the curve of the speed/density function between k_{min} and k_{max}. In this case these values are determined using the resulting visual fit of the function to the data (see figure 10).

As discussed before, this area consists of a freeway with one on-ramp and one off-ramp. The speed limit on the mainline decreases from 90 km/h to 70 km/h, and the on-ramp and off-ramp have such configuration that they also have their own free-flow speeds (70 km/h on the off-ramp and 85 km/h on the on-ramp).
Therefore three speed/density functions were calibrated for the mainline: one for the 90 km/h section, one for the 70 km/h section and one for the 'transition' section (from 90 km/h to 70 km/h). For the on- and off-ramp separate functions were calibrated.

Function	V_{min}	V_{max}	k_{min}	k_{max}	a	b
90 km/h	21.6	104.4	8	130	1.3	10
90-70 km/h	21.6	97.2	8	130	1.5	11
70 km/h	21.6	90	10	130	1.4	6
On-ramp	21.6	86.4	10	130	1.6	11
Off-ramp	21.6	68.4	8	130	1.5	10

Table 1. Parameters of calibrated speed/density functions. V in km/h and k in veh/km/lane



Calibration Speed / Density function freeway 90 km/h speed limit

Calibration Speed / Density function freeway transition 90-70 km/h limit

Calibration Speed / Density function freeway 70 km/h speed limit

Calibration Speed / Density function 70 km/h on-ramp

Calibration Speed / Density function off-ramp

Figure 10. Calibrated Speed/Density functions

In table 1 the resulting parameter values are shown. The resulting functions are depicted in figure 10. The fits are visually close enough to the 'measured' data (remember that the speeds are 'smoothed' and the densities approximated using the flow and speeds over the sensors). While a goodness-of-fit test could be applied, this seems to be outside the scope of this section.

The node-servers are calibrated in a more general way, by calculating the maximum flow that is reached on the links. Using the maximum (1 min) observed output flow of around 2300 veh/h/lane, the node servers were assigned a mean service or node processing time of 1.55 seconds, and a standard deviation of 0.01 seconds, making the resulting capacity almost deterministic. While it is possible to assign different (higher) values for the standard deviation of the capacity, it is not clear what this should be, or how it can be measured.

When determining capacity from measured flows, the danger exists that a measured maximum flow rate is not due to the capacity being reached at that point, but due to a limited demand (or a bottleneck upstream). However, in our case it seems clear from the speed drops at the time of maximum flows, that the observed maxima are likely to be due to capacity being reached, rather than drops in demand.
Another problem is that while measured flows under the full range of flow conditions may give a good idea of the *mean capacity*, it is more difficult to determine the deviation, since the measurements only show the realized flows, which can depend on the deviations in capacity, but also on deviations in demand or downstream conditions.

The time-dependent OD demand matrix is determined from the flow counts at the sensors upstream of link 0, and downstream of link 7, as well as the flow counts on the on- and off-ramp. Due to the network configuration, there is only one path from each origin to each destination and there is only one OD matrix that fits the flow counts. The resolution of the OD matrix slices is 15 minutes, and the total simulated period is 4,5 hours during the morning peak (16300 seconds). Since the area is congested, it is likely that the observed flows at the on-ramp are determined not so much by the actual demand, but rather by the available capacity, which may have resulted in an OD matrix in which the demand is underestimated.

Results

Due to the fact that each link in the network has a sensor at the downstream end (except for link 5), the flow over time can be compared against data from these sensors. In order to obtain a reasonable estimate of the simulated flows, 10 replications have been made, and the averages over these runs are compared against the measurement data. There exist methods to determine the number of replications that are required for reliable forecasts with simulation models ((Law, A.M. & Kelton, W.D. 2000), (Burghout, W. 2004)), however, their application is still time-consuming and deemed outside the scope of this test.

	Link 0	Link 1	Link 2	Link 3	Link 4	Link 6
RMSE	143	48	57	49	82	46
RMSNE	12%	4%	9%	4%	6%	3%
MNE	-7%	-1%	-4%	-2%	-2%	-1%

Table 2. RMSE and RMSNE and MNE values for 15 min average flows.

In table 2 the results of the goodness-of-fit measures are shown for the 15-minute average flows on the sensor locations. The RMSE has the same unit as the original measurements: veh/h/lane, and the RMSNE and MNE show percentages, relative to the observed measurements.

The RMSNE values range from 12 % on the first link to 3 % on the last link of the network and the RMSE values range from 143 to 46 veh/h/lane, while the MNE shows a negative bias in the simulated flows ranging from –7% to –1%. It is interesting to note that the off-ramp (link 2) and the first link (link 0) in the network show the largest deviation; 9 % respectively 12 % in RMSNE, -4% and –7% in MNE. While all locations show a slight negative bias in the modelled flows, the simulated flows on link 0 are consistently lower than the observed flows, as can also be seen in figure 11. This may indicate an error in the OD matrix, but the flows on the downstream sensors are much closer, throughout the simulation period, and do not show the same under-prediction. Therefore it may be possible that the flows from this particular sensor station contain some error. Overall, it is clear that with limited modelling effort, the flow profiles over time have been reproduced well in Mezzo.

In figure 11 the flows over time are shown for all sensor locations (that correspond with the link boundaries in the modelled network). The flows are 1-minute counts, averaged over all lanes, expressed as hourly flows (veh/h/lane).

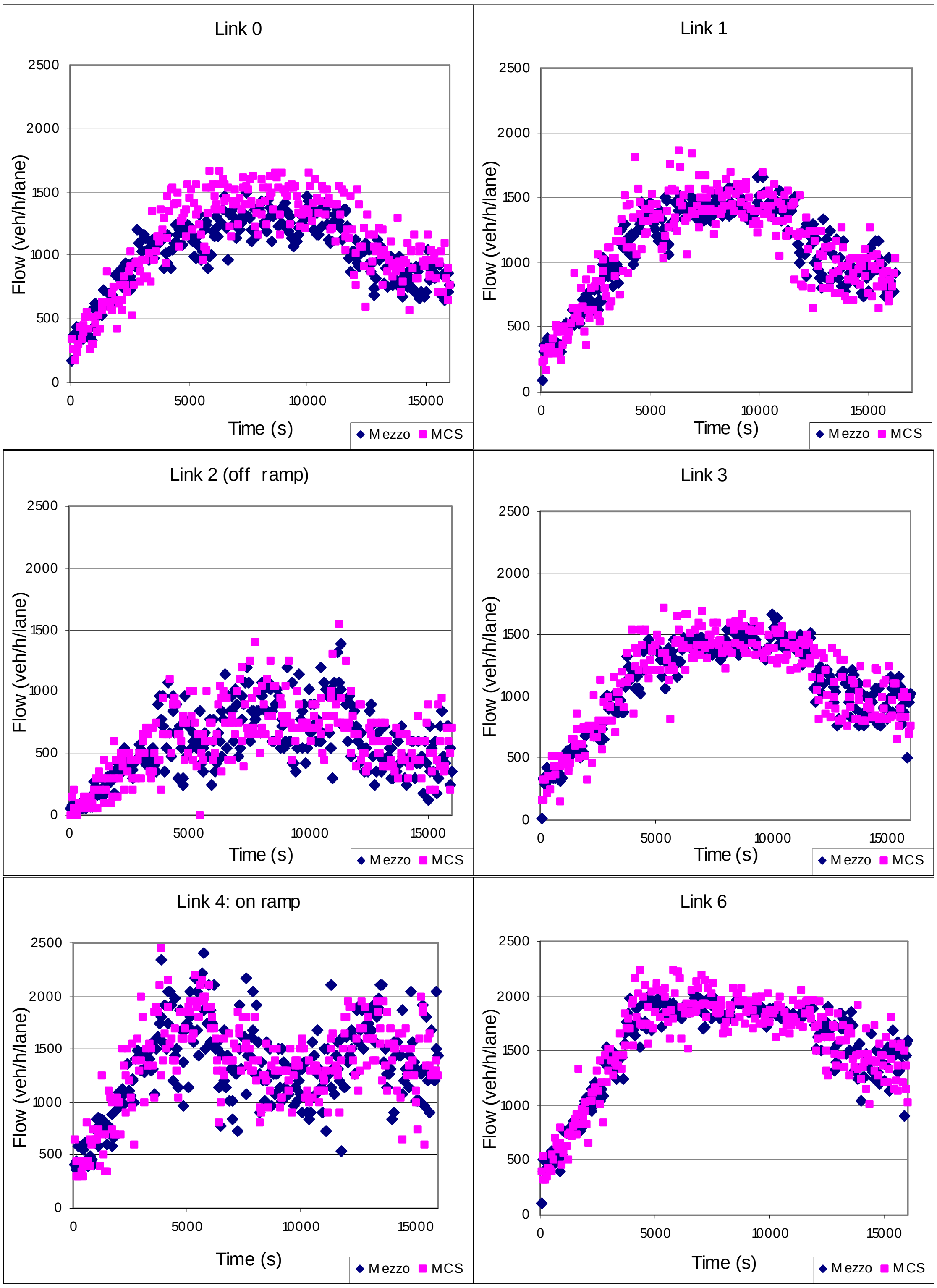


Figure 11. Flows over time for the six sensor locations and the corresponding links, for Mezzo and the (MCS detector) measurements

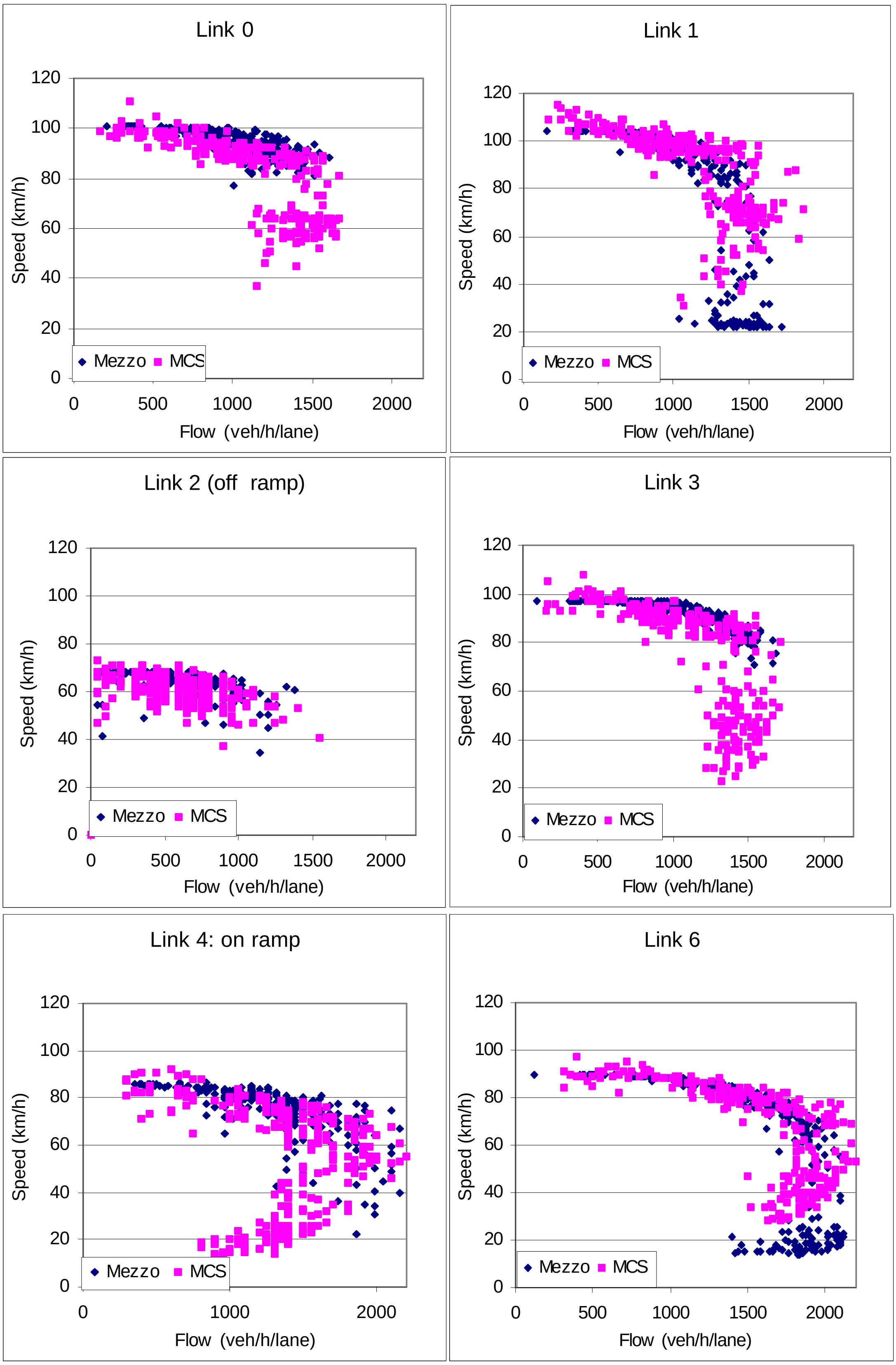


Figure 12. Speed/(in)flow diagrams.

As can be seen from figure 12, the speed/(in)flow diagrams show that there are some differences between the simulated and observed data. First of all, it should be reiterated that the speeds that are shown for the MCS detectors are smoothed point speeds, averaged over one minute, over all lanes, whereas the speeds produced by Mezzo are one minute *space mean speeds* over the whole link (for all lanes). Therefore the two may not be completely comparable.

It seems that link 0 gets more congested in reality than in the simulation, whereas link 1 is less congested in reality than in the simulation runs. This seems to indicate that the capacity of the server from link 0 to link 1 may be too low in the simulation, on the other hand, the space mean speeds from Mezzo can be expected to be higher than the point speeds from the detectors at the exit of link 0, in case the disturbance is local. In that case, the travel time on the link is not affected to a large degree, and thus the space mean speed (travel time divided by link length) will not be either.

Again, on link 3 the simulation is less congested than in reality, possibly due to the high congestion that occurs on the upstream link 1. While the off-ramp (link 2) shows good correspondence with the observations, the on-ramp shows less congestion than in reality. This is may be the result of the fact that (at this moment) the weaving-in of traffic from the on-ramp into the mainline traffic is not modelled, and thus the outflow at high saturation on the mainline will be too high. Implementation of this relationship should alleviate the problem. Another (simpler) solution could be to vary the server rate for the turning movement using measured flows for different periods of the day.

Finally, link 6 shows a fairly good correspondence of the speed/flow profile, although the simulation shows more congestion than reality, possibly due to the too high inflow of the upstream on-ramp. In addition, the congestion that is backing up from downstream (and outside the boundary of our network) is emulated by varying server rates at the exit, corresponding to the measured flows over a detector downstream. This may also be the reason for lower space mean speeds than the spot speeds measured at the detector. In fact, if we compare the speed/flow data over this detector with the speed/flow on link 5 (figure 13), which is upstream of the on-ramp, we see that apart from the higher free-flow speeds (since this is the area where the 90 km/h speed limit changes into 70 km/h), the fit with the speed/flow data from the sensors is better.

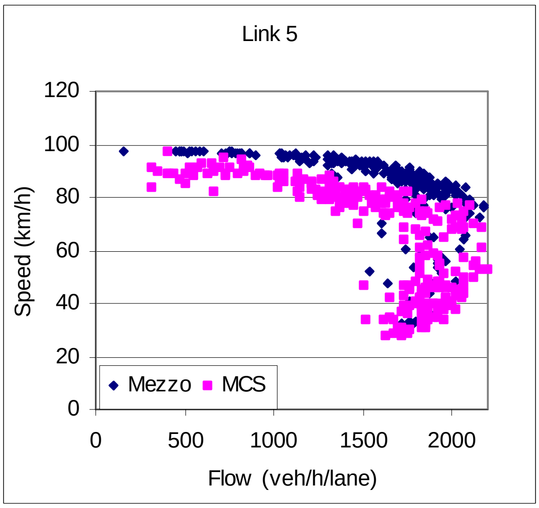


Figure 13. Speed / flow on link 5 compared with MCS detector at link 6.

5.3.3 Brunnsviken network

In order to test the behaviour of Mezzo on a network, the model was applied to a 5x5 km network in the north of Stockholm. It is usually called the "Brunnsviken" network, and it has been a standard network for the evaluation of a number of simulation models. Figure 14 gives a general overview of the network. The E4 (Uppsalavägen) on the left side is the main freeway into Stockholm and has a speed limit of 90 km/h in the north and 70 km/h in the middle part. The area in the south is the northern part of urban Stockholm and has a speed limit of 50 km/h. The E20 is the other main freeway leading into Stockholm from the North. The period of study is 6:45 – 9:00 a.m., during which the area is usually congested or close to congestion. This means that while the number of routes for any OD pair is usually just 1 or 2, the route choice between them is critical since deviations may result in extreme congestion and gridlock.

In an earlier project, which is published in (Toledo, T. et al. 2003), MITSIMLab (Mitsim) was calibrated and validated for this network, using floating car travel time measurements, and extensive flow measurements. This data is used here to evaluate the performance of Mezzo on the same network.

The OD demand matrix is taken from the one estimated for the MITSIMLab test (see (Toledo, T. et al. 2003) for details on the estimation). The measurement data consists of 15 minute flow counts and floating car travel time measurements from two days in May 2000. Here, the 15 min flow data for twelve locations (See figure 14) for one day are used.

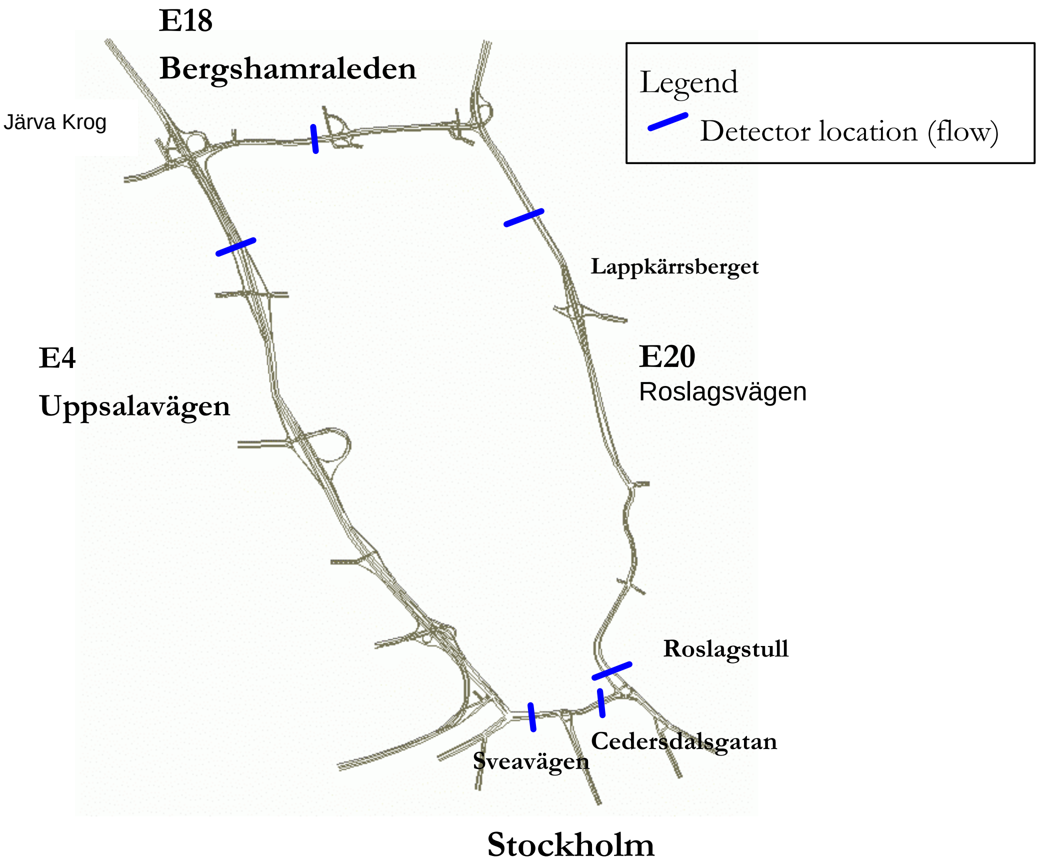


Figure 14. Brunnsviken Network

The network consists of both freeway sections as well as signal-controlled intersections. As described in chapter 3, in the current implementation of Mezzo the operation of signal control plans is not yet included. Instead, the ratio of effective green time over cycle time was used for the intersections. For the speed/density functions, the ones calibrated for the Järva Krog scenario (section 5.3.2) were used, as it makes part of the network (north-west). The speed density functions for the 50 km/h part were adapted from the 70 km/h ones, since there was no speed data available for those links.

In the Brunnsviken Network, the E18 Bergshamraleden (East-West in the north of the network) and the E20 Roslagsvägen contain dedicated bus-lanes, on which only buses and taxis are allowed to drive. In Mezzo lanes are not represented individually, and there does not (yet) exist a mechanism to decide which vehicle class is allowed on which link. Otherwise it would have been possible to add a parallel link for each bus lane. The vehicle classes themselves, with their own properties are already part of Mezzo. In the future this selective entrance to links should be added to be able to simulate such facilities properly. In this case, the bus lanes were simply ignored. Since the amount of buses (and taxis) is not known for each OD pair, it was difficult to subtract them from the flows. It would be possible to decrease the amount of traffic with the percentage buses and taxis (around 6% respectively 10% of the total fleet) but that would skew the assignment for those parts where no bus lanes are present. So it was decided to simply leave the OD matrix as it is.

Route and Travel Time generation

Using the algorithm described in chapter 3, the historical travel times and path sets are generated using alternative iterations of simulation and shortest path search. The new paths that are found by the shortest path algorithm are added to the set of known paths. The travel times for each simulation iteration are a weighted average of the input travel times and the output travel times of the previous iteration. While there is no theoretical guarantee that this process should converge (to a certain set of routes and time-dependent travel times), in practice this occurs. In this section the results of iteration with the Brunnsviken network are shown.

Because this is a small network, where OD pairs have only one or two viable routes, the iterations stabilise fairly quickly and it was observed that after three iterations there are hardly any new routes found.[10] In addition, due to the special topology of the network, and the high level of congestion, it has proved to be important that *only* relevant routes are generated, since any 'extra' route changes the assignment, and may have large effects on the network performance. Therefore the route set is manually checked after each iteration, and this showed that 99% of all relevant routes (all but one) were generated in the first two iterations. Moreover, whereas there were only 4% irrelevant routes in the second iteration, 90% of the new routes found in the third iteration were not relevant. Therefore it seems important that:

1. The simulation loop, Loop 2 (which generates the travel times) is iterated after each route finding iteration, to create stable travel times that reflect the network performance associated with the route set.
2. A check needs to be performed after each route-finding iteration, so that these iterations can be stopped when no more relevant routes are found. In our case this check was manual, using knowledge of the network, but maybe an automatic check would be possible, such as only allowing routes that are within a certain percentage (distance or time) of the shortest free flow path.

[10] Due to the separation of the simulation and the shortest path search, routes are found for ALL OD pairs, even those that have no traffic. Naturally, this does not affect the simulation results.

Therefore the iterations for the generation of the link travel times are started using the pruned route set. Due to the fact that there are only a small number of OD pairs with two routes, the travel times are expected to converge quickly.

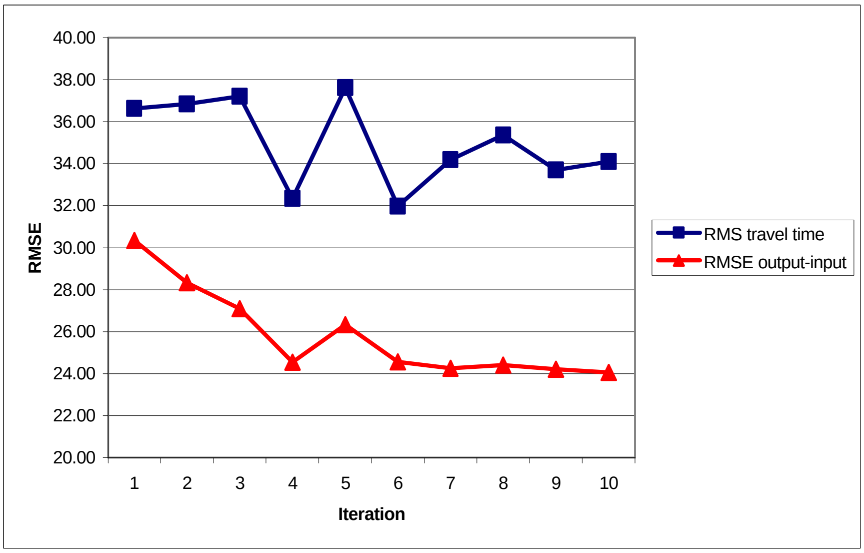


Figure 15. Performance measures for travel time generation

In figure 15 we see a graph indicating two quantities that give an idea of the change in travel times over the iterations. The *RMS travel time* indicates the root mean square of the travel times for al links and all periods. It indicates a sort of "average" link travel time for all links and periods:

$$RMS\,travel\,time = \sqrt{\frac{1}{NM}\sum_{j=1}^{M}\sum_{i=1}^{N}\left(tt_{i,j}\right)^2} \qquad (9)$$

Where

$tt_{i,j}$ = travel time for link j and period i (in seconds)

N = number of time periods

M = number of links

The second quantity, *RMSE output-input* shows the RMSE of the difference between input and output link travel times. This indicates the gap between the expected travel times (input travel times) on which the drivers base their choices, and the resulting output travel times. Large RMSE values mean that there are large (absolute) differences between input and output link travel times:

$$RMSE\,output - input = \sqrt{\tfrac{1}{NM}\sum_{j=1}^{M}\sum_{i=1}^{N}\left(tt_{i,j}^{output} - tt_{i,j}^{input}\right)^2} \qquad (10)$$

Where

$tt_{i,j}^{input}$ = input travel time for link j and period i (in seconds)

$tt_{i,j}^{output}$ = output travel time for link j and period i

N = number of time periods

M = number of links

The RMS travel time gives an idea of the 'system performance' of the network assignment, whereas the RMSE output-input indicates the distance from a dynamic user equilibrium, since in that case the difference between the expected (input) travel times and the output travel times should be zero.

As mentioned in chapter 3, the input travel times for iteration n are calculated from the input and output travel times of iteration n-1:

$$tt_{i,j}^{input}(n) = a * tt_{i,j}^{output}(n-1) + (1-a) * tt_{i,j}^{input}(n-1) \qquad (11)$$

Where

$tt_{i,j}^{input}(n)$ = input travel time for link j, period i, and iteration n

$tt_{i,j}^{output}(n)$ = output travel time for link j, period i, and iteration n

a = travel time smoothing factor.

The averaging increases stability of the iteration process and should diminish oscillations in travel times from one iteration to another. The value chosen for a is 0.2 . The iterations start with the free-flow travel times for each link, and it can be observed from figure 15 that the resulting RMS travel times are not that bad. The gap between the expected travel times and the experienced ones (RMSE output-input) is not exceptionally large either. Over the 10 iterations there is a clear downward trend in the RMSE output-input, indicating that the travel times are moving towards a kind of equilibrium. The RMS travel time seems to decrease as well, but the oscillations are too big to be able to observe a clear downward trend. A probable cause for the trend not being more pronounced is the fact that in this network there are only a few OD pairs with more than one route, and that the network is so close to saturation, that changes in output travel times are not necessarily consequences of the input travel times on which the route choices are made, but rather influenced by stochastic factors such as arrival processes and node server processes. This brings us to another important point that can improve the travel time and route generation scheme:

3. It is important to replicate the simulations *within* the iterations to obtain a good estimate of the network performance that is associated with the path set and travel times for that iteration.

After 10 iterations, the RMSE input-output does not decrease any further and neither does is the RMS travel time.

In figure 16 (below) we can observe two typical routes for an OD pair from the North-West to the South-East, the route via the E4 and the route via the E20 (dashed). Usually the E4 route is faster, and the majority of the drivers take this route.

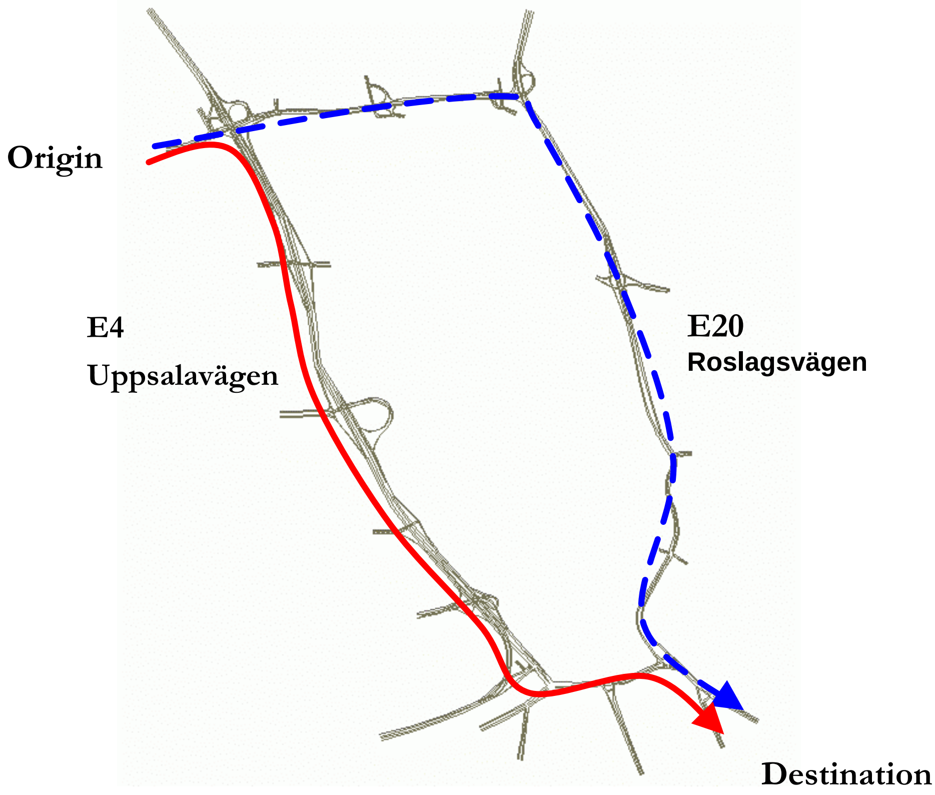


Figure 16. Two routes from Origin (northwest) to Destination (southeast).

In figure 17 the travel times are shown for vehicles using this OD pair, for run 1 and run 10. The travel times along the shortest path (via the E4) stay around 350 seconds. However, a small percentage of the vehicles take the alternative route via the E20. Their travel times are about the same for the first periods, but from t=3000s onwards, their travel times are substantially larger than those for the normal route. After 10 iterations, the difference between the two paths is almost zero for the first periods and in the later periods it is smaller than in the first run. In the case of a larger network, where there are more alternatives per OD pair, and less congestion, one would expect the travel time differences between the paths to diminish and virtually disappear in the course of iterations.

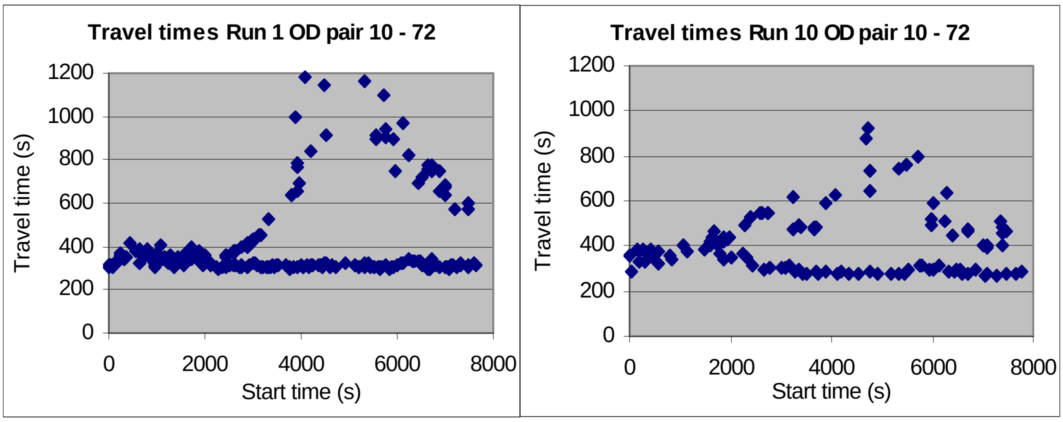


Figure 17. Vehicle Travel times by start time for one OD pair.

Comparison of flows

In this section the flows produced by the Mezzo model are compared against the measured flows. These flows are a result of both capacity in the network, and the route choice of the drivers. This means that it is important to have a good route choice model, a good choice set (the alternative routes) and good historical travel times on which the drivers base their routes. Thus the procedure discussed previously to generate this choice set and travel times is as essential as the correct modelling of network capacity in obtaining the correct flows.

The first period (06:45-7:00) generally has slightly lower flows, since this period was used as a “warming-up” period, in which the simulation network is filling up (from the empty start state).

	E4		E20 Roslagst.		E20 Lappkärrsb.		E18		Cedersdals- gatan		Sveavägen	
	SB	NB	SB	NB	SB	NB	EB	WB	EB	WB	WB	EB
RMSE	67	83	134	30	97	30	80	140	49	66	75	56
RMSNE	5%	8%	34%	9%	13%	9%	16%	21%	13%	12%	16%	8%
MNE	0%	2%	26%	6%	9%	6%	12%	17%	-1%	-1%	-5%	3%
Theil U	0.024	0.034	0.140	0.039	0.054	0.039	0.066	0.091	0.067	0.057	0.081	0.028

Table 3. RMSE, RMSNE, MNE and Theil U errors for all flow locations.

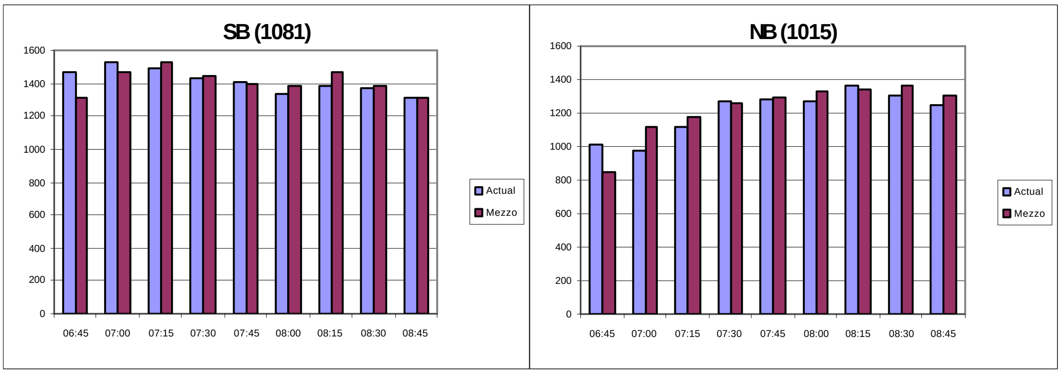


Figure 18. Flows on E4 Southbound (left) and Northbound (right)

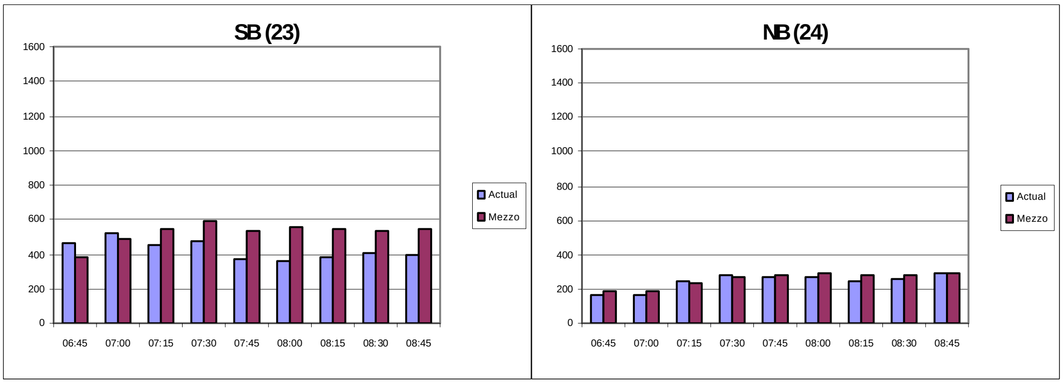


Figure 19. Flows on E20 Southbound (left) and Northbound (right) near Roslagstull

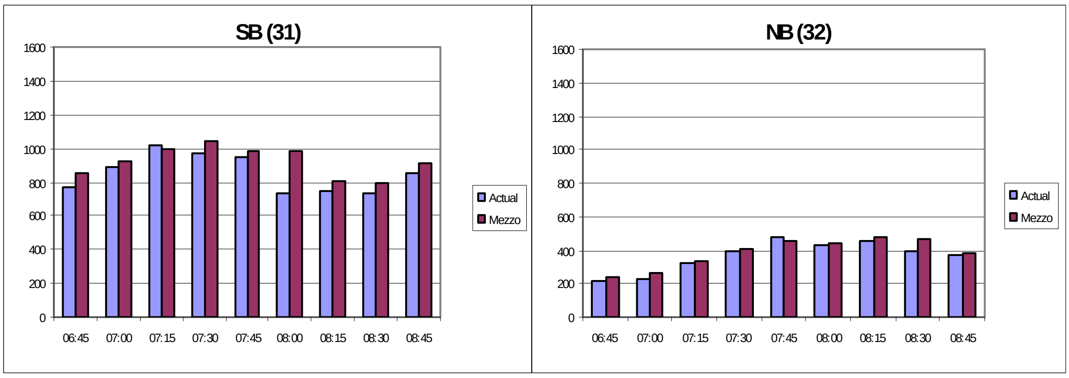


Figure 20. Flows on E20 Southbound (left) and Northbound (right) near Lappkärrsberget

In Figure 18 the flows on the E4 in Southbound and Northbound direction are shown. It is clear that the simulation matches the real flows closely. Table 3 shows a MNE (or bias) of 0% and 2% for the southbound and northbound directions, respectively. This means that overall, the model neither over, nor underpredicts the flows, on this location. However, as can be seen from the RMSNE of 5% and 8%, the individual periods may show a (small) over-/underprediction at times. The RMSE of 67 and 83 (vehicles per 15 min. period) shows that also the absolute errors are small.

On the other hand, the flows on the southbound direction on the E20 are higher than the measured flows, both in the south, near Roslagstull (figure 19) and in the north, near Lappkärrsberget (figure 20). This is possibly due to the simplification of both the traffic signal control at two intersections between Lappkärrsberget and Roslagstull, and the operation of the roundabout in the southeast corner of the network, which is approximated using estimated saturation flows from Mitsim. The errors in table 3 confirm this, showing a MNE (bias) of 26% for the southbound direction near Roslagstull, and 9% near Lappkärrsberget, and 6% in the northbound direction at both locations.

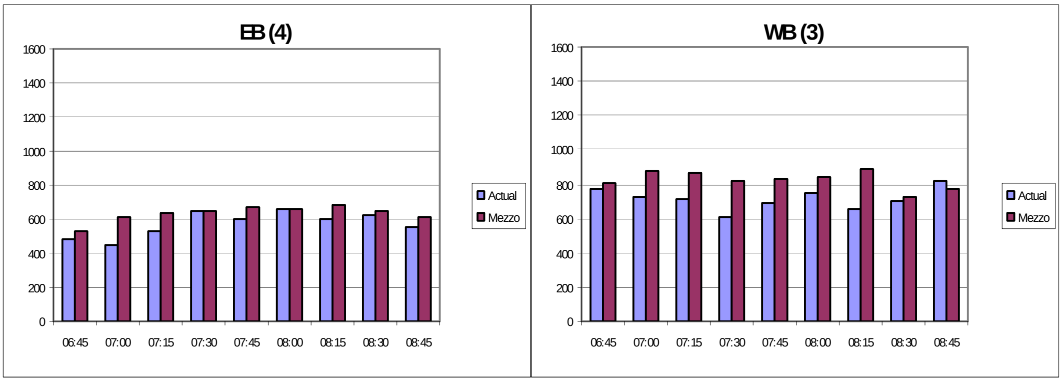


Figure 21. Flows on E18 Bergshamraleden Eastbound (left) and Westbound (right)

On the E18 freeway (figure 21) the eastbound traffic seems to be fairly close to the measured flows, but the flows in westbound direction are higher, possibly indicating that a larger fraction of traffic than in reality takes the E18-E4 route into Stockholm, instead of the E20 Southbound. The errors in table 3 indicate that both directions have a positive bias (12% and 17% respectively), indicating that in the model, a larger share of traffic from the two major origins in the north (E4 and E20) seems to take the alternative route into Stockholm, using the E18. The positive bias on the E18 westbound is likely due to the fact that Mezzo only approximates[11] the functioning of the three traffic signal controls that regulate the traffic on this link and especially the access to the E4 Southbound into Stockholm. In reality these traffic signals cause a queue to build up, which is not the case in the simulation. Therefore, in reality, the congestion results in a lower throughput than simulated during the congested periods, and therefore the flows would be higher than the simulation in the two periods after (8:30-9:00).

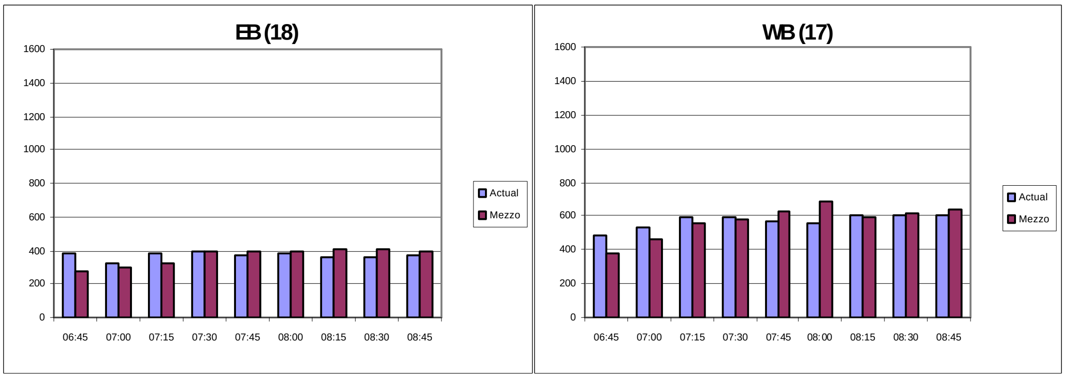


Figure 22. Flows on the Cedersdalsgatan Eastbound (left) and Westbound (right)

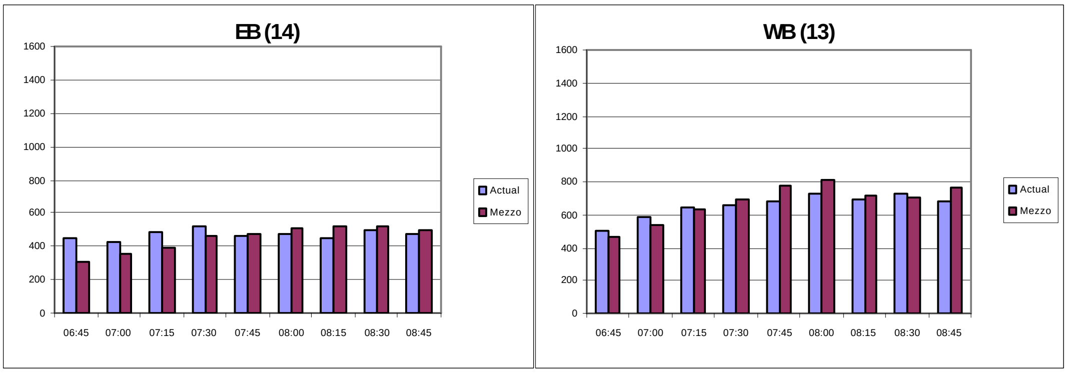


Figure 23. Flows on the Sveavägen Eastbound (left) and Westbound (right)

In figure 22 and 23, the flows on the 50 km/h urban part in the south of the network are shown. On the eastbound direction the flows are a bit on the low side, especially during the first three quarters. The traffic in westbound direction seems to match the measured flows throughout the simulation period, being slightly higher than measured for the periods 7:45-8:15. This area is regulated by the large roundabout at Roslagstull and the traffic signals at Sveaplan and Norrtull. The high flows in the westbound direction may indicate that the capacity of the Mezzo approximation of the operation of the roundabout at Roslagstull is too high, as was

[11] However, as mentioned in chapter 3, Mezzo can easily be extended to include the operation of traffic signal controls.

already seen in the measurements at Roslagstull itself. The errors in table 3 show low bias errors (MNE) for these locations, but the RMSNE (ranging from 8% - 16%) shows that there are some deviations from the measurements for the individual time periods.

The Theil U values in the bottom row of table 3 show that all of the locations show a good approximation of the actual flows (U=1 means completely different, U=0 means a perfect fit), where the best value is shown for E4 Southbound (0.024) and the worst for Roslagsvägen Southbound (0.14).

5.3.4 Conclusion

In the previous sections the Mezzo model was tested and evaluated on three networks, of increasing complexity and realism. While the model is still under development, and important parts such as traffic control are yet to be implemented, the general behaviour of the modelling of traffic flows seems to be satisfactory. The fundamental diagrams correspond to what one would expect (See (May, A.D. 1990),(HCM 2000)). The test on the congested freeway corridor with an on- and off-ramp (Järva Krog) showed that the flows are reproduced well, even though this test area contains elements (weaving under congested traffic on both mainline and ramp) that would normally require microscopic simulation to reproduce. The third test case, the Brunnsviken network, is definitely a difficult one, due to the high level of congestion and the mix of urban areas containing signal controlled intersections and freeway sections with congested weaving, as well as a roundabout in the southeast corner. In addition only limited effort has gone into calibration of the capacity of turning movements, and the traffic signals operation was modelled in a simplified way.

In spite of these challenges, the flow counts on most sections are approximated closely by the model, but it seems likely that the application of Mezzo on this network could benefit from careful modelling of the traffic control.

The generation of routes and travel times was also tested on this third network. Due to the limited number of viable routes for each OD pair (one or two), the route set was checked manually, and spurious routes were removed. The route generation had found all relevant paths in two iterations. The generation of the travel time database, on which the route choice is based, converged within a few iterations to a stable level (although not equilibrium).

However, a number of lessons were learned regarding the route set and travel time generation. Firstly, it is important to replicate the simulation runs within each iteration, to create a good estimate of the (average) network performance to be associated with the current route set and (input) link travel times. Secondly, it would be wise to have iterations exclusively for the generation of travel times after each 'route-finding' iteration. This should provide better estimates of link travel times associated with the current route set, and should help to prevent the generation of spurious routes. Thirdly, a separate manual or automatic check could be applied on the route generation, so that only routes that are within a certain percentage (distance or travel time) of the shortest route are taken into consideration.

5.4 Evaluation of MiMe

In this section the MiMe hybrid model is evaluated. The Mezzo model was evaluated in the previous section, and Mitsim has been previously validated (Toledo, T. et al. 2003). In this section the focus will be on the performance of the boundaries and virtual links, which are the new components in addition to the Mitsim and Mezzo models.

In the first subsection the inter-model boundaries are tested for consistency. As we saw from the literature review in Chapter 2, and the discussions about the integration framework in Chapter 4, these boundaries are critical in the operation of the hybrid model. The mechanisms to ensure consistency are described in Chapter 4, and the behaviour of the vehicle-loading algorithm (speed generation) is already evaluated there. Here the propagation of variations in flows, speeds and densities across the boundaries is evaluated.

In the second subsection the MiMe model is applied to the Brunnsviken network, and simulated flows are compared with measurements.

5.4.1 Boundary consistency

In order to test the propagation of traffic dynamics across the boundaries between micro and meso and vice versa, a small test network is created that consists of a 5km straight two- lane road consisting of 10 consecutive segments of 500 meters each. (See figure 24)

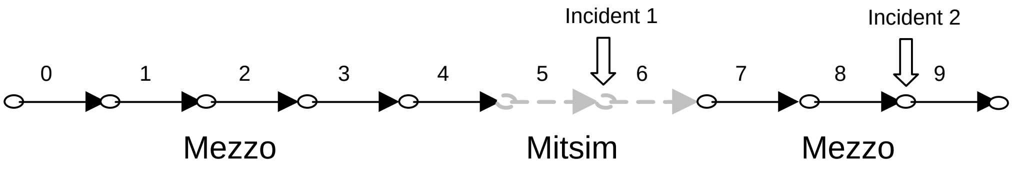


Figure 24. Test network for testing boundary consistency.

The first five segments are in Mezzo, followed by two segments in Mitsim, and the last three segments are in Mezzo again. The free-flow speed is 100 km/h and the speed limit is 90 km/h. The demand is constant at 3000 veh/h (1500 veh/h/lane) for the simulation time of 1 hour (3600 seconds).

Using this network, two scenarios are evaluated. In the first scenario an incident (incident 1 in figure 24) is created at the entry of (Mitsim) segment 6. The incident starts at t=20 minutes (1200 seconds) and blocks both lanes completely for five minutes (300 seconds) until it is cleared at t=25 minutes (1500 seconds). The queue build-up and dissipation over the Mezzo to Mitsim boundary will then be studied in detail. The cumulative flow over time, flow over time, speed over time, and density over time are evaluated for the Mitsim and Mezzo segments upstream of the incident, and the speed/density profile of Mitsim segment 5 is compared to Mezzo segment 4.

The second scenario studies the effect of queue propagation and dissipation over the Mitsim to Mezzo boundary. The incident (incident 2 in figure 24) is located at the node between segments 8 and 9 and like incident 1 in scenario 1, it starts at t=20 minutes, lasts for 5 minutes and blocks both lanes completely. As in the case of scenario 1, the section will evaluate the cumulative flow over time, flow over time, speed over time, and density over time for the Mezzo and Mitsim segments upstream of the incident.

Calibration of Mezzo Speed/Density function and Turning Server

In order to ensure that the modelled traffic performance in Mezzo is comparable to that of Mitsim, we first need to calibrate the server capacity of the turning-movements at the nodes, and the parameters of the speed/density function in Mezzo to the performance of Mitsim

In this network, there is only one turning movement type, which is a straight movement. The flow over time is measured in Mitsim, given demands increasing from 3000 to 5000 veh/h (1500-2500 veh/h/lane).

Since it is a stochastic simulation, the results may vary from one simulation run to another. Therefore the simulation is repeated a (arbitrary) number of 10 times (see (Toledo, T. et al. 2003) or (Burghout, W. 2004) for a good way to determine the required number of replications in stochastic simulation). By identifying the 15-minute period with maximum (output) flow, and taking the mean and standard deviation over the 10 simulation runs, the capacity of the turning movement servers in Mezzo could be determined. The values used were a mean time headway of 1.44 seconds / lane and a standard deviation of 0.1 seconds. This corresponds to a mean capacity of 2500 veh/h/lane and a standard deviation of 360 veh/h/lane.

We then need to calculate the parameters of the Mezzo speed/density function so that it matches the speed/density measurements from the micro model. Again 10 replications are made, but to obtain speed/density values for high densities, a bottleneck is created at the downstream end of the second link in the network. The function parameters were fitted manually to the data obtained from Mitsim obtained. In figure 25 the match with the resulting speed/density function is shown, along with the fit of data from Mezzo simulation runs to the speed/density data obtained from Mitsim, using the calibrated functions. In table 4 the parameters of the resulting speed/density function are presented. Note that for densities larger than 60 veh/km/lane, the calibrated function deviates from the measurements. This is due to the relatively high Vmin (6 m/s), which is needed to provide high enough minimum speeds for queue dissipation. In addition, as the right graph in figure 25 shows, the queue-server and shockwave propagation mechanisms provide the correct behaviour for these levels of density, so that the overall fit of the two models' speed/density curves is quite close.

Vmax (m/s)	Vmin (m/s)	Kmax (veh/km/lane)	Kmin (veh/km/lane)	a	b
23	6	130	13	2	8

Table 4. Speed/Density function parameters.

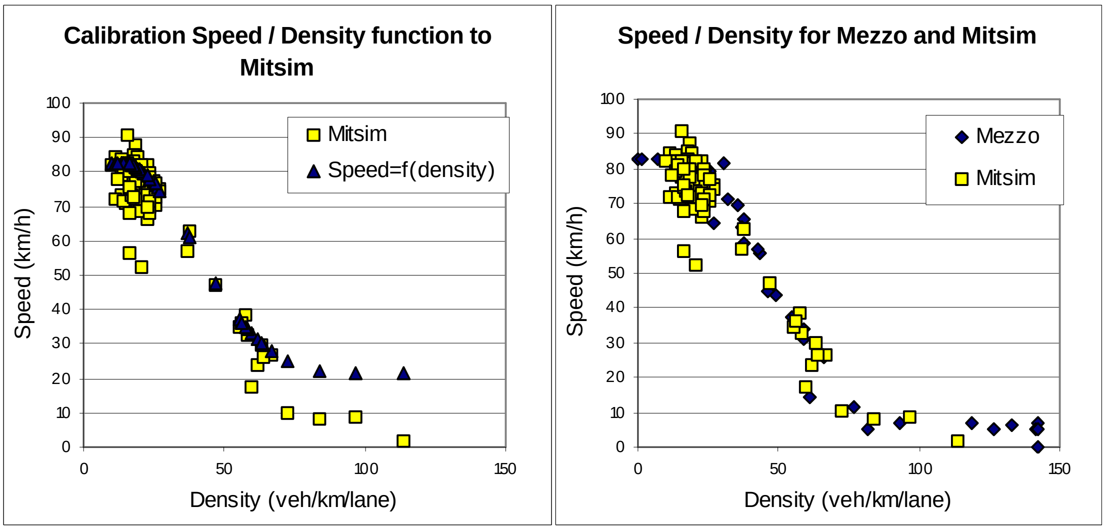


Figure 25. Calibration (left) of Mezzo speed-density function from Mitsim data, and results (right) from Mitsim and Mezzo

While in this case manual calibration of the parameters using the graphs for a measure of fit may be sufficient, there exist many tools that help both the calibration process and the test of the resulting fit. It is advisable that for larger calibration efforts, such tools are used.

Scenario 1. Meso to micro

In this scenario the boundary from the meso model to the micro model is evaluated. As we saw in chapter 4, there are a number of mechanisms that should provide the smooth coupling of the two models, such as careful vehicle loading (initial speed generation), the vehicle representation in Mezzo, the provision of queue servers for each outgoing lane to provide a good arrival at the entrance to micro, and the communication which transfers vehicles each micro simulation step (0.1 seconds). In this first scenario the incident is located inside Mitsim and the queue grows past the meso-micro boundary. Then the incident is resolved and the queue starts to dissipate. The first results are the cumulative flows in figure 26 . The cumulative outflow for a segment indicates the total number of vehicles that have left that segment, over time, since the start of the simulation. Each line shows one segment, and the distance between two segments shows the density over time.

At t=1200s an incident blocks the exit of segment 5, which is in Mitsim. We can see from figure 26 how the cumulative flow line that belongs to Mitsim is horizontal from t=1200s, until the clearance of the incident at t=1500s. This means no vehicles exit the segment during that period. Around 150 seconds later we observe the same behaviour for the upstream Mezzo segment 4. This means it took 150 seconds for the queue to reach the exit of segment 4, and it takes another 150 seconds to reach segment 3, etc.

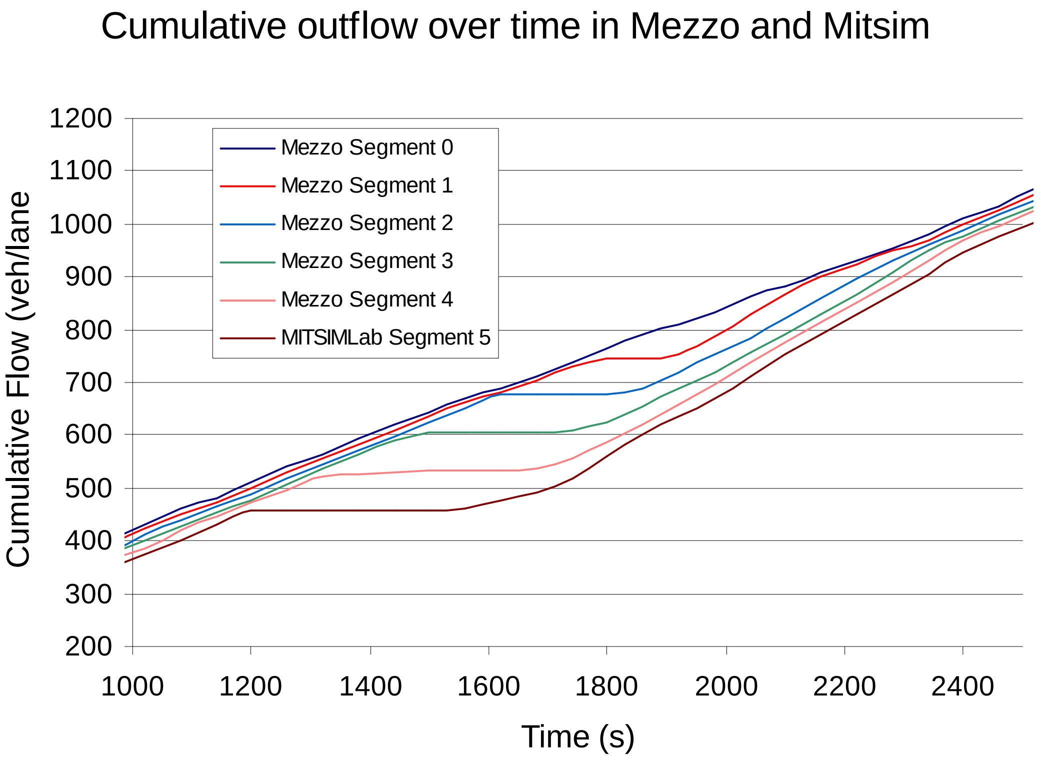


Figure 26. Cumulative flows

When the incident clears, it takes around 130 seconds before the queue front has reached the exit of segment 4 and traffic starts to flow there. From segment 4 to segment 3 it takes 110 seconds and we see how it propagates backward until the queue front meets the queue back in segment 1 at t=1920s, after which a small forward shockwave forms, which disappears in segment 2.

When we take a closer look at the density over time (figure 27), for the five segments of interest, we observe that indeed the density increases in segment 5 (MITSIM), after t=1200s. At t=1500 the queue reaches the boundary with the mezzo segment 4, and at the same time the incident is cleared, causing the queue on segment 5 to dissipate. However, the queue back is still blocked, and this shows in the fact that the density remains high for segment 4, and the subsequent segments start to get blocked. As in the cumulative flow plot, it can be observed that both queue forming and dissipation function properly over the boundaries. However, there are differences between the Mitsim segment and the Mezzo segments. First of all, the queue dissipation shockwave propagates faster through Mezzo than it does through Mitsim, which is shown in the fact that the time between the decrease in density in segment 5 (MITSIM) and segment 4 (Mezzo) is larger than the time between the decrease in density for the subsequent Mezzo segments. This means that the shockwave and exit speeds in the Mezzo segments are larger than in Mitsim, which is probably caused by the fact that Mezzo uses a simplification of the acceleration and reaction delay that is the underlying cause of the start-up shockwave.

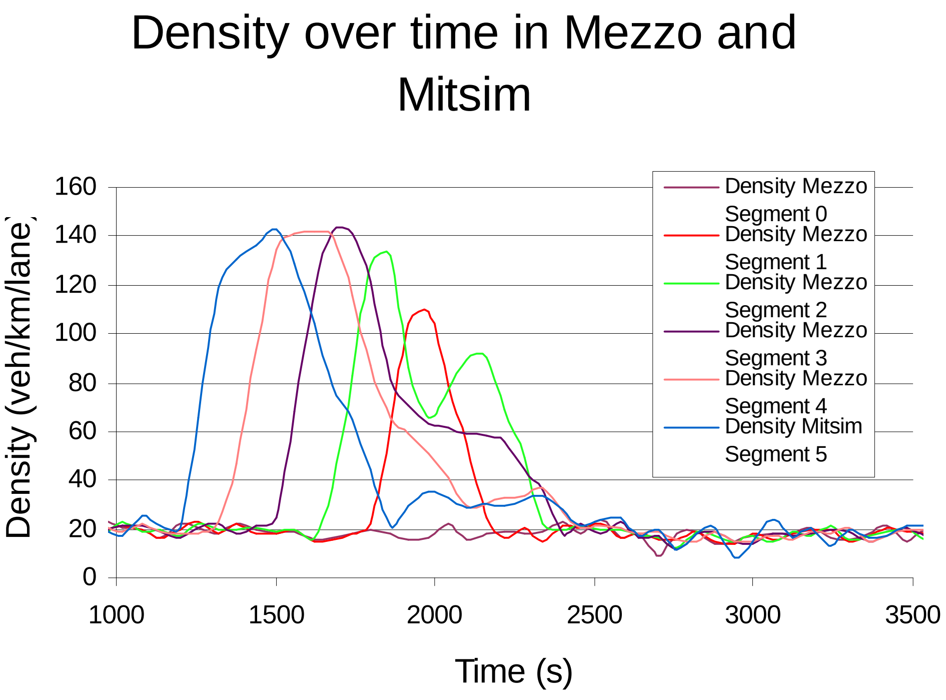


Figure 27. Density over time.

In addition to this observation, it can also be noted that when the queue front reaches the queue back on the last congested segment (segment 1), this results in an increase in the density in segment 2. This may be a result of the simplified mechanism in Mezzo, in that it does not take into account the vehicles that drive up to a queue back and start to brake, but never come to a complete stop. Because of this simplification, the outflow of this segment is rather large, resulting in a second 'bump' in the density of the downstream segment. As can be seen, this density 'bump' disappears on the subsequent segment.

Note that the maximum densities on different links may vary due to the fact that different vehicle types have different lengths, and the vehicle type is a stochastic variable. This may result in one link having more long vehicles, and therefore a lower maximum density than another.

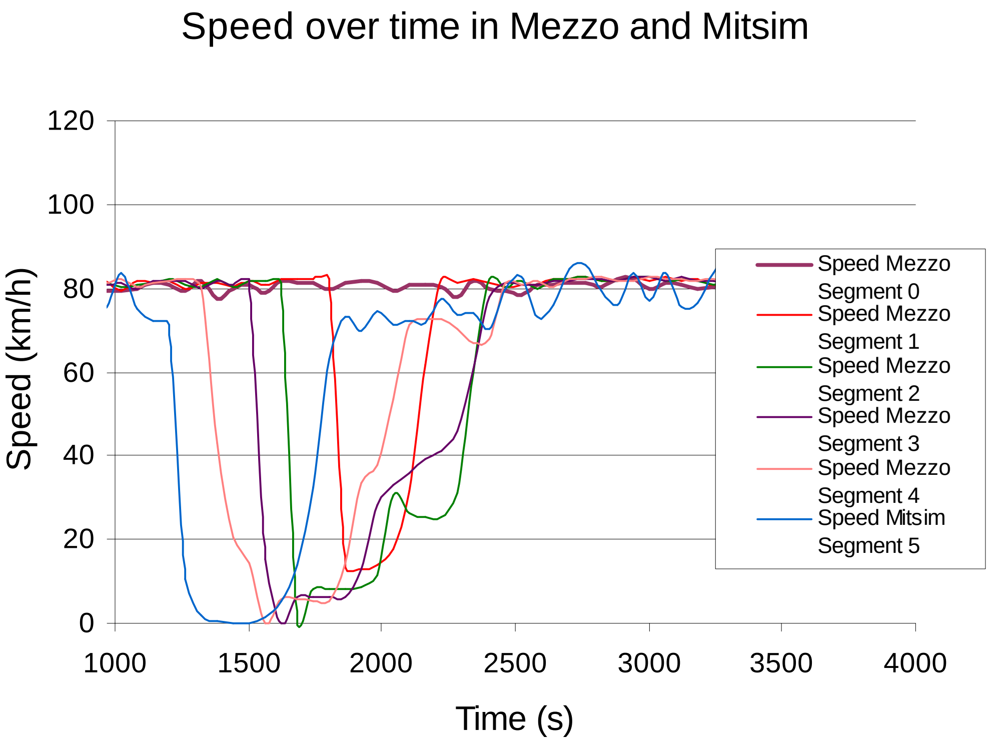


Figure 28. Speed over time.

In the speed plot (figure 28), the same can be observed as in the density. Due to the small secondary bump in the density on segment 2, the speed drops a little, causing a delay in the recovery of that segment. Apart from these effects, which are due to the simplifications explained, the traffic dynamics seem to travel correctly over the meso-micro boundary. At the boundary no disturbances are introduced (which would show up in the graphs presented, as well as the acceleration graphs in chapter 4).

Figure 29 shows a speed/density pattern for Mitsim segment 5 versus Mezzo segment 4. It shows the progress of the density and speed through different levels of congestion. The consecutive speed/density points over time are linked by lines to show the progress. As can be seen, the speed drops when the density starts to increase (in the direction of the downward arrow), both in Mitsim and in Mezzo. At the time of queue dissipation, the density slowly decreases and the speed recovers (upward arrow). The speed increases more rapidly for Mezzo than for Mitsim.

However, the patterns in Mezzo and Mitsim are quite alike for most of the speed/density range.

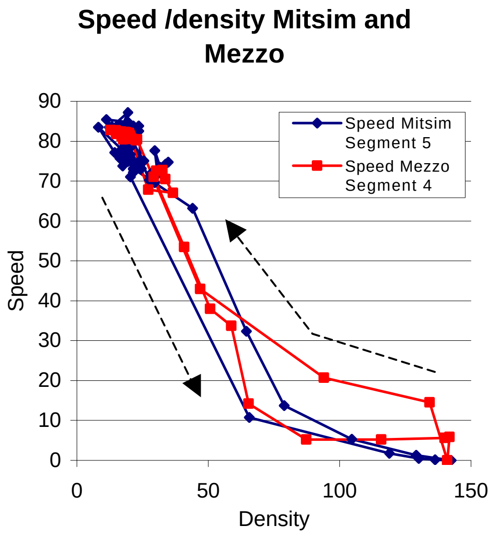


Figure 29. Speed/density for Mitsim and Mezzo segment

Note that in Mezzo, the speeds collected are the *space mean speeds*. The reason why these are relatively high at high density is that the queue on (Mezzo) segment 4 exists only for a short time. It reaches maximum density, when the incident on segment 5 (Mitsim) clears and the queue starts to dissolve. Therefore, the vehicles on segment 4 are delayed only for a short period, resulting in space mean speeds that are relatively high, compared to those one would expect from point speeds at high density.

Micro to meso

In this second test the evaluation focused on the propagation of a queue originating in a Mezzo segment, across the Micro-Meso boundary into Mitsim. The incident starts the queue build-up at segment 8 and then propagates over segment 7, across the boundary into Mitsim segment 6 and then 5, and again over the boundary to Mezzo segment 5, where it finally dissolves.

In figure 30 the cumulative flows are shown for all segments involved. It is clear that the plot is very similar to the one presented previously (figure 26). Both the queue back and queue front travel properly over the boundaries (both the micro-meso and meso-micro).

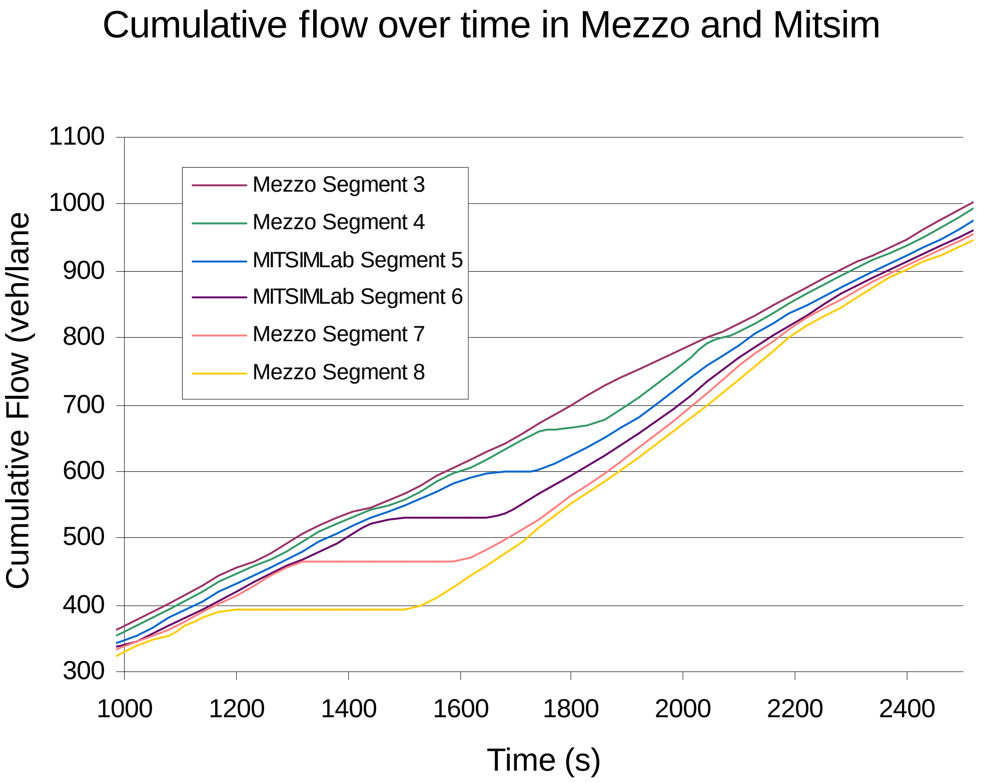


Figure 30. Cumulative flows in Mezzo and Mitsim

The density plot in figure 31 shows also the propagation of the increase and decrease in density upstream in the network. What can be observed is that the density in the two Mitsim segments does not reach the real jam density. This is due to the fact that in Mitsim, as in reality, the queue forms at a lower density, and then with time compresses to actual jam density. From figure 27 depicting the density over time in the micro-meso scenario, it can be noticed how the density quickly accumulates to around 120 veh/km/lane and then slowly increases to 140 veh/km/lane, whereas the density in Mezzo increases at a steady rate all the way to jam density. This is a logical consequence of the fact that Mezzo does not model the vehicle interactions directly, and therefore does not reproduce the stop/go traffic that precedes the fully compressed jam in reality. Therefore, in the propagation of the queues in this second scenario, the segments in Mitsim never reach the jam density fully, but do block back, as can be seen in figure 31.

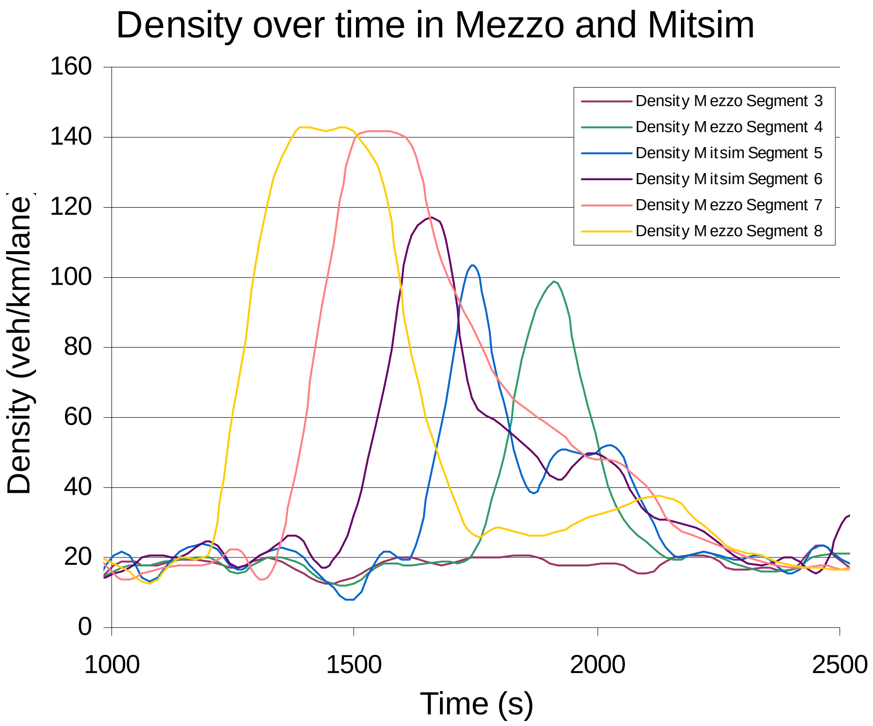


Figure 31. Density over time.

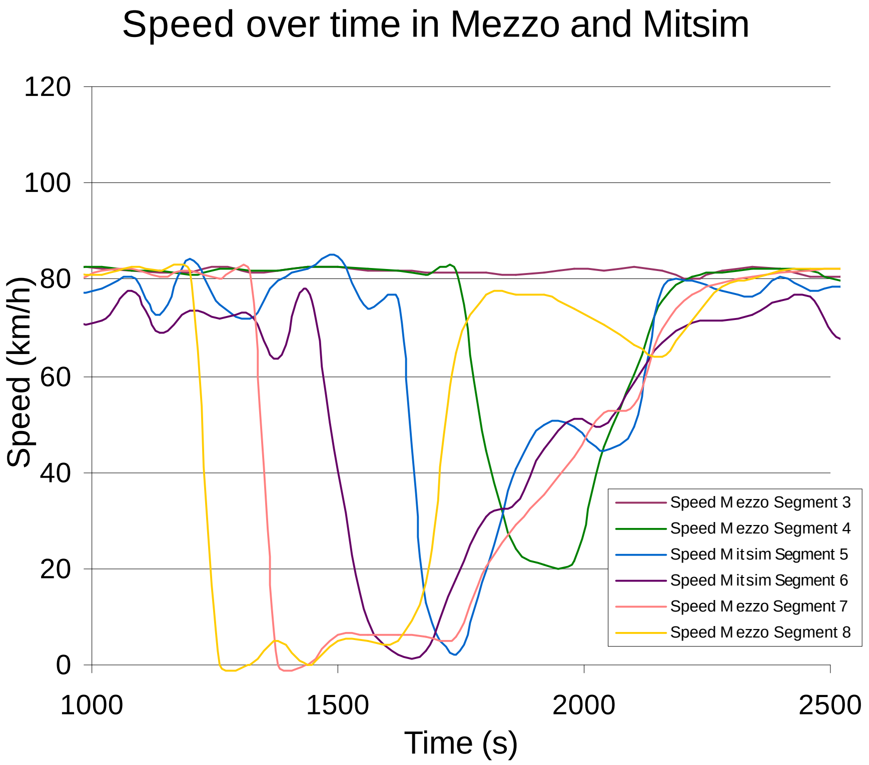


Figure 32 . Speed over time.

In the speed plot (figure 32) it can be noted that while the progress of speed drops progresses through the segments upstream, the recovery seems to differ from that observed in the earlier plot in the meso-micro scenario. It seems reasonable to attribute this difference to the fact that the Mitsim segments never get completely jammed and therefore the recovery of the speed in segment 7 is delayed due to a relatively quick inflow from the Mitsim segment. On the other hand it is interesting to note that the Mitsim segment 6 also shows a 'second bump' similar to the one discussed in the meso-micro scenario. Again this may be the result of the simplification in Mezzo, which ignores the anticipation of vehicles approaching slow moving traffic of a dissipating queue, which may result in a too large outflow in segment 3.

5.4.2 Brunnsviken network with MiMe

The previous section evaluated the functioning of the micro-meso boundaries using a simple artificial net. In this section the MiMe model is applied to the same Brunnsviken network as was used earlier in this chapter for Mezzo.

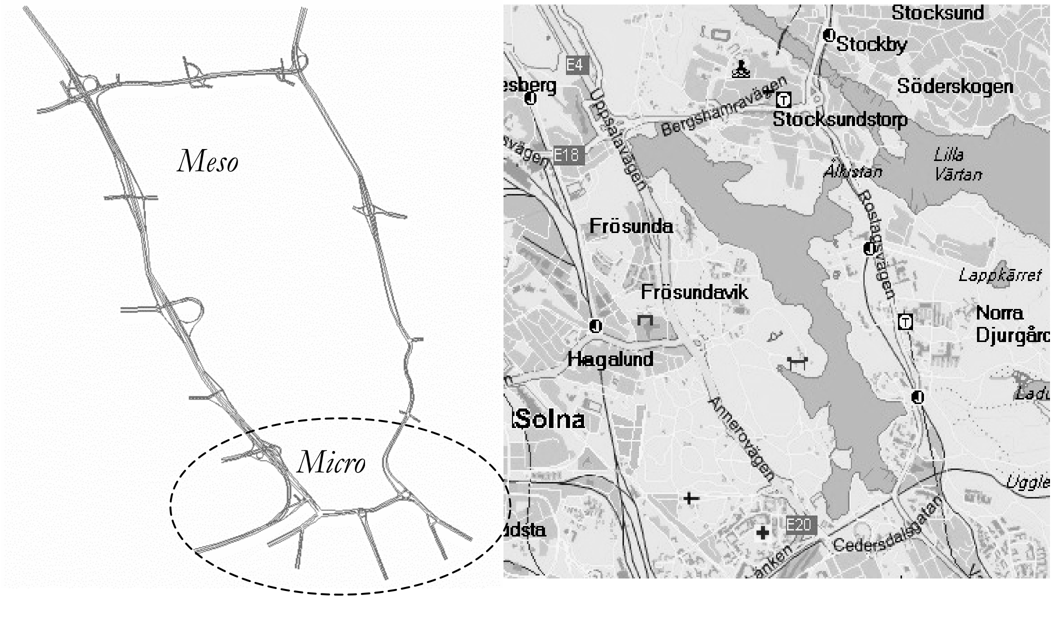
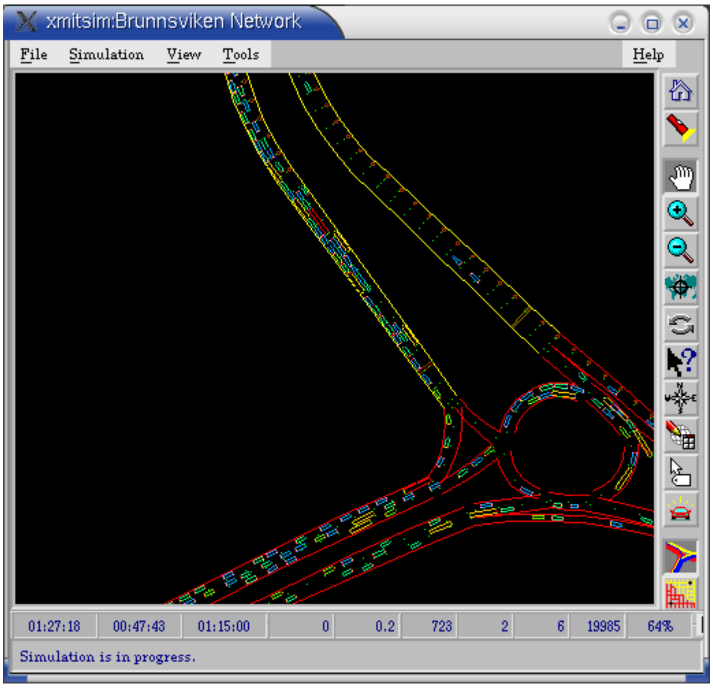


Figure 33. Application of the hybrid model (MiMe) to the Brunnsviken network

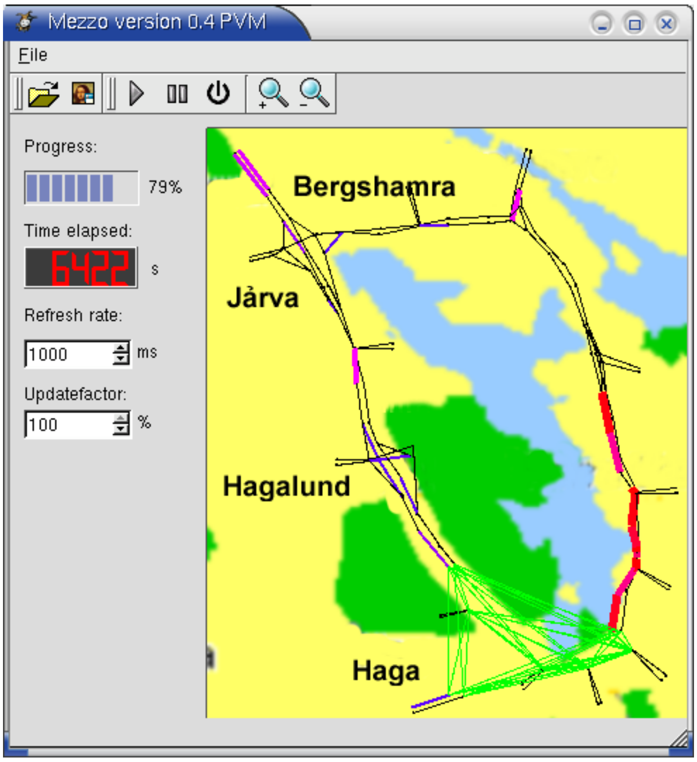
The network is cut in two parts (figure 33), Mezzo for the northern part and Mitsim for the southern part, which includes the traffic control in the southwest (Norrtull, Sveaplan) and the roundabout in the south-east. As we saw from the application of

Mezzo only, this area may benefit from microscopic modelling, since the performance is controlled by coordinated traffic signals and the traffic performance at the Roslagstull roundabout. See figure 34 and 35 for the screenshots of the Mitsim and Mezzo simulation models.



```
01:27:18 | 00:47:43 | 01:15:00 | 0 | 0.2 | 723 | 2 | 6 | 19985 | 64%
Simulation is in progress.
```

Figure 34. Brunnsviken MiMe: MITSIMLab Screenshot.



Progress: 79%

Time elapsed: 6422 s

Refresh rate: 1000 ms

Updatefactor: 100 %

Figure 35. Brunnsviken MiMe: Mezzo screenshot

The Mitsim network is adapted from the Brunnsviken network used in (Toledo, T. et al. 2003), and uses the same parameter set and traffic control plans as were used in that study. In addition, the OD matrix that was estimated in that study is used here (for the entire network).

The green lines in figure 35 show the virtual links that represent the subpaths in Mitsim. In this example, each boundary entry and exit pair has only one possible path, but in other cases where there is a more complicated micro subnetwork, each possible subpath is assigned its own virtual link. As was described in Chapter 4, these virtual links function as normal links in the Mezzo route choice, and performance measure collection (such as flows, travel times, speeds). Once a vehicle passes the boundary to Mitsim, the virtual link is translated into a subpath to the boundary exit. These virtual links and subpaths were specified by hand, although it would be possible to generate them automatically.

In order to prepare the historical travel times and route set, the same iterative procedure as described previously for the Mezzo only case was applied.

Results

In table 5, the results from the MiMe model on the Brunnsviken network are presented. Overall, they appear to be a bit closer to the measurements than those of Mezzo alone. This is likely attributable to the fact that the micro modelling of the south area offers more detailed modelling of the coordinated traffic signal control in the Norrtull-Sveaplan area. The Theil U values show good correspondence of the simulated with the actual flows on most locations, with a minimum of 0.021 at E20 Roslagstull NB and E20 Lappkärrsberget NB, and the largest value of 0.161 at E20 Roslagstull SB.

	E4		E20 Roslagstull		E20 Lappkärrsb.		E18		Cedersdalsgatan		Sveavägen	
	SB	NB	SB	NB	SB	NB	EB	WB	EB	WB	EB	WB
RMSE	70	79	152	16	107	16	54	103	50	76	67	70
RMSNE	5%	4%	38%	4%	15%	4%	11%	17%	9%	11%	10%	11%
MNE	2%	2%	28%	2%	8%	2%	9%	11%	0%	-1%	-1%	5%
Theil U	0.024	0.033	0.161	0.021	0.060	0.021	0.045	0.069	0.068	0.067	0.072	0.052

Table 5. RMSE, RMSNE, MNE and Theil U for 15 min flow counts

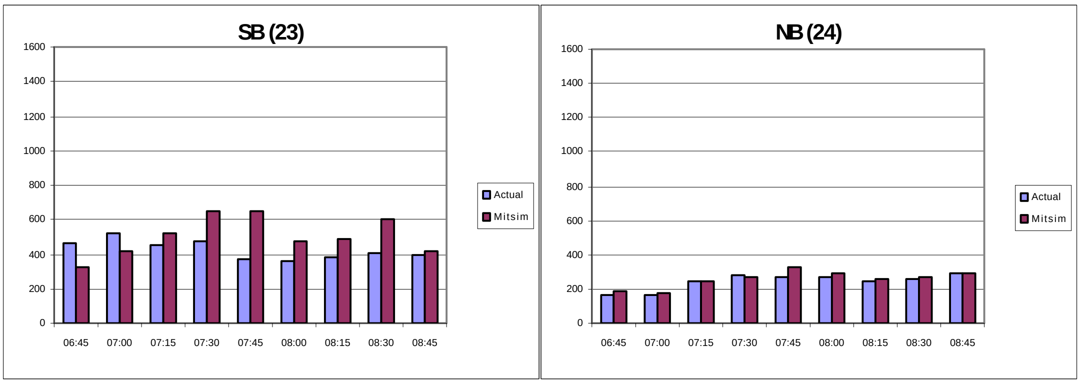


Figure 36. Flows in MiMe at Roslagstull (southeast) Southbound(left) and Northbound(right)

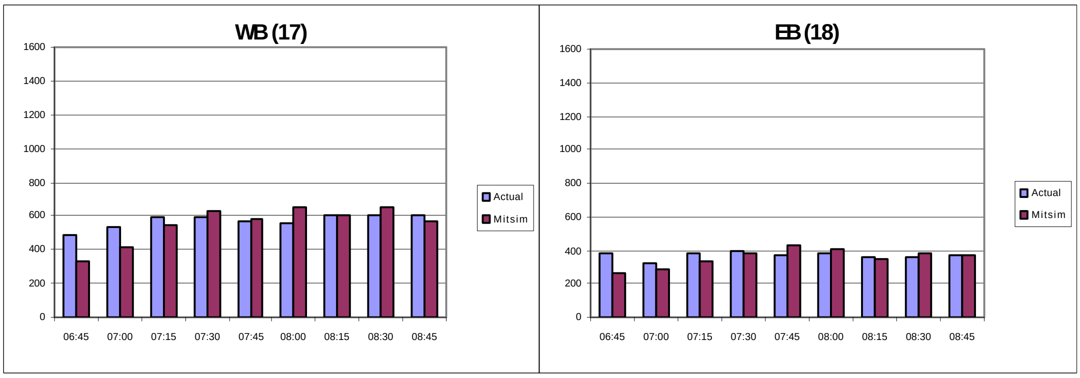


Figure 37. Flows in MiMe on Cedersdalsgatan (between Roslagstull and Sveaplan). Left: Westbound. Right: Eastbound

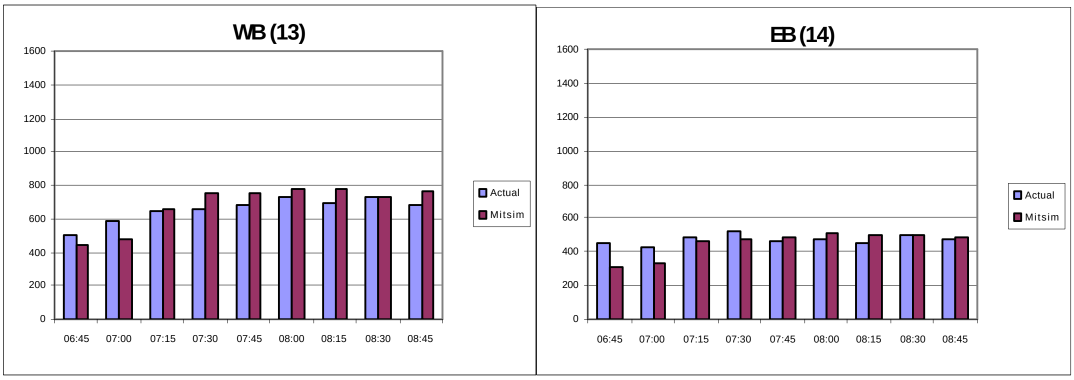


Figure 38. MiMe Flows on Sveavägen (between Sveaplan and Norrtull). Left: Westbound. Right: Eastbound

In figure 36, 37 and 38 the flows for the Mitsim section of MiMe are shown. The hybrid model performs a bit better than the mesoscopic one for this area, but the flows at Roslagstull Southbound are still on the high side (and slightly higher than in Mezzo). The MNE of 28% and RMSNE of 38% show that the prediction of the flow at this location is still the largest source of error. The MNE and especially the RMNE errors at Cedersdalsgatan and Sveavägen seem to show that the exact replication of the signal control at Norrtull and Sveaplan may have improved the quality of the simulation, compared to the Mezzo only case, although the Theil U values are slightly better for the Mezzo results.

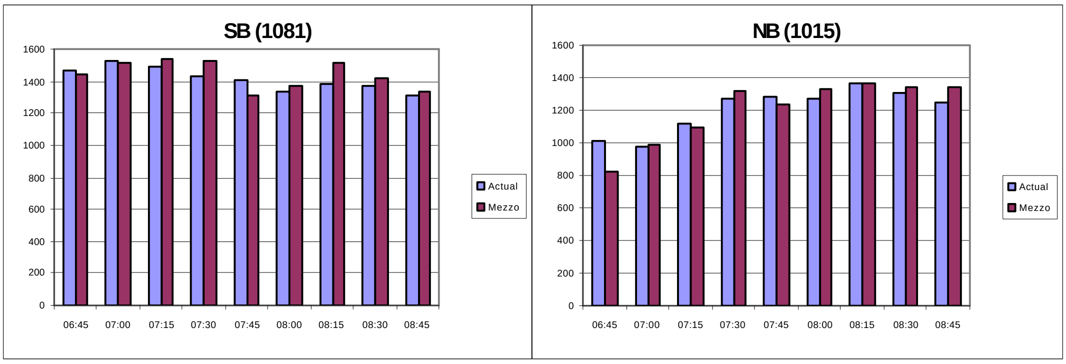


Figure 39. Flows in MiMe on E4. Left: Southbound, Right: Northbound.

Figure 39 shows the flows on the E4, which is the main arterial from the North into Stockholm. Except for the warming up period (at least in northbound direction) (6:45-7:00), the simulated flows match the measurements well. These are important locations, since the flows are high, and small deviations in the assignment may cause the queue at Norrtull to back up much more quickly than usual, causing the flow in the southbound direction to drop significantly. The RMSE, RMSNE, MNE and Theil U errors in table 5 corroborate these observations and show that MiMe and Mezzo perform equally well for this location (with MiMe slightly better for Northbound direction).

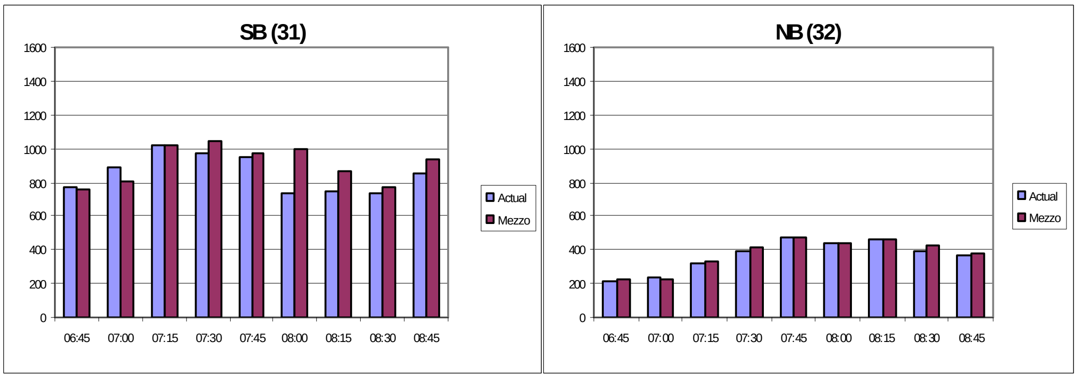


Figure 40. MiMe flows on E20 Lappkärrsberget. Left: Southbound, Right: Northbound direction

In figure 40 the flows on the E20 at Lappkärrsberget are shown. In the northbound direction the match is closer than in the Mezzo only case, but in the southbound direction the flows are too high, and show a deviation comparable to the Mezzo only case. The queue that builds up at Roslagstull backs up properly, and the flows in the periods 8:15-8:30 are limited due to the stop-and-go traffic that occurs on this stretch. In reality this capacity breakdown already occurs at 8:00. Both MiMe and Mezzo over-predict the flows for this period.

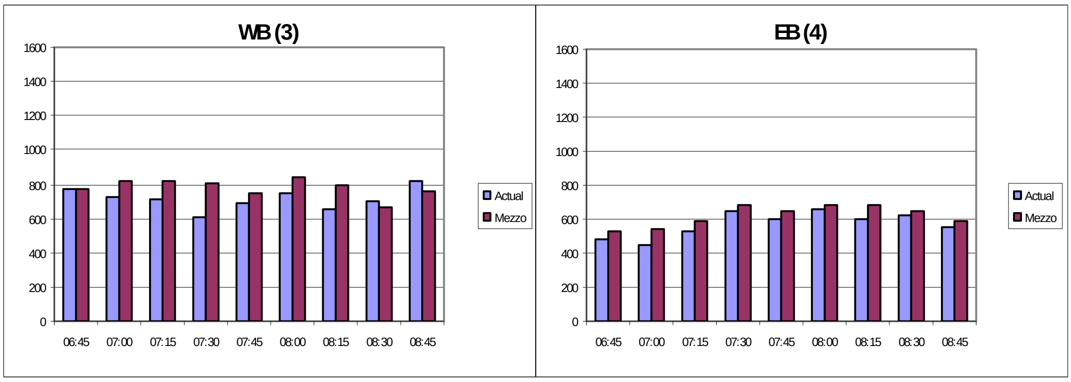


Figure 41. MiMe flows on E18 Bergshamraleden. Left: Westbound, Right: Eastbound

The flows on the E18 Bergshamraleden (figure 41) match the measurements well in eastbound direction, but are on the high side in the westbound direction. Nevertheless, the results here are significantly better than in the Mezzo only case. As table 5 shows, the RMSNE (11% EB, 17% WB) and MNE values (9% EB and 11% WB) are better than the Mezzo values (RMSNE: 16% EB, 21% WB and MNE: 12% EB and 17% WB). Maybe this is due to the better assignment that is caused by the better prediction of the operation of the microscopic area in the south part of the network. As a result, possibly fewer drivers from the two main origins in the north (E4 and E20) take the 'alternative route' via the E18 into Stockholm.

5.4.3 Conclusion

In this section the performance of the implemented hybrid model (MiMe) was demonstrated using laboratory tests to show the validity of the traffic dynamics across the boundaries. The flows, density and speeds propagate properly across the boundaries, in both upstream and downstream directions. The network test with the Brunnsviken network showed that the virtual links function properly in the route choice, and the flows inside both the microscopic and mesoscopic area were matching the measurements. The introduction of a microscopic area did not introduce any disturbances, and may have improved the quality of the results.

5.5 Comparison Micro, Meso and Hybrid

The micro model (MITSIM), meso model (Mezzo) and hybrid model (MiMe) have been applied on the same Brunnsviken network, using the same OD data. In (Toledo, T. et al. 2003) the comparison for MITSIM is shown in detail, and the other two comparisons were discussed in this chapter. In table 6 the overall RMSE, RMSNE, MNE and Theil U errors are shown for the three models, as well as the U^M (bias or systematic), U^S (variance) and U^C (covariance or non-systematic) proportions of the error.

	MITSIM	Mezzo	Hybrid (MiMe)
RMSE	75.43	82.45	80.55
RMSNE	9%	14%	12%
MNE	3%	6%	6%
Theil U	0.051	0.055	0.054
U^M	0.001	0.147	0.075
U^S	0.010	0.002	0.017
U^C	0.989	0.852	0.909

Table 6. RMSE, RMNE,MNE and Theil U errors for the micro, meso and hybrid model.

MITSIM, which has been calibrated extensively, performs the best of the three models, followed by the hybrid model MiMe and Mezzo. All three models seem to have a positive bias (MNE) overall, which is possibly due to the estimated OD matrix. The difference between the MiMe and Mezzo models is small, but important. Although the MNE errors are the same, The RMSNE errors show that for the individual measurement periods, the MiMe model matches better the observations. The RMSE values of 75.43, 82.45 and 80.55 vehicles per measurement location and 15 min. time period, show that all three models fit the measurements well. It should be noted that the (time-sliced) OD matrix was estimated for MITSIM, using MITSIM, and the other models used this same matrix. In addition, the Mezzo and MiMe models (at least the Mezzo part) were only roughly calibrated.

The overall U corroborates the two main observations from the previous measures, that on the one hand the models produce overall comparable results, and on the other hand that MiMe produces slightly better results than Mezzo and Mitsim is somewhat better than MiMe. The U^M proportions show an interesting point, namely that the bias proportion of Mezzo is larger than MiMe and Mitsim has hardly any bias. The U^S proportions, on the other hand, show that the variations in Mezzo follow best the variations in the data, followed by Mitsim and MiMe. The U^C proportion, finally, shows again that MITSIMLab has the largest proportion non-systematic error, followed by MiMe and Mezzo.

Without replicating the complete set of flow diagrams, in figure 42 the 15-min flow counts for the three models and the measurement data is shown for the E4 in Northbound direction and the E18 in the Southbound direction (near Lappkärrsberget).

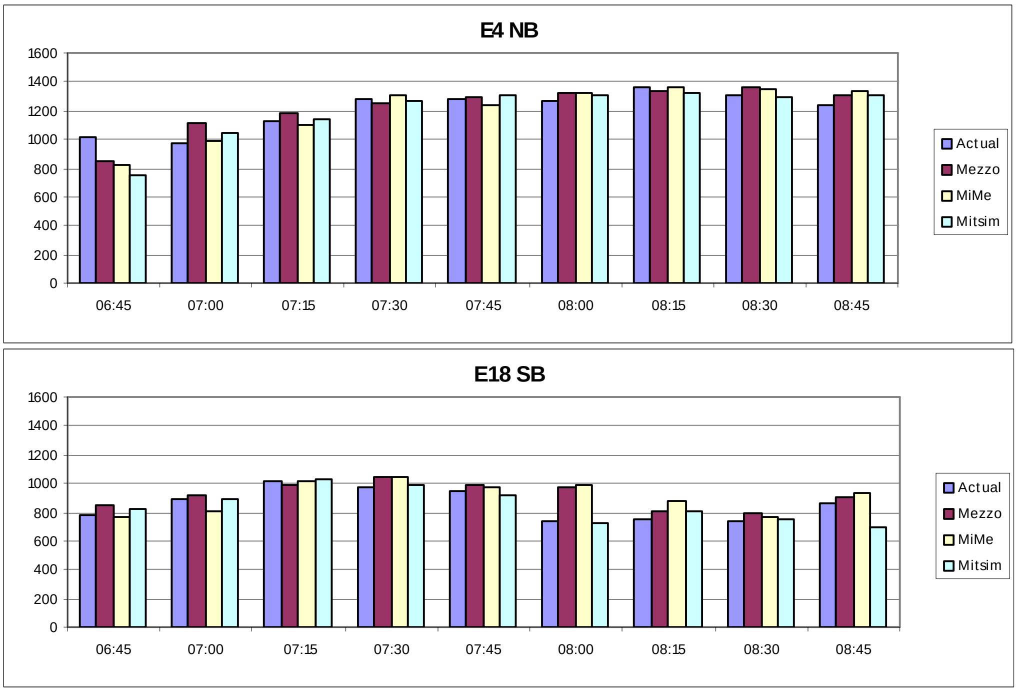


Figure 42. Flows at E4 Northbound (top) and E18 Southbound

The flows on most locations were relatively close for the three models (except for the warm-up period 6:45-7:00), and the only location where MITSIM seemed to perform clearly better than MiMe and Mezzo is on the E18 Southbound, for the period 8:00 - 8:15, where MITSIM reproduces the queue that has backed up all the way from Roslagstull, and Mezzo and MiMe reproduce the queue at that location only in the next two periods (8:15-8:30 and 8:30-8:45).

Chapter 6. Conclusions

6.1 Introduction

The aim of this thesis was to investigate the possibility of integrating mesoscopic and microscopic traffic simulation models in such a way that the result is a consistent and coherent hybrid simulation model that combines the strengths of meso and micro models, and diminishes the effects of the limitations of these types of models.

6.2 Contributions

This thesis contains the following contributions to the existing body of research:

1. Mezzo, a new mesoscopic traffic simulation model is presented, which is event-based, vehicle based and simulates the forming and propagation of queues. The queue dissipation is simulated using shockwaves, thereby ensuring correct spatial and temporal location of congestion. This model is especially suited for combination with microscopic models, since it is vehicle based and event-based, but is equally well suited for stand-alone use on large networks. Due to the structure of the route-choice, the model does not need to iterate, once a set of historical link travel times and routes are generated. This means that even large networks can be simulated faster than real time.
2. A framework has been developed for integrating microscopic and mesoscopic traffic simulation models into one hybrid model. This framework contains a set of requirements that need to be met, as well as an integration architecture to meet these requirements. The issues that are presented and resolved range from consistent meso-micro boundary transitions, consistent network representation and route choice, to inter-model communication and synchronisation.
3. As part of the integration architecture, a new method of initial speed generation for vehicles entering microscopic models is presented. This method proves to be superior to the existing method and other methods tested, in that it creates minimal initial acceleration disturbances, and

increases the implicit capacity of the entry links in the microscopic model. Field data supports the underlying assumptions of highly correlated speeds for close-following vehicles, and decreasing correlation when the time headway to the vehicle in front increases. This method is general for micro simulation models.

4. Another contribution that is part of the integration architecture is the mesoscopic virtual link, which allows for a continuous representation of the network, where the pre-trip route choice decisions are taken on the mesoscopic level. The virtual links are abstractions of the (sub)paths in the microscopic area, and for each pair of boundary entry and exit nodes there is a virtual link for each used (predetermined) microscopic subpath. This allows for a consistent representation of routes throughout the mesoscopic network, and the collection of time dependent travel times for the virtual links, which is necessary for a consistent route choice over both the meso and micro parts of the hybrid model.
5. Conversely, microscopic virtual links in the microscopic model provide the flexibility needed for en-route diversions. Microscopic virtual links represent the continued path from boundary exit nodes to the vehicles' final destinations. Using these microscopic virtual links enables vehicles in the micro model to choose alternative routes that go through a different exit node.
6. The integration framework has been implemented in a working hybrid meso-micro model (MiMe) consisting of the Mezzo mesoscopic and Mitsim microscopic simulation models. The models communicate and synchronise every micro (Mitsim) time step via a Parallel Virtual Machine (PVM), which can contain multiple physical computers. This implies that the Mezzo and Mitsim components can be run on different computers.
7. The Mezzo mesoscopic model and the MiMe hybrid model are evaluated and tested using both laboratory case studies as well as field data from a small network in the north of Stockholm (Brunnsviken). The results show a consistent behaviour at the meso-micro boundaries, even in the presence of queue propagation and dissipation.
8. The Mezzo model applied to the Brunnsviken network showed good correspondence to the measurement flows for most of the locations. The MiMe application to the same network also showed good correspondence to the measurement flows for most locations, and indicates the applicability

and relevance of applying microscopic modelling in certain areas of the network where the mesoscopic modelling may be too coarse.

9. As part of the Mezzo mesoscopic model, a new algorithm for generating path set and historical link travel times is presented, where the path (choice) set for the (pre-trip) route choice and the (equilibrium) historical link travel times are generated using consecutive iterations, alternating between simulation to find new travel times (which are averaged with the previous set of travel times), and a shortest path algorithm to find new routes (given the new travel times). This method generates one or few paths for origin-destination (OD) pairs with little demand that go through uncongested parts of the network, and many paths for OD pairs that go through heavily trafficked parts of the network, provided there are sufficient alternatives. Convergence is reached when for a specified number of iterations the output travel times do not differ significantly (given a specified margin) from the input travel times, and no new routes are found.

6.3 Discussion and Further Research

As hybrid simulation modelling is a rather new field of research, the existing body of knowledge was small at the start of this thesis. Although the interest and activity in this area has increased over the last few years, and this thesis has tried to contribute to this body of knowledge, there remain many aspects to be investigated and solved.

6.3.1 Theoretical aspects

One of the issues that need further investigation is the representation of networks and route choice. In this thesis the concept of virtual links was used to create a homogeneous representation of the network, which enabled homogeneous route choice.

However one of the problems that has not yet been solved is the problem of extra capacity, when a microscopic window contains more links than the original mesoscopic network, or in general, when part of the network includes only the main roads and part of the network includes the small roads as well. Virtual links do not solve this problem, since each ‘extra’ path that uses the added small roads would present an extra virtual link, not present in the original network representation. Maybe one should conclude that both mesoscopic and microscopic sub-networks should include all existing links.

Another aspect that needs consideration is the selective blocking of boundary nodes, especially meso-to-micro boundaries. When at a meso-to-micro boundary a queue is building on only one lane, but the remaining lanes remain fluid, the only vehicles in Mezzo that need to be blocked are those that have to use this particular lane. While the virtual links help to deal with the correct selective blocking of lanes, the continuation of the queue over links upstream in Mezzo will be a homogeneous queue, where the lanes are not modelled separately.

A possible solution may be to extend Mezzo to model individual lanes. This would also enable the modelling of bus-lanes. However, this would require the route-choice to be extended to include lane-choice, and this lane choice would have to be made consistent with the lane-choice and look-ahead behaviour of the microscopic model. In addition this would add to the model's complexity and reduce computation speed. In the application of the hybrid model to the Brunnsviken network, this problem was solved by increasing the look-back limit of turning movements for those parts of the network where this 'selective' blocking is known to occur.

Another aspect that has received little attention in this thesis is the route-choice implementation. In chapter 3 it was mentioned that the utility function is a function of travel time on the path. It is recommended that this function is extended to include other decision variables such as the distance, number of stops, number of signalised intersections, types of road, etc.

However, in the case of MiMe, if this function is extended, all these aspects will have to be collected throughout the network, and be made consistent across the micro and meso parts.

In addition it is recommended that the implementation of the route choice mechanism is improved by using C-Logit or Path Size Logit to account for the overlap between alternative paths. This comes at a computational cost, since the overlap factors for the different paths have to be calculated.

Again, in the case of MiMe, this means that also the overlap (commonality) factors of virtual links have to be calculated and communicated.

The new method of combined path and travel time generation presented in this thesis, should be studied more carefully, and compared with other methods. One important aspect that should be studied is the coverage of the paths in a network that are found using this method, with the paths that drivers actually use.

Another issue is that of dependence on initial conditions. The path set generation always starts with the shortest free flow paths for each OD pair. However, which paths are found in the next iteration depends on the outcome of the simulation, which is stochastic and can vary between runs. The paths that are found in the third iteration are a function of *both* the paths found in the previous iterations *and* the travel times that are generated using those paths. It seems possible that small differences in simulation outcomes may result in different paths to be found, or the paths are found in a different order. This means that due to initially small differences it is possible that different path sets are generated, and also that a different equilibrium is found. This emphasizes the importance of sufficient replications of simulation runs during the process of path and travel time generation.

More specifically, it would be advisable to extend the scheme to the following procedure:

1. Start with free flow travel times and free flow shortest paths.
2. Iterate the simulation to create stable travel times that reflect the effect of the current set of routes.
3. Within each iteration, replicate the simulation to account for the stochastic nature of the model and its results.
4. Find new routes using the stable travel times, and repeat step 2 and 3.
5. Provide an automatic or manual check of the found paths, to minimise spurious paths in the route set, and to determine when the path set is sufficiently large.

In the case of heavily congested networks, it may be advisable to downscale the OD matrix during the initial iterations, since the extreme congestion that will otherwise occur may lead to unrealistic paths to be found, which will skew the assignment (since every path will get at least *some* share of traffic), thereby complicating the search for equilibrium assignment. Once the travel times converge and the path set is determined, the OD matrix can be scaled back to its original size.

6.3.2 Empirical and practical aspects

Mezzo

The tests in chapter 5 show that the Mezzo model produces good results, but show also that there is room for improvement. They emphasize the importance of careful network coding, and calibration of speed/density functions, even on the mesoscopic level. They also seem to suggest that it is important to model the operation of traffic signal control in more detail. This aspect should be added to the implementation of the model. As was mentioned before, the route choice should be improved to include the overlap between paths, and a more realistic cost function. Another important aspect is that of simulation replications. As described in (Burghout, W. 2004), the number of replications needed depends on the variability found between simulation runs, and should be determined 'on the run'. Based on a number of iterations, the required number of iterations can be calculated automatically, and updated after each extra iteration. This method should be incorporated in the simulation model implementation. Furthermore, the model needs to be tested and validated on more and larger networks, and in greater depth.

Another aspect is the possibility of simplification of the model. At present, the traffic in Mezzo links is non-FIFO (First-In-First-Out). This was meant to model the effect of vehicles that enter at lower densities and higher speeds, can overtake vehicles that entered when the density was higher, and have lower speeds. However, this assumption may not be very realistic, and adds to the computational time of the model, as all vehicles need to be sorted by earliest exit time, upon entering a link. It is also possible that the operation of shockwaves will work better when a strict FIFO procedure is applied, since in the current implementation, the vehicles in the queue may receive a 'delayed' exit time, based on the time it takes for the start-up shockwave to reach them, but the vehicles that enter afterwards can get an exit time that effectively moves them past part of the queue.

Another simplification that may speed up the computation is the use of piecewise linear speed/density functions, that include the k_{min}, k_{max}, V_{min} and V_{max} limits, but have a linear function instead of the $O(x^n)$ type of function presented here. From the tests in chapter 5 it seems that the values for k_{min}, k_{max}, V_{min} and V_{max} are more important than the parameters a and b in the calibration of the speed/density functions. A further optimisation of computation time may be the replacement of stochastic by deterministic turning-servers, especially for locations where the stochasticity does not seem important, such as for straight-going movements on freeways and motorways.

For those turning movements that need stochastic servers, it should be investigated what level of variance is appropriate. This is a difficult thing to measure, especially as the measured differences in flows at a location do not need to represent fluctuations in capacity, indeed they to not need to represent capacity at all, but may be fluctuations in demand.

In addition, there are many improvements possible to the graphical user interface, and the implementation of the model in general. Integration with a disaggregate trip generation model may extend the area of application of Mezzo (or the MiMe hybrid model) to include traffic planning for larger areas and time scales. There are plans to integrate Mezzo with such a model developed at KTH.

MiMe

The hybrid integration architecture presented in this thesis, was developed into a hybrid micro-meso model called MiMe. This implementation consisted of an existing microscopic simulation model, MITSIMLab, and a new mesoscopic model Mezzo. The integration of existing models required some simplifications to the integration architecture, to limit the communication overhead, and to make the network representations of the two models compatible. It would be interesting to develop a model from scratch that includes both the microscopic and mesoscopic simulation components. In this case, the inter-model communication overhead is not as critical, as both models will be part of the same program. In addition, a common network format might be possible, especially if the mesoscopic model is lane-based as opposed to link or segment based.

On the other hand, there exist many microscopic models today, which have continuously been developed and improved. Therefore the replacement of the MITSIMLab micro component with another microscopic model should be investigated. Since these are often commercial models, the amount of freedom in adding and modifying these models is usually limited to an API in which existing functions in the program can be called and existing variables can be read and set.

This means that adding components, as we saw was necessary, may be difficult. In addition, it may be possible that a more limited implementation of the framework can provide a sufficiently good hybrid model.

At the minimum the following component would need to be added:

1. Communication of vehicles across meso-micro boundaries.

This component would require the addition of a communication module, the possibility to insert and extract vehicles from the micro model, and for consistent vehicle trips across the network, the addition of vehicle state variables such as global ID, (meso) path ID, vehicle type.

In addition the following components are needed in order to avoid disturbances at the boundary locations:

2. Communication of traffic conditions upstream and downstream.
3. Virtual vehicle at end of micro segments

These components require more changes and additions to the model. Especially the virtual vehicle concept needs the possibility to change / influence the driver behaviour module, to make it react to this new object as if it were a regular vehicle object.

The last components needed for correct paths throughout the hybrid model are:

4. Meso virtual links.
5. Micro virtual links.

The Meso virtual links are associated with subpaths through the micro model. This requires the possibility in the micro model to define fixed paths from entry to exit points, overriding any internal route-choice algorithm. Without this component, it is still possible to create a hybrid model, but the route choice may not be consistent across the meso and micro models. The fifth component, the micro virtual links, takes care of the en-route deviations that vehicles in the micro area may want to execute. With the micro virtual links, the en-route choice algorithm can determine paths through different exit nodes into the meso model. When the micro areas are small, or when changes of exit points are not realistic, due to the topography of the network, this component is not needed in the implementation.

The MiMe model was tested on laboratory networks to study the functioning of the boundary transitions, and on the Brunnsviken network. In addition to these evaluations, many more validation exercises are needed to establish the quality, strengths and weaknesses of the hybrid model. These validations should focus on larger networks, and other quantities besides flow counts, such as speeds and travel times.

Throughout this thesis the examples concerning MiMe application concerned a single microscopic area surrounded by a larger mesoscopic network. Neither the integration architecture nor the implemented prototype presented in this thesis, are limited to such a configuration. In fact, a hybrid network can consist of any number of microscopic and mesoscopic areas. An example of an application where there are two mesoscopic areas that 'sandwich' a microscopic ring is presented in figure 1. A typical area of application where such a configuration of micro and meso areas would be relevant is the analysis of network effects of congestion charging, where the toll-collection (manual and automatic) ring around a city centre is simulated in microscopic detail, but the rest of the network on the mesoscopic level.

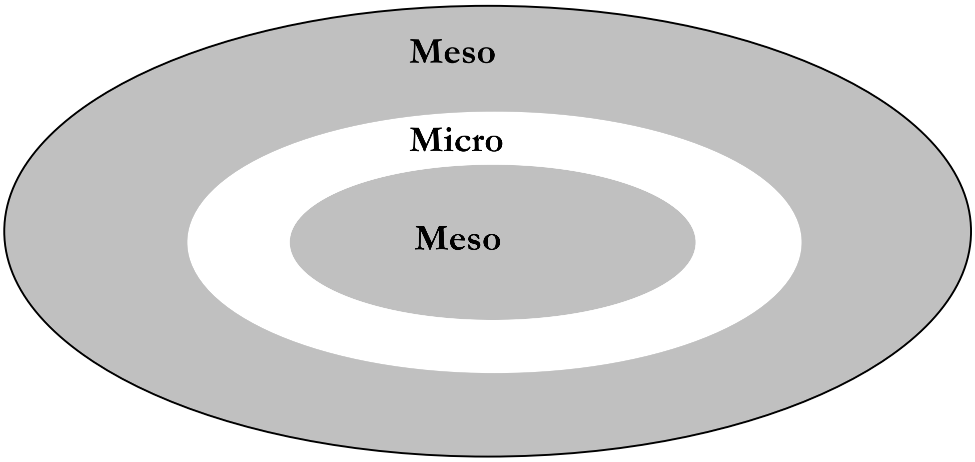


Figure 1. Micro area 'sandwiched' between two meso areas.

6.4 Summary

This thesis has demonstrated the possibility of a hybrid mesoscopic – microscopic traffic simulation model. In a conceptual framework, the consistency requirements for integration of meso and micro submodels were defined, and an integration framework was presented that addresses the issues and requirements defined in the framework. The resulting hybrid model architecture has been implemented using an existing micro model and a newly developed meso model. With this implemented hybrid model the correctness and applicability of the developed concepts was demonstrated, and the model was evaluated using flow counts from a small network in North Stockholm.

The newly developed mesoscopic model was presented as well in this thesis, and was evaluated on both laboratory and real networks from the North of Stockholm. The tests show the validity and applicability of both mesoscopic and hybrid models, and indicate that hybrid modelling is not only possible, but may increase the quality of the results as well.

References

Ahmed, K., 1999. Modeling Drivers' Acceleration and Lane Changing Behavior, PhD thesis, ITS Program, Massachussettes Institute of Technology, Cambridge, MA

Alexandrov, V. & J. Dongarra, 1998. Recent Advances in Parallel Virtual Machine and Message Passing Interface. 5th European Pvm/Mpi User's Group Meeting, Liverpool, UK, Lecture Notes in Computer Science 1497, Springer Verlag.

Antoniotti, M., P. Cooke & A. Deshpande, 1998. SmartAHS User's Manual, http://www-path.eecs.berkeley.edu/smart-ahs/sahs-manual/manual.html Last visited March 2004

Barcelo, J. & J. Casas, 2004. Methodological notes on the calibration and validation of microscopic traffic simulation models. Transportation Research Board Anual Meeting, Washington DC.

Barcelo, J., J. Ferrer & R. Grau, 1997. Microscopic traffic simulation for ATT systems analysis, Universitat Politecnica de Catalunya, Barcelona

Ben-Akiva, M., 1996. Development of a deployable real-time dynamic traffic assignment system, Task D Interim report: analytical developments for DTA system, MIT ITS Program, Cambridge (MA)

Ben-Akiva, M., H.N. Koutsopoulos, R. Mishalani & Q. Yang, 1997. Simulation Laboratory for Evaluating Dynamic Traffic Management Systems. ASCE Journal of Transportation Engineering **123** (4).

Ben-Akiva, M.E., A. Davol, T. Toledo, H. Koutsopoulos, W. Burghout, I. Andreasson, T. Johansson & C. Lundin, 2000. Mitsimlab for Stockholm: enhancements, calibration and validation., CTR, Royal Institute of Technology, Stockholm http://www.infra.kth.se/ctr/publikationer/ctr2000_09.pdf

Ben-Akiva, M.E. & S.R. Lerman, 1985. Discrete Choice Analysis: Theory and Application to Travel Demand. MIT Press, Cambridge MA

Bladh, P., 1996. A Traffic Generation Model for Multi-lane Roads, Masters thesis, Mathematical Institute, Linköpings Universitet,
http://www.infra.kth.se/ctr/projekt/tpma/finalreport/E_Generators.pdf

Booch, G., 1994. Object Oriented Analysis and Design with Applications. Benjamin/Cummings, Redwood City (CA)

Bourrel, E., 2003. Modélisation dynamique de l'ecoulement du traffic routier: du macroscopique au microscopique, PhD Thesis thesis, L' Institut National des Sciences Appliquées de Lyon, Lyon

Bourrel, E. & V. Henn, 2002. Mixing micro and macro representations of traffic flow: a first theoretical step. Proceedings of the 9th meeting of the Euro Working Group on Transportation, Bari, Italy.

Bourrel, E. & J.-B. Lesort, 2003. Mixing Micro and Macro Representations of Traffic Flow: a Hybrid Model Based on the LWR Theory. Transportation Research Board, Washington DC.

Box, M.J., 1965. A new method of constrained optimization and a comparison with other methods. The Computer Journal **8** (1): 42-52.

Broucke, M., P.P. Varaiya, M. Kourjanski & D. Khorramabadi, 1996. SmartCap User's Guide, PATH Technical Memorandum No. D96-53,

Buisson, C., J.-P. Lebaque & J.-B. Lesort, 1996. A discretized macroscopic model of vehicular traffic flow in complex networks based on the Godunov scheme. CESE'96, Lille, France.

Burghout, W., 2001. Evaluation of MITSIMLAB in Stockholm as a tool for evaluation of ITS strategies. Scientific Paper. 8th World Congress on Intelligent Transportation Systems, Sydney.

Burghout, W., 2004. A note on the number of replications in stochastic simulation models, Centre for Traffic Simulation Research (CTR), Royal Institute of Technology, Stockholm

Burghout, W., H. Koutsopoulos & I. Andreasson, 2004. Hybrid mesoscopic-microscopic traffic simulation. Revised Papers CD-ROM. World Conference on Transportation Research, Istanbul.

Burghout, W., H. Koutsopoulos & I. Andreasson, 2005. Hybrid mesoscopic-microscopic traffic simulation. To be presented at: Transportation Research Board, Washington DC.

Bush, B.W., 2000. An Algorithmic Overview of TRANSIMS, Los Alamos National Laboratories,

Bång, K.-L.E., 2000. Traffic in Major Cities - Problems and Prospects, Royal Institute of Technology, Department of Infrastructure and planning. TRITA-IP FR 00-66, Stockholm

Carey, M. & Y.E. Ge, 2003. Comparing whole-link travel time models. Transportation Research **37B**: 905-926.

Cascetta, E., A. Nuzzolo, F. Russo & A. Vitetta, 1996. A Modified Logit Route Choice Model Overcoming Path Overlapping Problems: Specification and some Calibration Results for Interurban Networks. Thirteenth International Symposium on Transportation and Traffic Theory, J.B. Lesort, Ed., Lyon, France, Pergamon.

Cherkassky, B.V., A.V. Goldberg & T. Radzik, 1993. Shortest Paths Algorithms: Theory and Experimental Evaluation, Computer Science Department, Stanford University, Stanford

Chu, L., H. Liu, J.-S. Oh & W. Recker, 2004. A calibration procedure for microscopic traffic simulation. Transportation Research Board, Washington DC.

Cremer, M. & F. Meissner, 1993. Traffic Prediction and Optimization Using an Efficient Macroscopic Simulation Tool. European Simulation Multiconference, Lyon, France.

Daganzo, C., 1994. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. Transportation Research B **28B** (4).

Daganzo, C., 1995. The cell-transmission model, part II: Network traffic. Transportation Research B **29B** (2): 79-93.

Daganzo, C., 1997. Fundamentals of Transportation and Traffic Operations. Pergamon,

Daganzo, C., 1999. The lagged cell-transmission model. International Symposium on Transportation and Traffic Theory, A. Ceder, Ed., New York, Pergamon.

DelCastillo, J.M., 1996. A car following model based on the Lighthill-Whitham theory. proceedings of the 13th International Symposium on Transportation and Traffic Theory, J.B. Lesort, Ed., Lyon, France, Pergamon.

DelCastillo, J.M. & F.G. Benitez, 1995. On the functional form of the speed-density relationship I: general theory. Transportation Research **29B** (5): 373-389.

Drake, J.S., J.L. Schofer & A.D. May, 1967. A statistical analysis of speed-density hypotheses. Highway Research Record **156**: 53-87.

Elloumi, N., H. Haj-Salem & M. Papageorgiou, 1994. METACOR: A macroscopic Modelling Tool for Urban Corridor. TRIennal Symposium on Transport ANalysis, Capri, 1.

Fellendorf, M., 1996. VISSIM for traffic signal optimisation. Traffic Technology International, Dorking, UK.

Ferrari, P., 1988. The Reliability of the Motorway Transport System. Transportation Research **22b** (291).

Gabard, J.F. & L. Breheret, 2000. The Sitra-B+ Microscopic Traffic Simulation Model. Examples Of Use And Future Developments. INFORMS, Salt Lake City.

Gartner, N.H., A.K. Rathi & C. Messer, Eds., 1997. Monograph on Traffic Flow Theory, Oak Ridge National Laboratory

Gawron, C., 1998. Simulation-Based Traffic Assignment; Computing User Equilibria in Large Street Networks, PhD thesis, University of Cologne, Cologne

Gazis, D.C., R. Herman & R. Potts, 1959. Car-Following -Theory of Steady-State Traffic Flow. Operations Research **7**: 499-505.

Gazis, D.C., R. Herman & R.W. Rothery, 1961. Nonlinear Follow-the-leader models of Traffic Flow. Operations Research **9**: 545-567.

Gerlough, D.H. & M.J. Huber, 1975. Traffic flow theory - A monograph. Transportation Research Board Special Report **165**.

Greenberg, H., 1959. An Analysis of Traffic Flow. Operations Research **7**: 79-85.

Greenshields, B.D., 1935. A study in Highway Capacity. Highway Research Board Proceedings **14**: 448-477.

HCM, 2000. Highway Capacity Manual 2000. Tranportation Research Board,

Helbing, D., A. Hennecke, V. Shvetsov & M. Treiber, 2002. Micro- and Macrosimulation of Freeway Traffic. Mathematical and Computer Modelling **35** (5/6): 517-547.

Hennecke, A., M. Treiber & D. Helbing, 2000. Macroscopic Simulation of Open Systems and Micro-Macro Link. In: Traffic and Granular Flow '99. D. Helbing et al., Springer, Berlin**:** 383-388.

Hochstaedter, A., D. Ehmanns & D. Neunzig, 1999. PELOPS as a Tool for Development and Configuration. EUROmotor Telematic/Vehicle and Environment 1999: Adaptive Cruise Control, Series Introduction and Future Development, Aachen, Germany.

Horowitz, R.e.a., 2002. Development of Integrated Meso/Micro scale Simulation of AHS, http://www.path.berkeley.edu/PWM2002/PWM2002presentations/11c.pdf Last visited March 2004

Hourdakis, J., P. Michalopoulos & J. Kottommannit, 2003. Practical Procedure for calibrating microscopic traffic simulation models. Transportation Research Record **1852**.

INRO-Consultants, 1996. Emme/2 User's Manual, Montreal

Jayakrishnan, R., H.S. Mahmassani & T.Y. Hu, 1994. An Evaluation Tool fo Advanced Traffic Information and Management Systems in Urban Networks. Transportation Research C **2C** (3): 129-147.

Jayakrishnan, R., J.-S. Oh & A.B. Sahraoui, 2001. Calibration and Path Dynamics Issues in Microscopic simulation For Advanced Traffic Management and Information Systems. Transportation Research Record **1771**.

Kates, R. & A. Poschinger, 2000. Investigation of a stochastic disaggregation model. World Congress on Intelligent Transportation Systems, Torino.

Law, A.M. & W.D. Kelton, 2000. Simulation Modeling and Analysis. McGraw-Hill,

Leonard, D.R., P. Power & N.B. Taylor, 1989. CONTRAM: structure of the model, Transportation Research Laboratory, Crowthorn

Lerner, G., A. Hochstaedter & R. Kates, 2000. The interplay of multiple scales in traffic flow: coupling of microscopic, mesoscopic and macroscopic simulation. World Congress on Intelligent Transportation Systems, Torino.

Lighthill, M.H. & G.B. Whitham, 1955. On kinematic waves II: a theory of traffic flow on long crowded roads. Proceedings of the Royal Society of London, series A, 229.

Lind, G., F. Davidsson, B. Hugosson & K. Schmidt, 1999. Modeling of ITS applications: Test of four dynamic models, Centre for Traffic Simulation (CTR), Royal Institute of Technology, Stockholm
http://www.infra.kth.se/ctr/publikationer/ctr1999_01.pdf

Magne, L., S. Rabut & J.-F. Gabard, 2000. Towards an hybrid macro-micro traffic simulation model. INFORMS, Salt Lake City, USA.

Mahut, M., 2001. A multi-lane extension of the Space-Time Queue Model of Traffic Dynamics. TRISTAN IV, Azores Islands.

Mahut, M., M. Florian & N. Tremblay, 2002. Application of a simulation-based dynamic traffic assignment model. IEEE 5th International Conference on Intelligent Transportation Systems, S.D. Cheu RL, Lee DH, Ed., Signapore.

Mahut, M., M. Florian & N. Tremblay, 2003. Space-Time Queues and Dynamic Traffic Assignment: A model, algorithm and applications. Transportation Research Board, Washington DC.

May, A.D., 1990. Traffic Flow Fundamentals. Englewood Cliffs, NJ

Merritt, E., 2003. Hierarchical Modeling of Road Traffic Networks, PhD thesis, Department of Infrastructure, Royal Institute of Technology, Stockholm

Messmer, A. & M. Papageorgiou, 1990. Metanet: A Macroscopic Simulation Program For Motorway Networks. Traffic Engineering and Control **31**: 466-470.

Microsoft, 2004. DCOM, http://www.microsoft.com/com/tech/DCOM.asp Last visited 2004-10-15

Montero, L., E. Codina, J. Barcelo & P. Barcelo, 1998. Combining Macroscopic and Microscopic Approaches for Transportation Planning and Design of Road Networks. 19th Conference of the Australian Road Research Board, Sydney.

MPI, 2004. Message Passing Interface, http://www-unix.mcs.anl.gov/mpi/ Last visited 2004-10-15

Nagel, K. & M. Schreckenberg, 1992. A cellular automaton model for freeway traffic. J. Phys. I France **2** (12): 2221-2229.

Newell, G.F., 1961. Non linear effects in the dynamics of car-following. Operations Research **9**.

Nizard, L., 2002. Combining microscopic and mesoscopic traffic simulators, Ecole Polytechnique, Paris

Oh, J.-S., C.A. Cortes, R. Jayakrishnan & D.-H. Lee, Microscopic Simulation with Large-Network Path Dynamics for Advanced Traffic Management and Information Systems, Institute of Transportation Studies, University of California, Irvine

OMG, 2004. Corba, http://www.corba.org/ Last visited 2004-10-15

Papageorgiou, M., J.-M. Blosseville & H. Haj-Salem, 1989. Macroscopic modelling of traffic flow on the Boulevard Peripherique in Paris. Transportation Research B **23B**: 29-47.

Parma, A.d., F. Marchal & Y. Nesterov, 1996. METROPOLIS: a modular system for dynamic traffic simulaton. Transportation Research Record (1607).

Payne, H.J., 1971. Models of freeway traffic and control. Simulation Council **1**: 51-61.

Pindyck, R. & D. Rubinfeld, 1997. Econometric Models and Economic Forecasts. Irwin/McGraw-Hill,

Poschinger, A., R. Kates & J. Meier, 2000. The flow of data in coupled microscopic and macroscopic traffic simulation models. World Congress on Intelligent Transportation Systems, Torino.

PTV, 2003. VISSIM User's Manual 3.7, PTV AG, Karlsruhe

PTV, 2004. PTV simulation -VISSIM, http://www.english.ptv.de/cgi-bin/traffic/traf_vissim.pl Last visited march 2004

Ramming, M.S., 2002. Network Knowledge and Route Choice, PhD thesis, Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge MA, USA

Richards, P.I., 1956. Shockwaves on the Highway. Operations Research **4**: 42-51.

Rumbaugh, J., M. Blaha, W. Premerlani, F. Eddy & W. Lorensen, 1991. Object-oriented modelling and design. Prentice-Hall,

Smith, M., S. Druitt, G. Cameron & D. MacArthur, 1994. Paramics Final Report, University of Edinburgh, Edinburgh

Theil, H., 1961. Economic forecasts and policy. North Holland, Amsterdam

Theil, H., 1966. Applied Economic Forecasting. North Holland Publishing company,

Toledo, T., H. Koutsopoulos, A. Davol, M.E. Ben-Akiva, W. Burghout, I. Andreasson, T. Johansson & C. Lundin, 2003. Calibration and validation of microscopic traffic simulation tools: Stockholm Case Study. Transportation Research Record (1831): 65-75.

TRL, 2004. Contram website, http://www.contram.com/ Last visited march 2004

Trolltech, 2004. QT 3 GUI Libraries, http://www.trolltech.com Last visited march 2004

Underwood, R.T., 1961. Speed, Volume and Density Relationships, Quality and Theory of Traffic Flow, Yale Bureau of Highway Traffic, Newhaven, Connecticut

US-DoT, 1995. TRAF user reference guide, Federal Highway Administration,

Wardrop, J.G., 1952. Some theoretical aspects of road traffic research. Proceedings of the Institute of Civil Engineers **II** (1): 325-378.

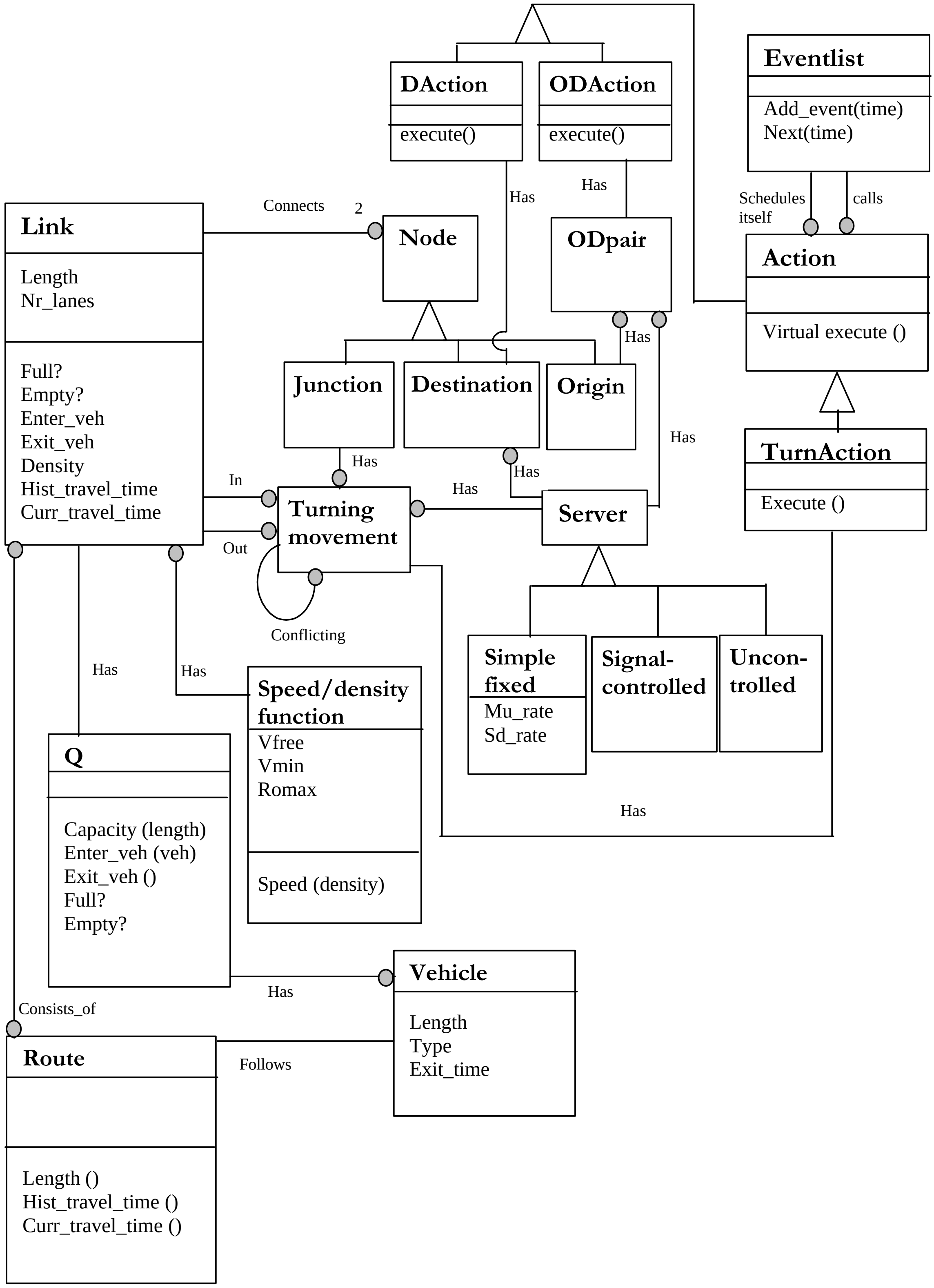
Vliet, D.v. & M. Hall, 1998. The Saturn Users Manual, Institute for Transportation Studies, Leeds

Vogel, K., 2002. What characterizes a "free vehicle" in an urban area. Transportation Research Part F **5** (1): 15-29.

Yang, Q., 1997. A Simulation Laboratory for Evaluation of Dynamic Traffic Management Systems, PhD thesis, ITS Program, Massachusetts Institute of Technology, Cambridge, (MA)

Yang, Q. & H. Slavin, 2002. High Fidelity, Wide Area Traffic Simulation Model. Caliper Corporation, Boston, USA.

Appendix A. Mezzo Object Model (simplified)



Appendix B. Explanation of Object Model elements

This is an introduction into the general notation used in Object Models that make part of the Unified Modelling Language ((Booch, G. 1994), (Rumbaugh, J. *et al.* 1991)). While not yet common in other disciplines, this notation has been the standard for denoting the internal structure of problems in computer science for the past ten years. The main property of this methodology is that the specific entities in a problem are specified explicitly (as Object classes), and the relations between these object classes are specified explicitly. For each relation (called association) the number of objects participating needs to be specified explicitly (one, zero or more, one or more).

In each object class, the data concerning these objects (called attributes) and the operations that can be performed on the entities (called methods) are grouped. Hierarchies of subclasses can be defined to specify common characteristics between subclasses as well as their specific characteristics.

While this notation has not yet become standard in fields outside computer science, the general applicability and its property of forcing the modeller to structure the 'what' (structure of entities involved) before the 'how' (the algorithms applied to solve problems), makes it a strong tool for modellers in any field.

Below follows a legend to reading the object models used in this thesis:

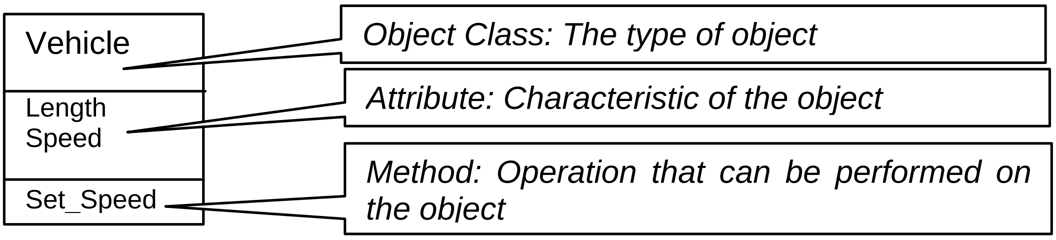

Vehicle
Length
Speed
Set_Speed
Object Class: The type of object
Attribute: Characteristic of the object
Method: Operation that can be performed on the object

Figure 1. Object class with two attributes and one method.

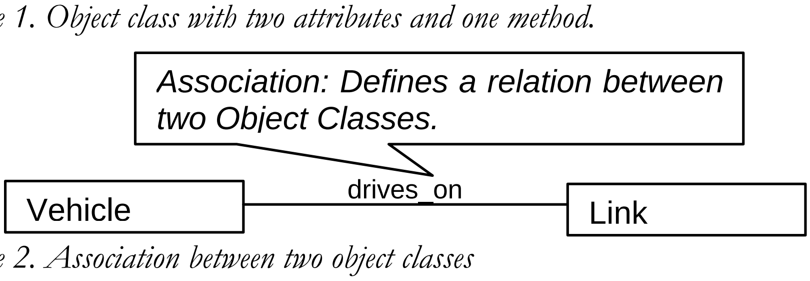
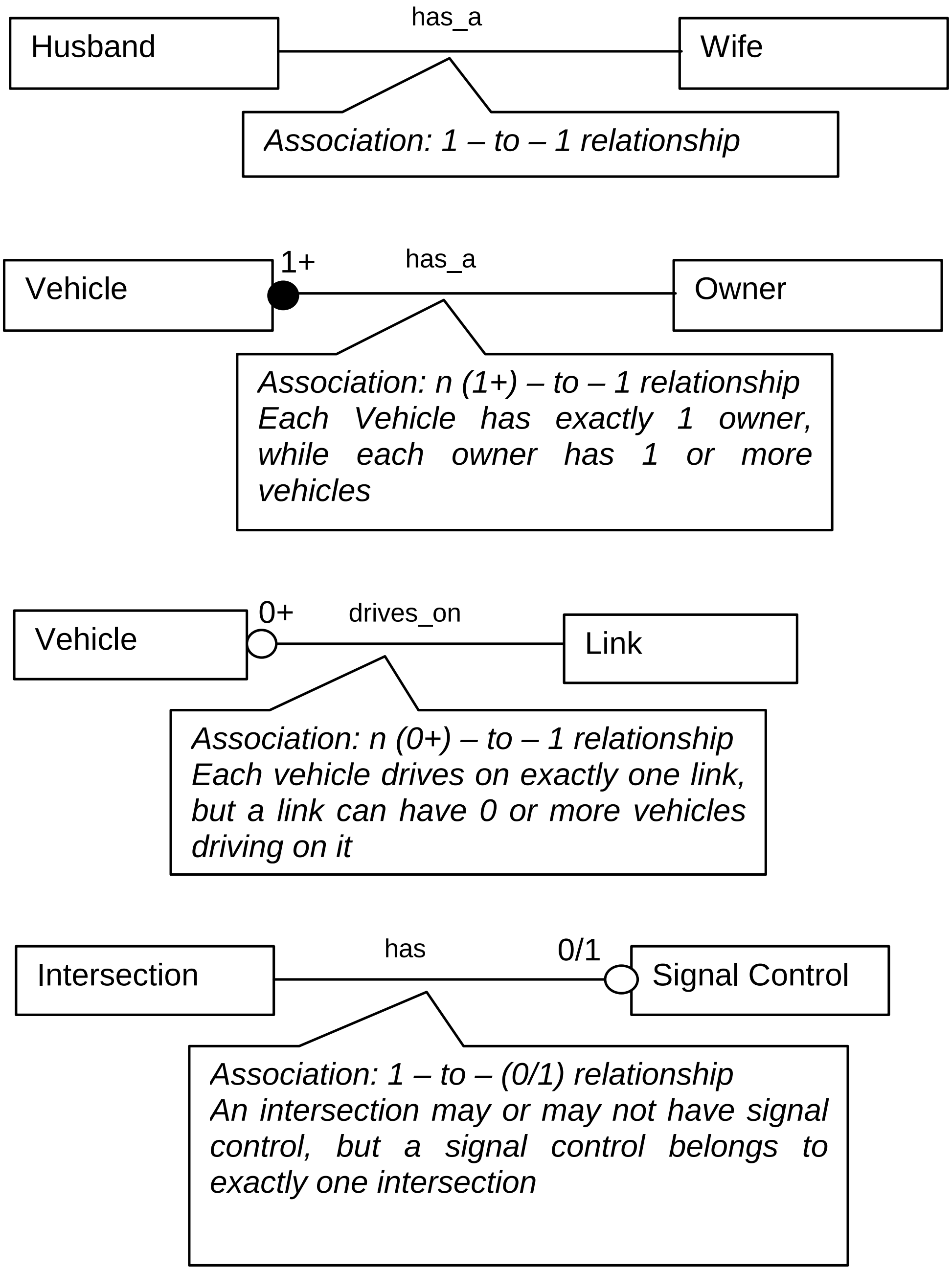


Figure 2. Association between two object classes



Association: 1 – to – 1 relationship

Association: n (1+) – to – 1 relationship Each Vehicle has exactly 1 owner, while each owner has 1 or more vehicles

Association: n (0+) – to – 1 relationship Each vehicle drives on exactly one link, but a link can have 0 or more vehicles driving on it

Association: 1 – to – (0/1) relationship An intersection may or may not have signal control, but a signal control belongs to exactly one intersection

Figure 3. Multiplicity of associations

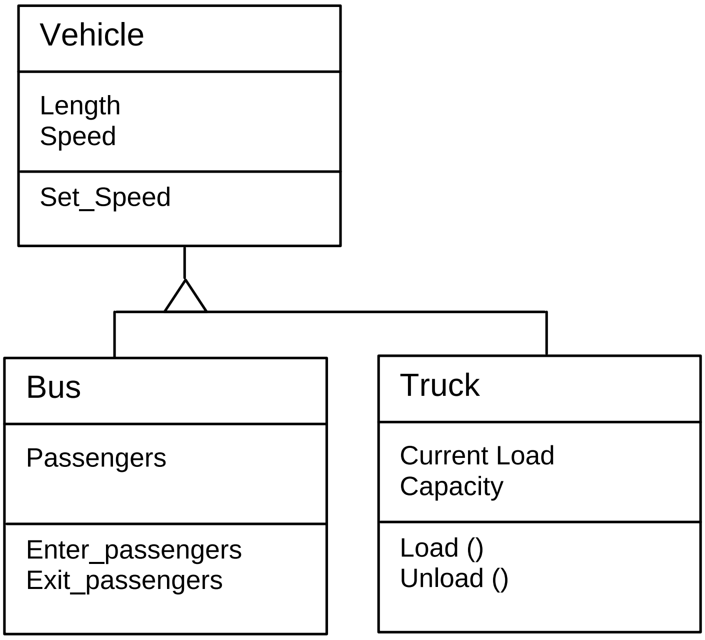


Figure 4. Hierarchy of subclasses. Any subclass (for instance 'Truck') has ('inherits') all the attributes and methods of the superclass (here: 'Vehicle'), and can add its own specific attributes and methods.

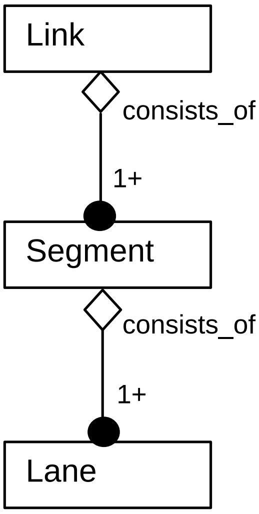


Figure 5. Parts-whole relationship. One link consists of multiple segments, and each segment consists of multiple lanes.

Appendix C. MITSIMLab

C.1 Introduction

As discussed in the first chapter of this thesis, microscopic simulation models are characterised by their modelling the behaviour of individual vehicles (with drivers) that are interacting with other vehicles, the road infrastructure and traffic controls. In this thesis the focus is on the combination of mesoscopic and a microscopic simulation models. In this section the structure and functioning of the microscopic model that is used is described. The model chosen for integration is MITSIMLab, but the structure of most microscopic simulation models is similar. Large parts of the following description are excerpted from (Ben-Akiva, M.E. *et al.* 2000) and (Toledo, T. *et al.* 2003). More detailed information can also be found in ((Yang, Q. 1997) , (Ben-Akiva, M. *et al.* 1997)).

C.2 Generic structure

MITSIMLab is a microscopic simulation laboratory for analysis and design of dynamic traffic management systems, especially for evaluation of advanced traffic control and route guidance systems. MITSIMLab consists of two main components. A Traffic Management System (TMS) that simulates the generation of route guidance and the operation of traffic signals and signs, and a microscopic traffic simulator (MITSIM) that models individual vehicle movements over a network and drivers route decisions and response to traffic controls. In this thesis we are mainly interested in the simulation of the vehicles, which is done by MITSIM.

C.2.1 Network representation

The road network is represented in MITSIM by nodes, links, segments and lanes.

- Nodes: A node is either an intersection of several roadways, or an origin / destination of the traffic network. Each node is represented by type (intersection or origin/destination), identification number and name (optional).
- Links: Links are directed roadway connections between nodes. Each link consists of one or more segments and is characterised by its type (freeway,

ramp, urban street or tunnel), starting node, end node and the segments that it contains. Links are connected via nodes.

- Segments: Segments are parts of a link with uniform (geometric) characteristics such as number of lanes, grade, curvature, design speed, etc. Each segment is characterised by its speed limit, design speed, grade, geometry and the lanes it consists of.
- Lanes: The lowest level of network elements in MITSIM are lanes. For each lane the lane-change and lane-use rules are specified. The lane change rules determine which lane changes are permitted (to left, to right, both, none) and the lane-use rules determine which vehicle types are allowed to use the lane (such as bus-lanes for buses only).
- Lane-connections: The elements of the network are connected via lane-connections. For each lane of a link in an intersection (node) the connections to lanes of outgoing links are explicitly coded. Also the lanes of segments that make up a link are connected by lane-connections.

C.2.2 Traffic surveillance & control

In this thesis the traffic surveillance and control aspects of Mitsim are not directly relevant to the model integration issues and therefore the reader is referred to (Ben-Akiva, M. *et al.* 1997), (Yang, Q. 1997) for a more detailed description. It suffices to mention that a large range of sensor types and functions as well as fixed and adaptive traffic control systems can be modelled.

C.2.3 Incidents

Incidents can be activated and cleared at any particular time and location in the network, and are specified in a specific file. Spontaneous accidents are not simulated. Each incident may affect one or more lanes. The information for incidents includes visibility, number of lanes affected, segment code, and position. For each lane the severity of the incident and its length, maximum (passing) speed, start time and expected duration are specified.

C.3 Behavioural Models

The behavioural models are the heart of a microscopic simulation model. It is these models that determine the behaviour of the simulated drivers/vehicles. A number of different behavioural models are usually identified. The car-following model captures the acceleration/deceleration behaviour of vehicles in response to vehicles in front. The lane-changing model captures when and how drivers change lanes, for instance when overtaking, or to continue along a certain path. A route-choice model captures the choices and changes in routes to get to the designated destinations. Intersections consist of a more complex set of interactions, and the behaviour in terms of accepting gaps in opposing flows, adaptation of speed to turnings etc. are modelled separately.

C.3.1 General acceleration

A vehicle accelerates (decelerates) in order to react to the vehicles ahead, perform a lane changing or merging manoeuvre, or respond to events (e.g. red signals and incidents). The most constraining of these situations determines the acceleration (deceleration) rate to be implemented in the next simulation cycle. Depending on the degree of interaction with the vehicle ahead, the subject vehicle can be in free-flowing, car-following, or emergency regime. The degree of interaction is determined by the time headway between the two vehicles. Excerpt from (Ben-Akiva, M.E. *et al.* 2000), (Toledo, T. *et al.* 2003)

Free-flowing regime: In the free-flowing regime, the vehicle accelerates if its current speed is different from the driver's *desired speed*. The acceleration applied by a driver in this regime is assumed to have the following functional form:

$$a_n^{ff}(t) = l^{ff}\left[V_n^*(t - t_n) - V_n(t - t_n)\right] + e_n^{ff}(t) \qquad (1)$$

Where,

$a_n^{ff}(t)$: acceleration of driver n at time t

l^{ff}: parameter

$V_n^*(t)$: desired speed of the driver

$V_n(t)$: speed of subject vehicle at time t

$e^{ff}(t)$: error term

t_n reaction delay of driver n

Car-following: The car-following model is used for calculating a vehicle's acceleration or deceleration rate in various cases such as:

- Car-following relationship with the leading vehicle;
- Competition with other vehicles if two or more lanes merge into a single downstream lane;
- Yielding to another vehicle shifting into the same lane.

The car following model is a generalization of the non-linear GM model (Gazis, D.C. *et al.* 1961). Furthermore, the parameters of the model can be different for acceleration and deceleration situations. The general structure of the model is shown in Figure 1:

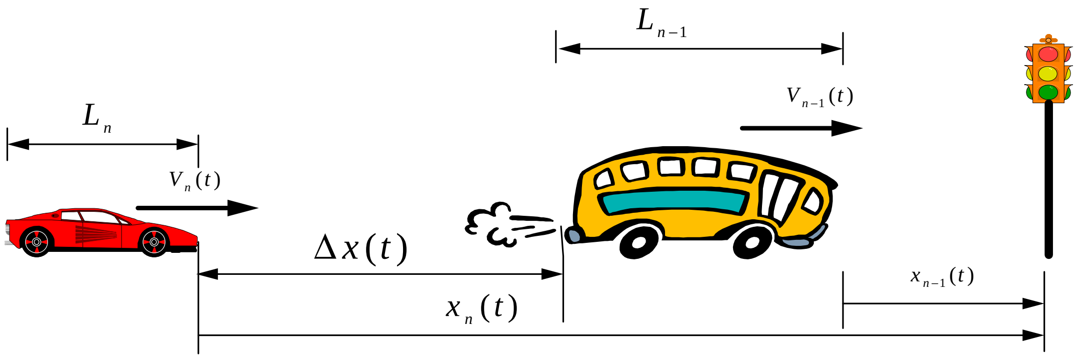


Figure 1. Car-following model

The car-following model can be expressed mathematically as:

$$a_n^{cf}(t)=a\frac{V_n(t-T)^a}{[\Delta x(t-T)]^b}k^d[V_{n-1}(t-T)-V_n(t-T)]^g+e_n^{cf}(t) \qquad (2)$$

Where,

$a_n^{cf}(t)$ = acceleration of vehicle n at time t;

$Dx(t)$ = gap between vehicles at time t;

k = density of traffic in the vicinity of the vehicle;

$V_n(t)$ = speed of vehicle n at time t;

$\alpha, \beta, \gamma, \delta$ = parameters

$e_n^{cf}(t)$ = error term

t_n = reaction delay of driver n

Emergency regime*:* In the emergency regime, the vehicle uses an appropriate deceleration rate to avoid collision. The deceleration rate depends on the state of the front and subject vehicles. In all cases though, the applied rate guarantees that the subject vehicle will always decelerate to extend the headway to a safe range.

C.3.2 Lane changing

In Mitsim, vehicles may change lane for two reasons: either because it is necessary (mandatory) (for instance to continue their trip in the case of a lane-drop, or to take an exit-ramp), or because they desire to do so (discretionary) (for instance when they are driving slower than they want to and can drive faster in another lane). The lane-changing process consists of three steps. First it is determined if a lane change is necessary or desired, then a target lane is selected, and finally a suitably large gap between vehicles on the target lane is selected.

Gap Acceptance

Drivers have minimum acceptable lead and lag gap lengths (lead and lag critical gaps) when they consider lane-changes. These critical gaps vary not only among different individuals, but also for a given individual under different traffic conditions. The value of the critical gap is a function of traffic density and distance to the point by which the driver has to complete a mandatory lane change.

Forced Merging (Nosing)

When a vehicle is in mandatory state it may push into the desired lane ('nosing') with a probability dependent on the distance to the exit, number of lane changes necessary and the time the vehicle has been waiting for a lane change. The lag vehicle (i.e. the vehicle affected by the nosing) will automatically yield (courtesy yielding).

C.3.3 Intersection gap acceptance

The acceptance of gaps at intersections is more complicated than when changing lanes. In MITSIMLab, the drivers will first identify possible conflicts by predicting the paths of drivers on conflicting movements, extrapolating their positions using their current speeds. Using these conflict points, drivers try to estimate different gaps that will exist by the time they will themselves arrive at these points. The drivers will then adapt their speeds to maximize the size of the gaps, and keep on doing so during their approach of the intersection. Their speed is also adapted to the maximum speed at which they can perform the turning movement. Whether a gap is accepted or rejected depends on whether it is bigger than the critical gap for that driver and that turning movement. Critical gaps are distributed randomly over the drivers.

C.3.4 Route choice

In Mitsim there are two main route choice models: Pre-trip path choices with en-route switching, or a link-based route choice model that lets drivers choose their route one link at a time. The link-based model is not used in this work, since the path-based with en-route switching is considered more appropriate.
Mitsim uses two kinds of travel time information to base the route choice on: historical or 'experienced' link travel times (time-dependent) for pre-trip route choice, and real-time travel time information for en-route diversions. Real-time travel times are provided by in-vehicle devices (radio, in-vehicle navigation systems) or Variable Message Signs (VMS) along the road.

The path based route choice uses C-Logit ((Ramming, M.S. 2002), (Cascetta, E. *et al.* 1996)), which is a variation of the Logit model that accounts for the effect of overlap between alternative routes. In short, if there are two or more routes from a given origin to a given destination, the drivers choose between them based on their respective utilities. The utility function can be defined differently for different users and can depend on more factors than travel time, such as number of traffic lights, type of roads, tolls.
In most cases, a vehicle will stick to the path chosen at the start of the trip, but in some cases a driver may want to change (for instance when faced with excessive delays on the chosen path, or when he receives information about delays further on). In this case the driver makes a new choice between the remainder of his chosen path and the available alternatives.

Path awareness

In relation to the route that a driver wants to follow, the necessary lane-changes (mandatory) need to be made at the right moment. The drivers have a distance (stochastically distributed among drivers) they can look ahead along their route, to anticipate future necessary lane-changes.