

University of Louisville

ThinkIR: The University of Louisville's Institutional Repository

Electronic Theses and Dissertations

12-2010

Development of a simulation and evaluation environment for a traffic flow analysis system.

Peter Bernhardt 1973-
University of Louisville

Follow this and additional works at: <https://ir.library.louisville.edu/etd>

Recommended Citation

Bernhardt, Peter 1973-, "Development of a simulation and evaluation environment for a traffic flow analysis system." (2010). *Electronic Theses and Dissertations*. Paper 103.
<https://doi.org/10.18297/etd/103>

This Master's Thesis is brought to you for free and open access by ThinkIR: The University of Louisville's Institutional Repository. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of ThinkIR: The University of Louisville's Institutional Repository. This title appears here courtesy of the author, who has retained all other copyrights. For more information, please contact thinkir@louisville.edu.

DEVELOPMENT OF A SIMULATION AND EVALUATION ENVIRONMENT
FOR A TRAFFIC FLOW ANALYSIS SYSTEM

By

Peter Bernhardt

Diplom-Wirtschaftsinformatiker (FH), Wildau, 2000

A Thesis

Submitted to the Faculty of the
Speed School of the University of Louisville
In Partial Fulfillment of the Requirements
for the Degree of

Master of Science

Department of Industrial Engineering
University of Louisville
Louisville, Kentucky

December 2010

Copyright 2010 by Peter Bernhardt

All rights reserved

DEVELOPMENT OF A SIMULATION AND EVALUATION ENVIRONMENT
FOR A TRAFFIC FLOW ANALYSIS SYSTEM

By

Peter Bernhardt

A Thesis approved on

July 14, 2010

by the following Thesis Committee:

Professor Dr. William E. Biles, Thesis Director

Professor Dr. Gerald W. Evans

Dr. C. Timothy Hardin

DEDICATION

This thesis is dedicated to my parents
who always believed in me and gave me the freedom
to pursue my dreams.

Additionally I want to dedicate this thesis
to my daughter who is my
never ending source of
inspiration.

ABSTRACT
DEVELOPMENT OF A SIMULATION AND EVALUATION ENVIRONMENT
FOR A TRAFFIC FLOW ANALYSIS SYSTEM

Peter Bernhardt

July 14, 2010

A system for analysis of the traffic flow on public streets and highways through the use of Floating Car Data (FCD) relies completely on the number of simultaneously contributing vehicles, a fact that is no barrier for the phases of conception and development but poses a serious issue for the testing of such a system. Especially for smaller institutions or companies where the ability and resources to field the required number of participants is not given which in turn leads to the need for computational support to substitute the use of real vehicles by simulation.

This thesis focuses on the task of the design and development of a simulation and evaluation environment for a pre-developed Traffic Flow Analysis System. The objective of this environment is to simulate the behavior of real vehicles on the existing street network including their most relevant characteristics for the purpose of congestion recognition. It is shown how simulation methods can be effectively used to create such an environment while using mathematical methods to model the characteristics of the participating system parts (vehicles) as well as the environmental influence on the external communication components (GPS, Radio).

TABLE OF CONTENTS

	PAGE
1.INTRODUCTION	1
1.1 Motivation.....	1
2.TERMINOLOGY	4
2.1 Acronyms	4
2.2 Introduction to the Traffic Flow Analysis System	7
3.SIMULATION DESIGN	11
3.1 Requirements.....	12
3.2 Simulated Vehicle Characteristics.....	14
3.2.1 Identification and Classification	14
3.2.2 Mathematical Modeling	17
3.3 Simulation Evaluation	20
3.4 Evaluation of Existing Simulation Systems	21
3.4.1 VISSIM	22
3.4.2 TSIS-CORSIM	24
3.4.3 Conclusion	25
3.5 Simulation Methods	26
4.SOFTWARE DESIGN	30
4.1 Design Approach.....	30
4.2 The Data Model.....	36
4.3 Program Language and Software Development Environment.....	38
5.SYSTEM INTEGRATION.....	39
5.1 GIS Applications	39
5.2 Routing.....	43
5.3 Simulated External Systems.....	44
6.IMPLEMENTATION.....	48
7.EVALUATION AND RESULTS.....	51
8.DISCUSSION	53
9.CONCLUSIONS.....	55
9.1 System Capabilities	55
9.2 System Limitations	55
10.FUTURE WORK.....	58
REFERENCES	59
CURRICULUM VITAE.....	64

LIST OF TABLES

Table 1: Selection of the relevant vehicle properties.....	16
------------------------------------------------------------	----

LIST OF FIGURES

Figure 1: Usage scenario for TFAS	8
Figure 2: Selection of vehicle properties	15
Figure 3: Acceleration diagram for the Volkswagen VW1200 (Reuter, 1998).....	18
Figure 4: 3D view of an intersection simulation with VISSIM (PTV AG, 2010).....	23
Figure 5: TRAFVU the visualization module of TSIS	24
Figure 6: Cellular Automation Model snapshot.....	27
Figure 7: Schematic overview of the simulation system	31
Figure 8: Vehicle properties (with translation).....	33
Figure 9: Detail of the Map Display showing a vehicle approaching a blocked street section	35
Figure 10: Detail of a road map for navigation purposes (www.navteq.de).....	36
Figure 11: Main window of the simulation application with two vehicles.....	39
Figure 12: Detail map with two vehicles and their routes	40
Figure 13: Two blocked street sections and their detail information (street name, ID).....	41
Figure 14: Detail map displaying a traffic congestion including the affected street section	42
Figure 15: Simplification of a road network as graph (Broadbent, 2008).....	43
Figure 16: Calculation of the current course using regression	46

1. INTRODUCTION

1.1 Motivation

The complexity of today's systems is far greater than that of the systems of the past. Especially with the intensive introduction of information systems into nearly all aspects of work is it more often than not the case that it is difficult for humans to fully understand or evaluate a system on their own. The main reasons for this are the diversity of cases to set up and evaluate as well as the amount of data to process and verify. This problem becomes even more important and obvious if the amount of participants in a system starts to scale.

Since about a decade there are efforts all over the world to supervise and analyze the traffic flow on public streets and highways more closely and especially more actively than only through stationary tracking devices. The work with Floating Car Data (FCD) created and transmitted from the driving vehicles themselves was and still is subject to intensive research projects, i.e. FleetNet¹, Car2Car. One of the main characteristics of nearly all this projects is the existence of central servers to singlehandedly process the FCD messages and transmit the results back to interested parties for further use, i.e. to navigation systems for rerouting purposes. The University of Applied Sciences Wildau in cooperation with industry partners deviated from this path and developed a system that only relies on the direct communication between the participating vehicles. The main objective for this path lies in the goal to avoid the necessity of a central infrastructure, the

implied dependence on a major operator of this infrastructure as well as in the sought independence of wireless communication technologies like UMTS or WLAN. This system relies only on the contributing vehicles and not on any existing infrastructure making it possible to use in developing countries or environments with changing infrastructural conditions like large construction sites.

The basic success factor for the operation or even the evaluation of a system like this or any system relying on FCD is the number of simultaneously contributing vehicles. A fact that is no significant barrier for large enterprises or well-funded governmental research projects but makes it very difficult for smaller institutions or companies where the ability to field the required number of vehicles is not given. This leads to the need for computational support to substitute the use of real vehicles by simulation.

This thesis focuses on the task of the design and development of a simulation and evaluation environment for the developed Traffic Flow Analysis System. The objective of this environment is to simulate the behavior of real vehicles on the existing street network including their most relevant characteristics for the planned purpose. And not only to simulate participating vehicles and then using the computed data to transmit them to the Congestion Analyzer, the simulation environment should also incorporate vehicle-side parts of the developed Analysis System to additionally evaluate their functionality as well as their performance capabilities.

The aim of this thesis is to show how simulation methods can be effectively used to create a simulation and evaluation environment for an existing system while using mathematical methods to model the characteristics of the participating system parts

(vehicles) as well as the environmental influence on the external communication components (GPS, Radio).

2. TERMINOLOGY

2.1 Acronyms

- FCD Floating Car Data means data that is generated from a vehicle out of its current participation on the road traffic. A record usually consists at least of the time stamp and the current coordinates. By transmitting floating car data cars essentially become mobile sensors. (Schoder, 2009)
- GIS The term ‘Geographic Information Systems’ or ‘Geographical Information Systems’ describes information systems that are responsible (in full or partly) for the recording, editing, organizing, analyzing and presenting of geographic data for common use or for highly specialized tasks. GIS can subsume the required hardware, operating software, data as well as applications. (Bartelme, 2005)
- GPS “The Global Positioning System is a constellation of over 24 U.S. Government satellites providing PNT services to civilian and military users on a continuous, worldwide basis -- free of direct user charges. The system provides highly accurate location and time information to anyone equipped with a GPS receiver. GPS provides a precise, common location and time reference to an unlimited number of people in all weather, day and night, anywhere in the world.” (PNT, 2009)

- TFAS The Traffic Flow Analysis System analyzes the traffic flow on public streets and highways to recognize traffic congestions - for e.g. route optimizations - by the use of FCD created and transmitted from and to the driving vehicles through direct communication between them without the need for a central infrastructure. It was a research project of the University of Applied Sciences Wildau in cooperation with industry partners in the years 2008 to 2010.
- TMC “Traffic Message Channel (TMC) is a technology for delivering traffic and travel information to drivers. It is typically digitally coded using the FM-RDS system on conventional FM radio broadcasts. It can also be transmitted on DAB or satellite radio. It allows silent delivery of high quality accurate, timely and relevant information, in the language chosen by the user and without interrupting normal services. Services, both public and commercial, are now operational in many countries worldwide.”(Wikipedia, 2010)
- UMTS The Universal Mobile Telecommunications System is a mobile telecommunications standard of the third generation (3G) providing much higher data transfer rates (up to 14.4 Mbps with HSDPA, otherwise max. 384 Kbit/s). It was introduced in 1999 and is now widely used all over the world. (Benkner, 2007)

WLAN A wireless local area network is a local area network where the participating clients are wirelessly connected within a limited range. The communication standard IEEE 802.11 defines the used communication methods. It was introduced in 1997 and is widely adopted. The abbreviations WLAN and Wi-Fi can be used synonymously. The initial version 802.11b of the standard provided a maximum data transfer rate of 11 Mbps. The most current fielded version is 802.11g with a theoretical speed of 54 Mbps. (Rech, 2008)

2.2 Introduction to the Traffic Flow Analysis System

The Traffic Flow Analysis System (TFAS) is a research project of the University of Applied Sciences Wildau² with the objective to develop a system to supervise and analyze the traffic flow on public streets and highways. It was created in cooperation with medium sized industry partners out of the region Berlin/Brandenburg.

The field of traffic analysis has already a seen and still sees a number of large research projects funded by the German federal government. These projects mainly follow the approach to develop and establish a complex and sophisticated infrastructure to receive the Floating Car Data (FCD) from the vehicles, to analysis the collected FCD messages and finally to transfer the determined traffic information back to the users in the vehicles. The reason for the University of Applied Sciences Wildau and the government of the federal state of Brandenburg to undertake and to fund another project in this area of research was the proposal to developed a system that only relies on the direct communication between the participating vehicles and hence the goal to avoid the necessity of a central infrastructure. This also removes the implied dependence on a major operator of this infrastructure as a significant cost factor. Additionally it was also the idea to seek the independence of costly wireless communication technologies like UMTS or WLAN cutting another important factor out of the prerequisites for such a system. TFAS completely relies only on the contributing vehicles and not on any existing infrastructure and makes it therefore possible to use in developing countries or environments with changing infrastructural conditions like large construction sites.

A typical usage scenario for TFAS is schematically shown in Figure 1. The following requirements have to be met: a certain number of TFAS equipped vehicles, a good GPS reception and sufficient conditions for the radio technology (e.g. no large

gorges). The terrain doesn't have to be flat but should also not be mountainous. The vehicles are driving on roads or tracks registered in the road maps available for all participants and the navigation systems used. The on-board units of all vehicles are constantly using the GPS signal to evaluate the current situation of own vehicle. The current coordinates, the current speed, the current course as well as the average speed (over a certain time) are continuously determined. As long as the vehicle is assumed to be moving normally nothing happens.

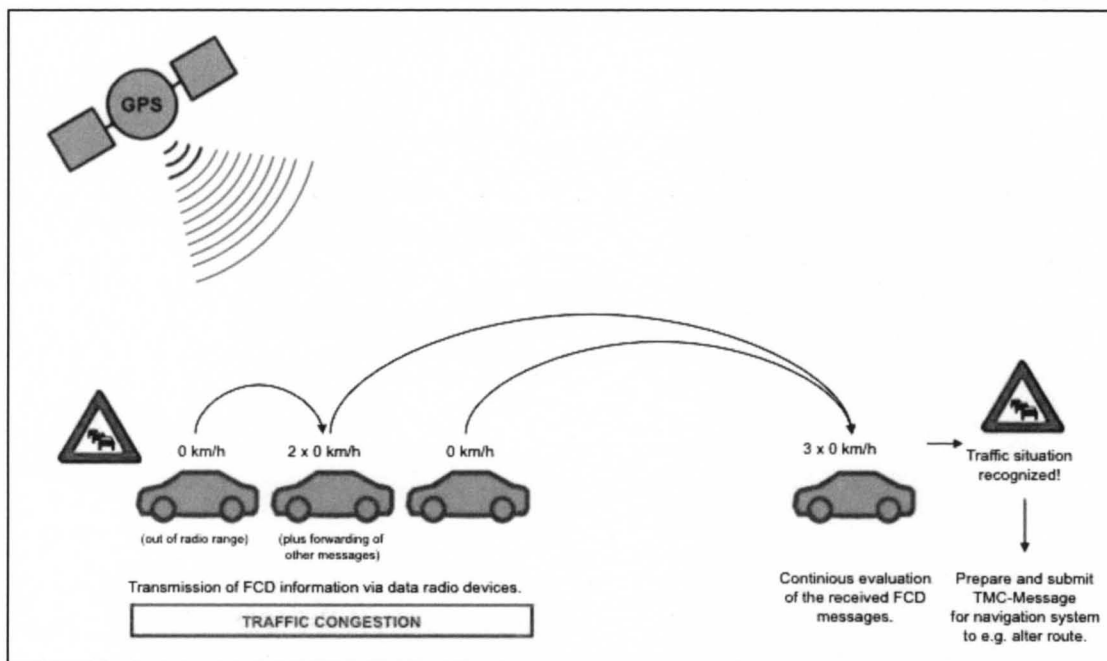


Figure 1: Usage scenario for TFAS

But when the vehicle is slowed or stopped by traffic congestion the average speed falls below a certain limit the on-board units are starting to transmit FCD messages via radio. The average speed is used as indicator because a vehicle should not evaluate a stop at a red light, at a junction or a halt for a left turn as congestion. A vehicle does not control the reception of its messages. There are just sent out.

A sent FCD message contains all necessary information about the current state of the vehicle as part of TFAS. This includes a timestamp in seconds and a vehicle indicator ID to differentiate the vehicles which is randomly renewed every 15 minutes for privacy purposes e.g. to block the creating of travel profiles. Additionally contained are the current speed in Kilometers per hour, the average speed in Kilometers per hours, the current course (driving direction) in degrees, and eventually the current coordinates in GPS format. A special encoding was developed to reduce the size of an FCD message for sending and to further include error-checking information that a recipient can use to validate a message.

Another vehicle (right side in Figure 1) in the area can then receive all presently transmitted messages and evaluate them to generate an image of the current traffic situation around it. A single vehicle that stopped for whatever reason and started transmitting does not constitute a traffic congestion. The evaluation algorithm itself is very complex and requires significant computing capacity, but basically all messages are filtered to improve performance, mapped onto the road network and if there are a number of vehicles (left three cars in Figure 1) reporting from the same road (but possibly multiple successional sections) and driving direction a traffic congestion is recognized. The true reason for this congestion cannot be found but this is not necessary for an alternation of the route by the navigation software. This process is started by the transmission of the congestion information encoded into a package on TMC messages to the navigation software.

A system completely relying on the participating vehicles always has the total number of simultaneously contributing vehicles as fundamental success factor. If there are not enough vehicles in the area the situation on the roads cannot be evaluated completely or possibly at all. Therefore a high number of fielded devices have to be

ensured. The solution for this in TFAS was the development of three types of devices with different functionalities and especially largely different costs:

Module 1 Variant A is the simplest device that can only send but not receive FCD messages. This device was specified for parties that have no interest in the resulting traffic evaluation but want to contribute like public transport organizations or taxi companies.

Module 1 Variant B is an improved Version A that can receive FCD messages too but only to forward them further thereby increasing the effective radio range of the system. This variant does not have the ability to evaluate FCD messages.

Module 2 has the full capabilities for contributing as well as evaluating all received FCD messages. This is the only variant that connects to a navigation system.

While the devices of Module 1 can effortlessly run on off-the-shelf mobile devices with embedded processors like PDAs does Module 2 need the computing capacity of a notebook. But the use of PDAs did reduce the device cost to a high degree for these modules.

Nevertheless did the need for a simulation of participating vehicles arose due to the fact that the project funding does not allow the fielding of the larger number of devices needed to really evaluate the Congestion Analyzer.

3. SIMULATION DESIGN

The objective in the work for this thesis is to simulate vehicles as participants in TFAS to evaluate the performance of the Congestion Analyzer in recognizing traffic congestions. A simulation as tool for the generation of the input data for the Congestion Analyzer is a very favorable approach due to the advantages of (a certain) repeatability and control over the full traffic situation without the need to field this number of systems in real vehicles and therefore saving the costs to do so.

A simulation is described as “an experiment performed on a model” (Korn and Wait (1978) after Cellier (1991), page 6) followed by the definition of an experiment as „the process of extracting data from a system by excerpting it through its inputs“ (Cellier, 1991, page 4). The model selection and the selection of the simulation engine are therefore the most important parts of a simulation. This chapter first defines the vehicles as core part of the model as the maps are pre-defined, explains how the simulation results were evaluated and finally describes the process of selecting the right simulation engine.

3.1 Requirements

For the process of defining a good model and the selection of the right engine it is necessary to detail the requirements and possible assumptions for the simulation to be able to correctly fulfill this task:

- a) The vehicles have to move on a real road network. The Congestion Analyzer currently uses the road network of Berlin/Brandenburg as basis for its recognition of roads, successional sections and driving restrictions. Therefore must the simulation be able use the same maps or a subset thereof.
- b) For the analysis of the traffic situation is a minimum number of participating vehicles necessary. The simulation must be able to simulate multiple dozen of vehicles at the same time.
- c) The simulation is less about a single vehicle as about the activities of an amount of vehicles. It is not believed that the impacts of the details of vehicle movement are decisive for the work of the Congestion Analyzer. Nevertheless should the state of a single vehicle always be accessible. Environmental or road conditions for the vehicle movement can be neglected. The route of the vehicles can be chosen randomly. Vehicles can pass each other without restrictions.
- d) The simulation is about the generation of FCD messages not about the road network or the external behavior of the vehicles. The FCD generation should be an intrinsic part of the simulation not a separate task.

- e) The generation of FCD messages depends on the average speed of the vehicle therefore should the movement of the vehicle on the road network be controllable at all times.
- f) The simulation should display its state graphically this includes the result of the Congestion Analyzer.
- g) The operator of a simulation should be able to pause and resume the simulation at any time. It must be possible to save and load a simulation session to a file.
- h) The simulation software should be able to integrate pre-developed software components of TFAS into the simulation. The objective hereby is to create and transmit the FCD messages of each vehicle during the simulation like they would in real vehicles using the real external systems (GPS, Radio Modem). The different environmental conditions in regard of the transmission quality of the Radio Modem should be simulated as well despite the fact that this prevents a full repeatability of the simulation.
- i) The simulation software should be able to directly use the web service connection to the Congestion Analyzer.
- j) For performance reasons it would be possible for the simulation to assume that all the simulated vehicle are equipped with Module 1 Variant A and are therefore just data producers. The simulation of one Module 2 would be sufficient as the task of nearby vehicles equipped with Module 2 would be essentially similar.

These requirements are without any specific order. Common requirements like good usability and high configurability are implicitly set. There are no hard requirements

concerning the performance of the simulation as the Congestion Analyzer and other parts of TFAS are quality wise in a research stadium and not optimized for performance. Additionally should the minimum number of vehicles considered necessary pose no problem for the capabilities of up-to-date computers.

3.2 Simulated Vehicle Characteristics

The movement of vehicles on a road network is the outer objective of this simulation. As the inner processes for each vehicle are clearly defined is the outer behavior to be determined.

3.2.1 Identification and Classification

Many different aspects determine the behavior of vehicles in a street network in the real world. The main problem for a simulation of this behavior in the first place is to decide which of these aspects are relevant to the simulation purpose. A simulation has to be abstract to be feasible, so this selection cannot be done easily as it has a significant effect on the amount of data to be handled as well as the overall performance of the simulation.

To make this decision it is necessary to collect the properties of vehicles moving on a street network respecting that the road as well as the environmental conditions can be neglected. The mind map in Figure 2 lists common properties of vehicles without any order.

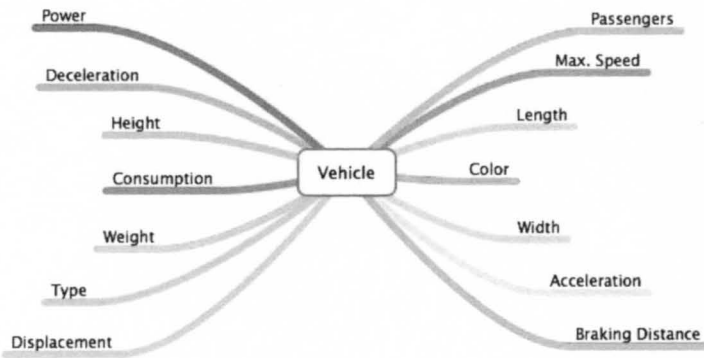


Figure 2: Selection of vehicle properties

There are still values that are not mentioned like maximum curve speed that are too special to be relevant. The first step of the selection process is the classification of these properties according to their relevance regarding the movement on a real street. This has been done in Table 1. There are some properties that are completely without any relevance like the color or the consumption. Properties like the number of passengers and the type of the vehicle can be ignored in this case because of their little influence for the simulation purpose. Factors like width and height that relevant in reality can be entirely ignored in the simulation. This leaves factors like power, displacement and weight that are important but can be subsumed by the factors acceleration and deceleration, as they have no meaning in the simulation. Finally there are additionally to these two the factors maximum speed and length of the vehicle. Both of these factors are very important in reality and for the simulation purpose as they have a direct effect on the behavior of the vehicle. A vehicle cannot move faster than its maximum speed despite speed limit of the street may allow so and the length of the vehicles waiting determines the length of traffic congestions.

Factor	Relevance		Selection
	Reality	Simulation	
Power	High	None	
Deceleration	High	Low	x
Height	Low	None	
Consumption	None	None	
Weight	High	None	
Type	None	None	
Displacement	High	None	
Passengers	None	None	
Max. Speed	High	High	x
Length	High	High	x
Color	None	None	
Width	Low	None	
Acceleration	High	Low	x
Braking Distance	High	None	

Table 1: Selection of the relevant vehicle properties

It is therefore definitely necessary to implement these in the simulated vehicles characteristics or otherwise the simulation would create significantly different results as reality.

The factors acceleration and deceleration are also important as they have an impact on the simulation on multiple occasions. They are relevant when a vehicle is starting or stopping e.g. on the beginning or after a roadblock. And they can be applied when the speed limit increases or decreases and a vehicle has to accelerate or decelerate. Therefore they should be implemented too.

The results of the selection process are color-coded shown in Table 1.

3.2.2 Mathematical Modeling

After the relevant characteristics – maximum speed, length, acceleration and deceleration - were identified it is necessary to develop mathematical models to correctly include them into the simulation.

Maximum Speed

The vehicle characteristic of its maximum possible driving speed is a single value and can therefore be modeled easily. It should be modeled as a low-pass filter allowing the vehicle all speeds below the set limit and stopping every acceleration attempt at the set limit.

Let v_{\max} be the maximum allowed speed and v the current speed, then

$$v = \{x \in R \mid 0 \leq x \leq v_{\max}\}$$

The unit used for maximum speed as well as current speed of a vehicle would be Kilometer per hour (km/h).

Length

The length of a vehicle is a set and fixed value. In reality vehicle do have many different lengths depending on the model, the maker and the year of manufacture. In the simulation it would be feasible to support different length values for different groups/types of vehicles if used.

Let l be the length of a vehicle, then

$$l = \{x \in R \mid 0 < x \leq +\infty\}$$

The unit used for the length of a vehicle would be Meter.

Acceleration

The acceleration as vehicle characteristic is more complicated than the two before. The rate of acceleration of a vehicle depends in the real world on a number of conditions. These are for instance the performance of the engine, the available torque at a certain number of revolutions, the gear ratio and so on (Braess & Seiffert, 2007).

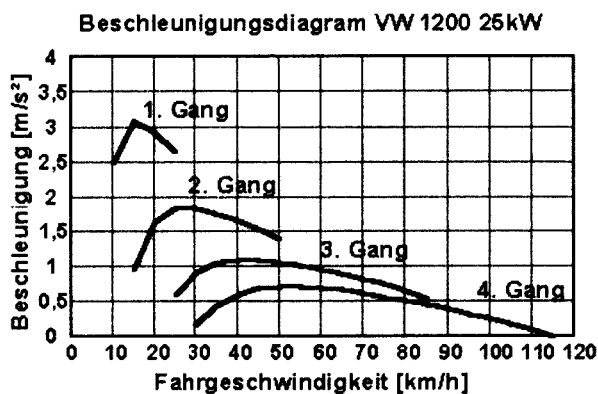


Figure 3: Acceleration diagram for the Volkswagen VW1200 (Reuter, 1998)

The acceleration diagram of an old and simple car like the VW1200 (the VW Beetle) in Figure 3 shows that the possible rate of acceleration also strongly depends on the selected gear, an attribute that is completely oblivious for the simulated vehicle. A simulation of the real behavior of an accelerating vehicle would therefore be very complex. Additionally it wouldn't be worth it for this simulation purpose as this simulation is less about the correct micro-behavior of each vehicle as it is about the movement of the entirety of vehicles (see Requirement c) in Chapter 3.1).

For this simulation purpose it would be sufficient to simulate the acceleration as a linear function of speed over time. This representation would simplify the acceleration to an applicable factor.

Let a_{acc} be the acceleration, then

$$a_{acc} = \{x \in R \mid 0 \leq x < +\infty\}$$

Then let v_s be the initial speed and the acceleration timeframe t , then the final speed v_e can be calculated using

$$v_e = a_{acc}t + v_s$$

The unit for the acceleration factor would be Meter per square second.

Deceleration

The deceleration is very comparable to the acceleration. In reality it also depends on a large number of inner and outer aspects of the vehicle and its environment (Braess & Seiffert, 2007).

For the simulation purpose it would be again sufficient to simulate the deceleration (the negative acceleration) as a linear function of speed over time to simplify it to an applicable factor.

Let a_{dec} be the deceleration, then

$$a_{dec} = \{x \in R \mid 0 \leq x < +\infty\}$$

Then let v_s be the initial speed and the deceleration timeframe t , then the final speed v_e can be calculated using

$$v_e = -a_{dec}t + v_s$$

The unit for the deceleration factor would be Meter per square second.

3.3 Simulation Evaluation

The evaluation of a simulation enfolds two aspects: verification and validation (Pegden, Shannon, & Sadowski, 1995, p.129 after Feinstein & Cannon, 2002, p. 425).

“**Verification** is the process of assessing that a model is operating as intended.

Validation is the process of assessing that the conclusions reached from a simulation are similar to those reached in the real-world system being modeled.” (Feinstein & Cannon, 2002)

Feinstein & Cannon also say “the process of verification involves debugging the model by isolating and eliminating as many errors as possible. This can be done by using internal debuggers of the simulation software, viewing output reports, evaluating step-by-step traces of a simulation run, and involving individuals who can evaluate the simulation. Many verification errors are simple problems of software debugging. Others involve fixing design errors, where the basic equations interact in unanticipated ways or the embedded response functions become invalid for extreme values.” (Feinstein & Cannon, 2002) This statement cannot be more than emphasized.

To find out however if the entirety of the FCD messages as the final and decisive output generated by the simulation is valid is a complex task. In the case of this simulation purpose and the set project limits is the automation of this task not affordable. The automation of the validity check would be required for simulations with a high number of runs and / or where the output of the simulation is the result of the project. But both of these requirements are not applicable in this case. Therefore it is and was sufficient to check the validity of simulation result (the FCD messages) manually. Stopping a simulation run, storing its state and the “transmitted” FCD messages, is the first step in this process. The FCD messages are then manually decoded to retrieve the vehicle property values and projected on the map. The evaluation is eventually completed by comparing the state of the simulation – the position of the vehicles and their current

status – with the decoded values and the computed position on the map. This evaluation process ensures that the simulation results are the same like they would have been reached in the real world.

Additionally to this there were numerous tests of produced prototype parts of TFAS (Module 1 Variant A) in real vehicles driving routes calculated by the simulation while storing the created FCD messages in files. These data was also compared to the simulated results with a positive outcome.

3.4 Evaluation of Existing Simulation Systems

The task of simulating traffic has been studied for a long time. The advantage of simulating the behavior of vehicles before any construction takes place or in numbers not reachable in reality has been seen for some decades now. There are a number of readily usable scientific and commercial products available (Federal Highway Administration, 2004). The used approaches fall into several main categories. The two possibly relevant for this simulation purpose are:

- *Microscopic traffic simulation:*

Microscopic traffic simulation allows the modeling and comparison of intersections/small road networks mainly with regard to design alternatives (roundabouts, unsignalized and signal-controlled, grade separated interchanges) and design, test and analysis of signal controls. (Halcrow, 2009)

Products available in this category are e.g. VISSIM (PTV AG, 2010), TSIS-CORSIM (McTrans Center, 2010) and AIMSUN (TSS-Transport Simulation Systems, 2010).

- *Mesosopic traffic simulation:*

Mesosopic traffic simulation uses larger road networks for transportation planning, feasibility analysis and traffic engineering on a city wide or even larger scale, where simulation is an invaluable and cost-reducing tool. “Mesoscopic model travel prediction takes place on an aggregate level and does not consider dynamic speed/volume relationships” (Federal Highway Administration, 2004). Examples for products in this category are CONTRAM (Mott MacDonald, 2009) and VISTA (Vista Transport Group, Inc., 2006)

There are also hybrid-solutions that support multiple traffic simulation approaches available. In all approaches are static and dynamic traffic assignments possible. But the requirement for a simulation tools for the evaluation of TFAS are complex. Especially the requirements for the use of pre-developed components are very specific and not common. Due to the fact that the movement of single cars is the basis for the calculation of the desired simulation result is the use of mesoscopic models not possible. Out of this reason is the following evaluation of the possibilities for the use of an existing simulation solution restricted to the two most prominent microscopic traffic simulation tools available.

3.4.1 VISSIM

VISSIM (“Verkehr In Städten – SimulationsModell” – “traffic in towns – simulation model”) is developed since 1979 by the PTV Planning Transport AG in Karlsruhe, Germany. The application areas range from junction design and the examination of traffic control installation up to visualizations for decision making by political committees (PTV AG, 2010). Additionally there is a VISSIM version for

pedestrians available that is used e.g. to simulate emergency situation in stadiums (Business Geomatics, 2010). There are different representations possible.



Figure 4: 3D view of an intersection simulation with VISSIM (PTV AG, 2010)

VISSIM is a fully featured traffic simulation solution. It offers a very wide range of features and is very flexible. It could be definitely be used for the simulation of vehicle on a road network. It can use road network models in SHAPE format as well as the one directly from NAVTEQ as simulation basis. Therefore providing the ability to use the road networks used in TFAS.

VISSIM offers a programming interface allowing access to the network topology, signal control, path flows, vehicle behavior and evaluation data also providing the possibility to modify simulation parameters during simulation run time (PTV AG, 2010). The format of the programming interface unfortunately uses a quite different format than TFAS can use or connect to. Special software acting as an adapter between the two systems would have to be developed to combine VISSIM and TFAS component.

The complexity of VISSIM requiring an substantial amount of training necessary to use the system effectively, the additional effort to develop an adapter component to use the pre-developed TFAS components (in a completely new programming environment)

and eventually the licensing costs exceeding the project budget led to the conclusion not to use VISSIM for this simulation purpose.

3.4.2 TSIS-CORSIM

The most widely used microscopic traffic simulation system in the United States is CORSIM (McTrans Center, 2010). It is part of the Traffic Software Integrated System (TSIS) can be used for the analysis of freeways, urban streets and road networks. The simulation system was created by the Federal Highway Administration (FHWA) in the 1970s and has been constantly improved ever since.

Many different road geometries, traffic conditions and special transport techniques are implementable in CORSIM. Interfaces to external programs are available (McTrans Center, 2010). Overall there are many similarities to VISSIM (Bloomberg & Dale, 2000).

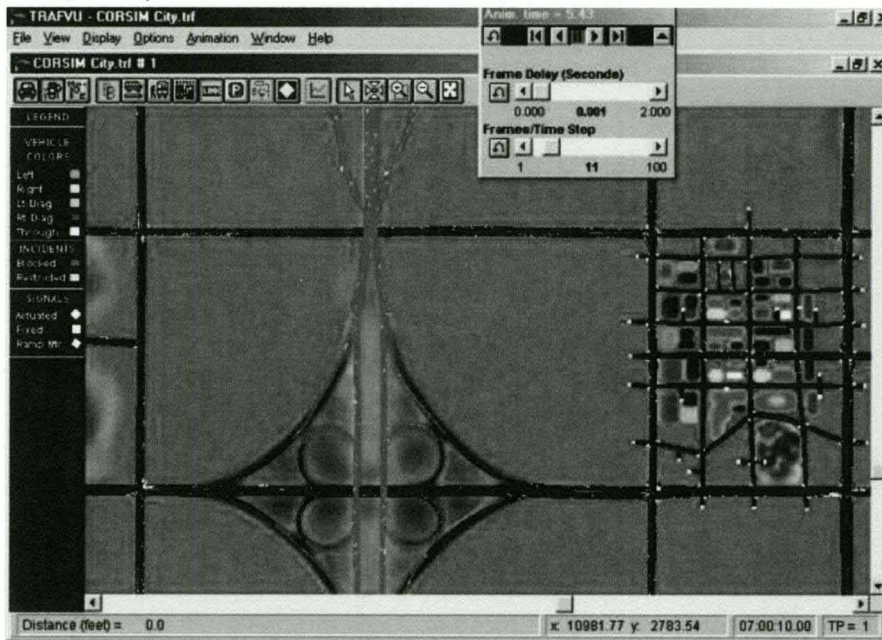


Figure 5: TRAFVU the visualization module of TSIS

CORSIM as part of TSIS does not directly have a rich visual interface. The TRAFVU component of TSIS acts as the visualization module for CORSIM. It can display running simulation to a great detail, despite the fact that it lacks the extended graphical features of VISSIM (like 3D rendering).

“CORSIM is based on a link-node network model. The links represent the roadway segments while the nodes mark a change in the roadway, an intersection, or entry points.” (Bloomberg & Dale, 2000) The map format of TFAS cannot be directly used by CORSIM. It would be necessary to develop special software for the transformation into the CORSIM format and has been done (in the other direction) before (Carbee & Trueblood, 2002).

TSIS provides a mechanism by which an external application can interface directly with CORSIM simulation. These run time extensions can hook into so-called “processing control points” and interact with CORSIM (McTrans Center, 2010).

CORSIM requires an even greater effort to use than VISSIM. The necessity to transform a map of TFAS into another proprietary format represents a new margin of error and additional work. The flexibility of CORSIM to integrate the pre-developed components of TFAS does not provide the required performance. All the capabilities of CORSIM together do not outweigh the presented drawbacks, which therefore excluded CORSIM as a possibility for this simulation purpose.

3.4.3 Conclusion

There obviously has been a lot of research in the field of traffic simulation systems and there are multiple systems available. But the objective of these systems is to simulate the traffic on certain roads networks with regard to the effect of modifications applied to the road networks themselves or controlling structures like signals. The special

requirements for a simulation system for the evaluation of TFAS do not readily fit into the mission statements of the examined systems.

It has to be stated that from today's view the combination of requirements cannot be fulfilled by any of these systems without major programming effort and unwished modification to the developed parts to be evaluated. This constitutes the need for research into and the development of an individual implementation of a simulation engine.

3.5 Simulation Methods

The development of an own simulation engine is a daunting task. This goal cannot be achieved without extensive research before the actual programming starts to develop and approve the software architecture. The first and most important step is the selection of the method used for the simulation.

The focus on the behavior of the individual vehicles predetermines the use of an agent-based model (ABM) or multi-agent simulation (Klügl, 2001). An agent-based model examines the nature of the aggregate behavior, i.e. the behavior at the system level, based on the individual decisions of its participants. Due to the fact that the individual decisions are clearly established it is possible to use a system using this approach for the intended simulation purpose of this thesis.

But which of the existing simulation methods for ABM is most suitable for this task? There is the 1970s proven psychophysical vehicle-follow-model (“Fahrzeugfolgemodell”) by Wiedemann (Wiedmann, 1974) that combines physical properties of the vehicles with driver-specific characteristics (e.g. safety desire,

sportiveness) which is used by VISSIM. The Intelligent Driver Model (IDM) (Treiber & Helbing, 2002) improves and simplifies the Wiedemann-Model by removing e.g. the discrete driving dynamics and is optimized for vehicle-borne traffic simulations e.g. for the use in adaptive cruise control systems.

A traffic model with discretely modeled space where the vehicles move in grid cells is the Nagel-Schreckenberg-Model (Nagel & Schreckenberg, 1992) based on the theory of cellular automata.

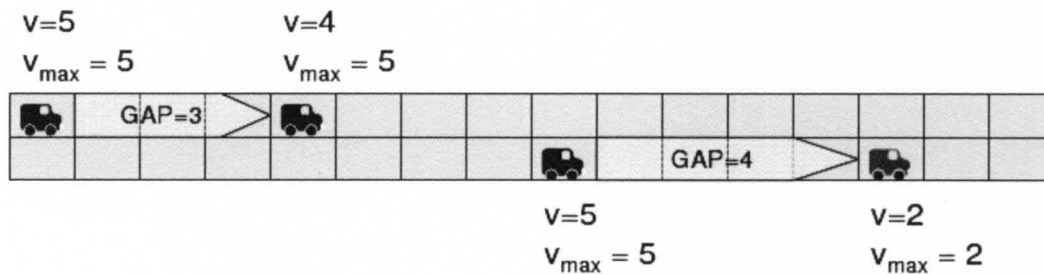


Figure 6: Cellular Automaton Model snapshot

It can be very effectively used to model even small-scale behavior e.g. on turning lanes while still providing enough performance for larger road networks and/or many vehicles.

Both of these approaches are proven and widely adopted for a large number of applications. But when the requirements of the simulation (see Chapter 3.1) are reviewed again it becomes clear that the detail level of the simulation approaches is not needed to fulfill the simulation purpose. Both models assure that the movement of the vehicles is free of collisions but the requirements (see Requirement c)) clearly state that the movement of the vehicles is not restricted and that they can pass each other freely. Additionally there are no detailed characteristics for the vehicles (e.g. turning speed) defined. Nor is there any information about possible drivers. The available space has no

discrete dimensions like width or number of lanes, which would affect the maximum number of vehicles on a road. It is just defined as a directed graph having start and end nodes as well as a defined length.

Evaluating these facts results in the acknowledgement that the simulation model can be much simpler than any of these well known models for complete traffic simulations. This is due to the fact that the result of this simulation is not the desired overall result but just the creation of the input data for the evaluation of TFAS.

Nevertheless was the eventually used simulation model not developed from scratch but does adopt some aspects of the Nagel-Schreckenberg-Model. It re-uses the concept of discrete space, just limited to the length of the street sections, and the concept of discrete time. The vehicles themselves are modeled as cellular automaton in the dimensions of mass points. The defined length for the vehicles is without relevance as long as the vehicles can pass each other freely making an exact positioning (ordering) impossible because there can be an unlimited number of vehicles side by side.

Using the characteristics for maximum speed and acceleration the basic model can be defined as follows.

Let be

- v_i the current vehicle speed,
- a the acceleration,
- v_{\max} the maximum vehicle speed,
- $v_{r, \max}$ the maximum allowed speed for the current road,
- s the distance to drive and
- Δt the elapsed time (update interval)


then for every update (discrete step in time) for vehicle i :

1. Acceleration:

2. Deceleration: $v_i := \min\{v_i, v_{road-max}\}$

3. Movement 1: $s := v_i * \Delta t$

This model provides the basic procedure for the movement on single graphs. For the movement on multiple graphs this model has to be extended:

4. Movement 2: when $s > l_{t-0}$ then $t_{use} := \frac{l_{t-0}}{v_i}$ and  and move to next road and go to step 1.

The resulting model is sufficient for the simulation purpose and allows for enough performance to simulate a larger number of vehicles on a road network.

4. SOFTWARE DESIGN

4.1 Design Approach

The simulation software for the Traffic Flow Analysis System cannot be just a single software package. The requirement to integrate pre-developed parts of TFAS that create and transmit the FCD messages of each vehicle during the simulation like they would in real vehicles defines an important part of the core simulation framework. Additionally is the TFAS server for the determination of traffic congestions currently only reachable through the use of SOAP web service so the necessary networking components have to be available.

Further requirements were further defined for the simulation software by the TFAS team:

- theoretically unlimited number of vehicles
- automatic routing for each vehicle
- graphical representation of the simulation state and of the results
- ease of use for the operators through a graphic user interface with common controls to facilitate the existing know-how
- extensive but simple configurability of the simulation in at least the following aspects:
 - o number of vehicles
 - o lower speed limit for FCD transmission
 - o age limit of FCD messages
 - o environmental influences (see Chapter 5.3)

- web address of TFAS service
- simulation speed (update interval, time warp)
- persistence functions to save and load the state of any simulation at any time to experiment with similar situations in different conditions

The combination of these requirements led to a highly modular design of the software. The main modules are the Simulation Engine, the Map Display, the Simulated Radio Receiver and the TFAS connector (see Figure 7).

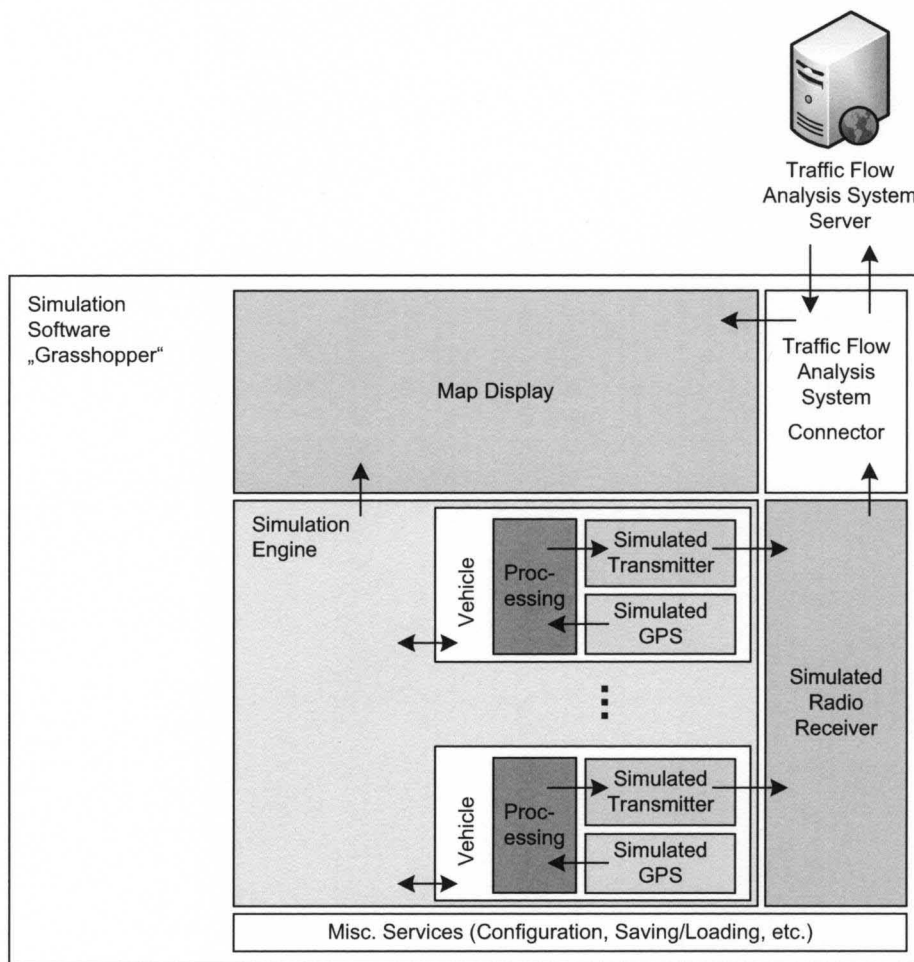


Figure 7: Schematic overview of the simulation system

The Simulation Engine

At the heart of all components is the Simulation Engine. This module is responsible for the initialization, the creation, the execution and the control of a simulation.

The simulation engine communicates with the other modules of the software, e.g. to update the map display or retrieve the configuration settings. It can initialize a simulation from scratch using the given parameters or restore a simulation session out of a save file.

The simulation engine has the ability to generate new vehicles including their initialization to create a new simulation or modify an existing session. When a new vehicle is generated its properties will be set according to the type and a random route will be calculated by the routing engine (see also Chapter 5.1).

The simulation runner as executioner maintains the list of vehicles and is responsible for the periodic update of each vehicle as time passes (execution). The control of the execution is in the hands of the simulation runner too. Every participant of the simulation (vehicle) can be controlled (Start, Pause, Reset) separately if necessary. The simulation runner also maintains an internal control clock that allows the simulation to be run a certain factor faster than real time, as an important tool for the efficient analysis of the simulation objective. Additionally the internal clock and with this the whole execution can be stopped or resumed at any time, e.g. to allow the run of simulation persistence functions.

The Vehicle

A vehicle represents a participant of a simulation. Each vehicle is a separate entity with its own state and properties. Every vehicle independently and continuously calculates a number of properties to represent the current state (see Figure 8), e.g. its average speed over a defined time frame using the speed limit of the current street section as its current

speed. The entirety of all vehicles does not share any common data. A vehicle in this simulation actively communicates with the virtual radio only. It is otherwise passively controlled and observed by the simulation runner.

The total number of vehicles in a simulation is theoretically unlimited only restricted by the capabilities of the executing computers hardware configuration, especially main memory (RAM) size.

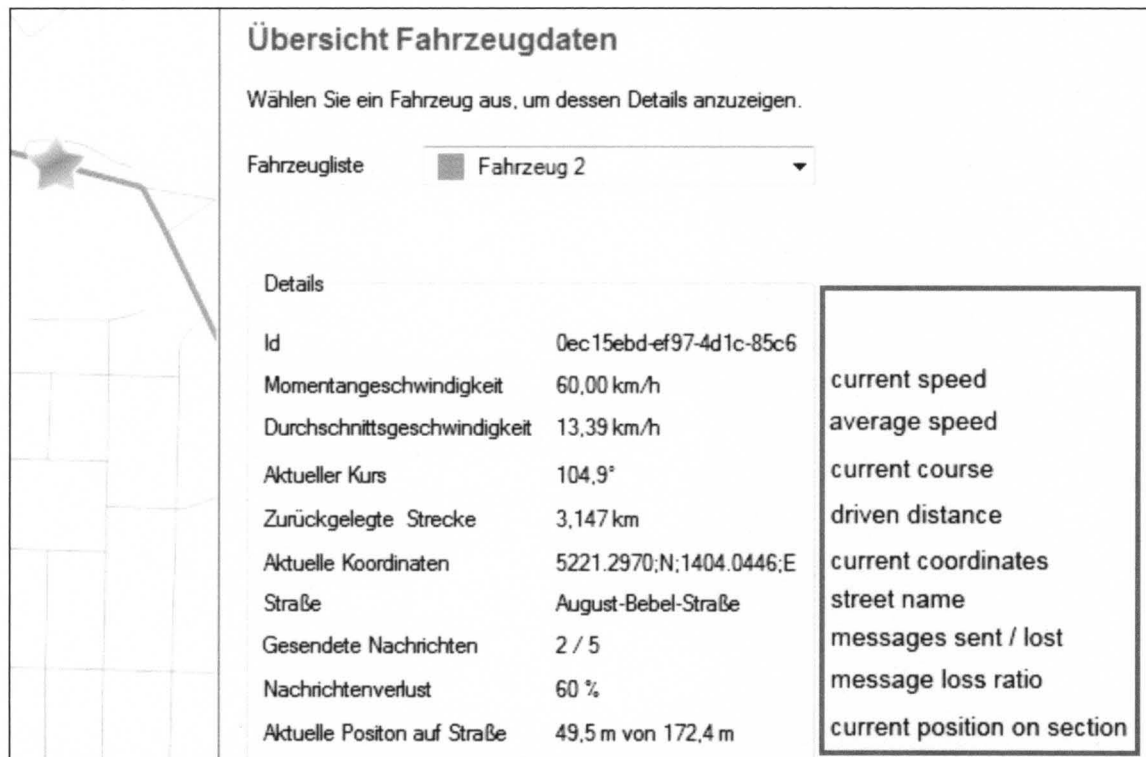


Figure 8: Vehicle properties (with translation)

The vehicles do not only create the data for the evaluation of the TFAS algorithms but also use a pre-developed component of TFAS (Processing) to generate this data. This approach allows the evaluation of these components at the same time as the simulation runs. The processing component retrieves position data from a location awareness component and transmits the calculated FCD messages to the radio component. Both these connections are loosely coupled using standardized interfaces following the

component-based development approach (Szyperski, 2002) that allow the exchange of the real components from the mobile devices with mock-up components of the simulation software. The mock-up component for location awareness retrieves the current coordinates of the vehicle out of its position on the map instead from GPS. While the simulated radio transmitter transfers the FCD message not over the air but into a “virtual ether” (see Chapter 5.3).

The Simulated Radio Receiver

As there is only one assumed vehicle with a receiver in this simulation while all the other simulated vehicles are only sending FCD messages there is also only one simulated radio receiver. The radio receiver retrieves all FCD messages out of the virtual ether. These messages are then validated and prepared for the transfer to the TFAS traffic congestion analysis service via the TFAS Connector. This process can be controlled manually in the simulation software GUI.

The TFAS Connector

The TFAS traffic congestion analysis service runs – at this stage of the development process – on a separate server to provide the necessary performance. The communication with this service has to be warranted over system boundaries, e.g. out of local networks as well as from mobile stations via the 3G networks. The service is therefore reachable for the simulation software as a SOAP web service (Hohpe & Woolf, 2003) using common standards via the TFAS Connector.

The input of the service is a collection of FCD messages representing the current traffic situation on a road network. The service analyses this situation and calculates any possible traffic congestions. The analysis result is then transferred back to the TFAS Connector using another web service integrated into it.

The calculated possible traffic congestions are then evaluated and prepared to be transferred and displayed on the map by the Map Display.

The Map Display

The simulation software uses a map to warrant the ease of use of the simulation and to provide a clear overview of the simulation state at all times.



Figure 9: Detail of the Map Display showing a vehicle approaching a blocked street section

The visible map itself is reduced to a high degree displaying roads as simple lines without any names or any other orientation points.

More details about the Map Display can be found in Chapter 5.1.

Miscellaneous Services

The simulation software provides means to setup, configure and control the simulations. The necessary supporting methods are integrated in this module. This includes all file handling, the persistence of configuration settings and general user interaction like error handling.

4.2 The Data Model

The data for the simulation consist of two parts: the map of the road network and a database of all nodes and connecting roads of the road network.

The map of the road network, in this case the map of the small city of Fürstenwalde³ in Brandenburg/Germany, was supplied by TeleAtlas BV⁴ out of the MultiNet product. TeleAtlas of Belgium is one of the two worldwide leading providers of street maps for routing purposes, the other being Navtec⁵ in Chicago, IL.

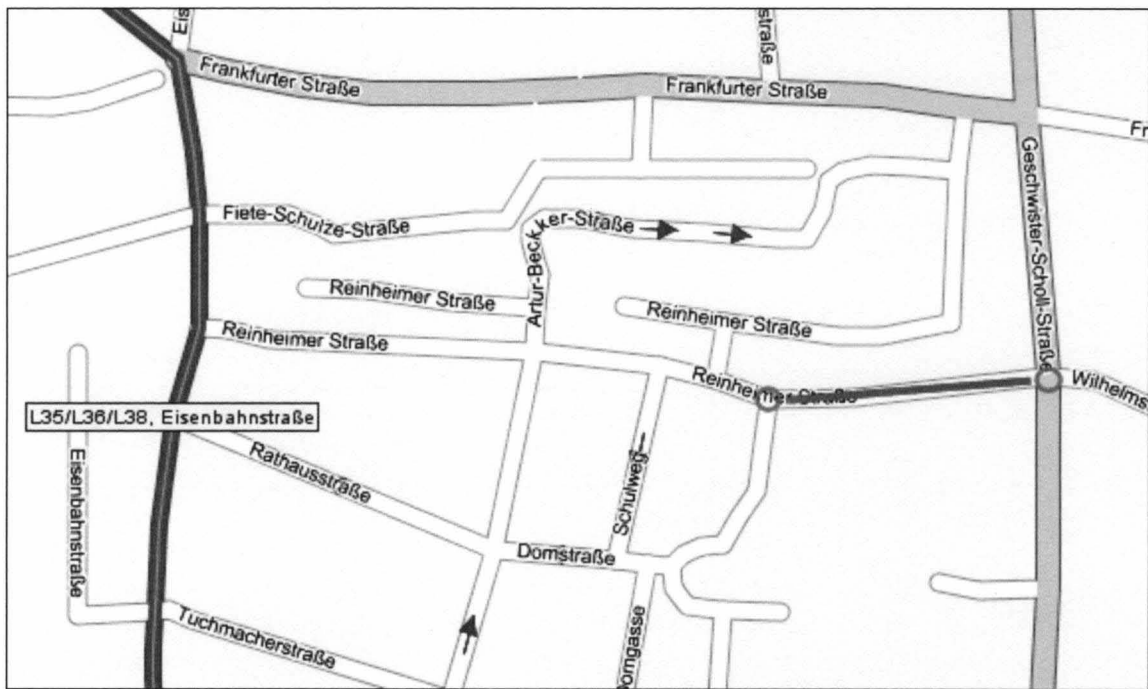


Figure 10: Detail of a road map for navigation purposes (www.navteq.de)

Maps for routing purposes are very detailed. Every street or way is separated into sections usually starting at an intersection and ending at an intersection. The intersections are called Nodes (see red circles in Figure 10). Connecting the nodes are street sections (see blue line in Figure 10). For every section there are at least the following facts registered:

- unique ID
- street name,
- street category
- driving restrictions (e.g. one-way roads)
- house numbers (e.g. from, to)
- length
- speed limit
- driving time at speed limit

There can be many more aspects for each section like number of lanes and road surface types but these are without any relevance for this simulation⁶.

Additionally to the map there is a separate database of the road network containing information about the nodes and the sections. This database in Microsoft Access format is used for routing purposes by the route finding method described in Chapter 5.1. The most important data contained are the length, the driving time and driving restrictions for each section. This redundant database is necessary due to the design decision to implement the routing mechanism as an independent component not using the maps of the GIS.

4.3 Program Language and Software Development

Environment

A wide spectrum of possible choices is available as software tools to implement the necessary functionalities, e.g. Java⁷ as the main programming language in the academic field. Due restriction in time and available capacities was the selection strongly influenced by the existing know-how and experiences of the author.

Therefore the implementation of the software architecture is done by the use of technologies and products from Microsoft. The C# programming language and the .NET Framework version 3.5⁸ are used as the development platform following an strictly object-oriented approach. The .NET Framework provides a vast variety of functions and provides a reliable basis to rapidly develop the necessary software components.

The integrated software development environment is Microsoft Visual Studio 2008 Professional⁹ and Microsoft Windows XP¹⁰ is the operating system of the development computers. Newer versions of Microsoft Windows are supported as well.

The ThinkGeo MapSuite Desktop Edition 3.0¹¹ as a third party component is used to implement the GIS functionality.

All products are characterized by good performance, ease of use and offer the best conditions for the fulfillment of the requirements.

5. SYSTEM INTEGRATION

5.1 GIS Applications

The graphical representation of the simulation on the screen is mainly realized by the third party mapping component ThinkGeo MapSuite Desktop Edition 3.0. This component displays a map showing the road network, the currently simulated vehicles and their route, the established roadblocks and the calculated traffic congestions.

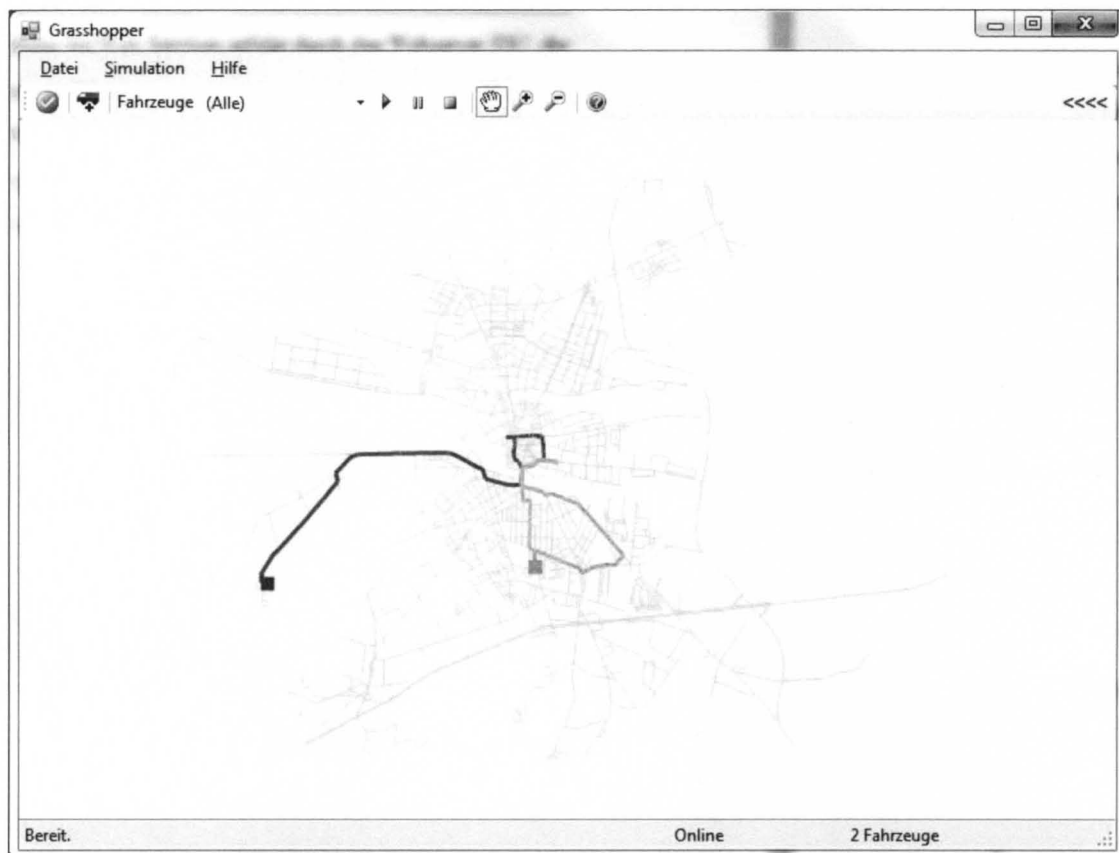


Figure 11: Main window of the simulation application with two vehicles

The World Geodetic System 84 (WGS84)¹² as the standard coordinate system in the geodetic world and the reference coordinate system used by the Global Positioning System (GPS)¹³ is facilitated by the simulation to display the map and calculate the positions.

Street Network

The street network is displayed on the in a much reduced way. Only the street sections are shown as simple light green lines (see Figure 9). There are no visible additional characterizations for e.g. street category or driving restrictions nor are there any street names or numbers. This minimalistic approach significantly increases the system performance for simulations with a large number of vehicles while having no negative impact on the informative value regarding the simulation state, as there is no importance on the real geographic location of the used road network.

Vehicles

The simulated vehicles are displayed on the map as squares. The randomly generated route for each vehicle is additionally displayed to the square. The color of the square and the route are different for every vehicle to allow easier differentiation.



Figure 12: Detail map with two vehicles and their routes

The map is periodically updated with a standard interval of 1 second. That means that the movement of the vehicle indicators is not smooth but sufficient. The update interval can be individually set by the operator in units of seconds, which is especially helpful when simulating a large number of vehicles.

Road Blocks

To provoke the creating and sending of a FCD message the average speed of a vehicle has to fall below a defined speed limit. It is therefore need to slow vehicles down to stop to reduce their average speed. This is accomplished by marking certain street sections as blocked using a tool of the Map Display.

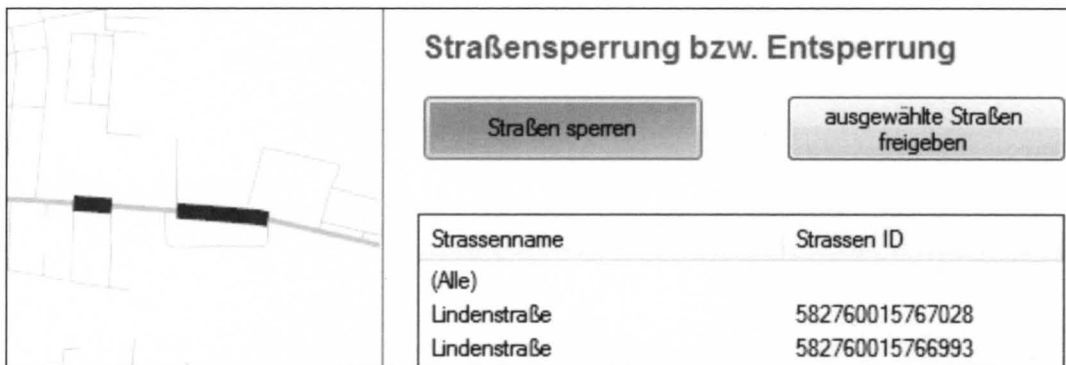


Figure 13: Two blocked street sections and their detail information (street name, ID)

The blocked sections are identified by their unique section ID and marked in the map by a thick magenta line. The simulation runner uses the list of blocked sections to determine the actions for each vehicle. The block of a section can be removed either via the map tool or through the list on the right (see Figure 13).

Traffic congestions

The analysis of a collection of FCD messages to recognize possible traffic congestions is the objective of TFAS traffic congestion analysis service. The result of this

analysis has to be displayed in the map to provide the means for a first evaluation of validity.



Figure 14: Detail map displaying a traffic congestion including the affected street section

A traffic congestion can start anywhere on a street section in any direction and can reach over multiple street sections (see Figure 14). Additionally to this the analysis result contains information about the number of messages evaluated, the number of cars involved and the time of first occurrence. All these data can be displayed in the simulation software.

Basic Mapping Functions

To provide a useful interface for the operator the Map Display also includes the standard functions of every mapping application. This includes:

- Panning of the map using the mouse
- Zoom In on the map using a rubber band or via mouse click and
- Zoom Out of the map using a rubber band or via mouse click

The use of the commercial 3rd party software component for the Map Display allowed the efficient and successful realization of the required functionalities in the limited available timeframe. Other option, like Open-Source components¹⁴, are available

but – at the time – did not provide the performance and ease of use necessary and were therefore after short test discarded.

5.2 Routing

A separate software component was developed for the calculation of the route for the vehicles. It uses the same road network like the map but out of the separate database (see Chapter 4.2) containing all the nodes and the directed and undirected graphs connecting them including information about the length, the respective driving time and potential driving restrictions for each graph. This data represents the basis for the construction of a complex graph to calculate a route between two of the nodes (see Figure 15). The availability of both length and time allows the calculation of the shortest as well as the fastest route.

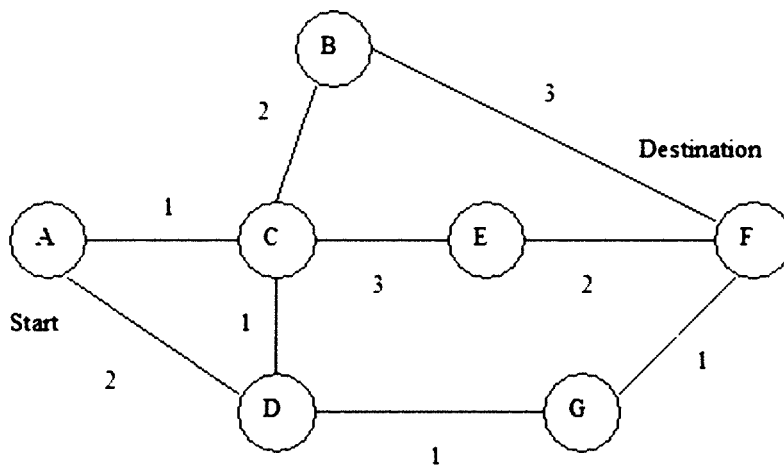


Figure 15: Simplification of a road network as graph (Broadbent, 2008)

The simplest possible route for a vehicle would be between two points but with only little relevance for reality with the exception of public transport systems. A more reasonable route would be one between three or more points. The routing component therefore allows for the specification of the desired number of nodes used as cornerstones

for the route to be calculated. The nodes themselves are randomly selected out of the road network to provide a route as unique as possible for each vehicle. This simulation uses three nodes for the routes as this value was found to be sufficient for the intended purpose as well as timesaving for the simulation of a larger number of vehicles.

The Dijkstra Algorithm (Cormen, Leiserson, Rivest, & Stein, 2009) is used to solve the shortest path problem for the calculation of the most efficient route between two nodes. This algorithm is a very proven method for producing a shortest path tree out of a graph without non-negative cost for the edge paths. The cost being the length of a street section for the shortest path or the time needed to travel along this section for the fastest path.

The result of the route calculation is a list of street sections forming a closed loop as route for a vehicle. This list is then returned to the calling component and used by the simulation runner to compute the movement of a vehicle.

5.3 Simulated External Systems

It is an important requirement for the simulation software to integrate and therefore evaluate the processing component of the TFAS software package for the real vehicles (see Chapter 4.1). This component runs on mobile devices and/or notebooks and has direct access to a GPS device for locational awareness and to a radio device for the sending and receiving of the FCD messages. To integrate this component into the simulation the simulation software has to provide the component exactly the same access capabilities to these devices. Differentiating only in the fact that it transparently (for the component) uses simulated instead of real devices.

GPS

The Global Positioning System is used in TFAS to provide a participating vehicle with information about its current location, its current speed and current course. All this data can be directly retrieved from the software module that integrates the GPS receiver into TFAS. This data forms the basis for the vehicles assessment of its current situation and eventually of the generation of the FCD messages when its average speed falls below a defined limit. This assessment is done periodically in an interval of seconds so the GPS information is retrieved as needed and not permanently available.

This fact is important for the simulated GPS device, as it has to determine the current location of the simulated vehicle using coordinates like the real device out of the properties available (see Figure 8). The relevant data out of the properties for this task are the current street section (out of the street name) and the current position on this section. It is important to note that the current position is given as the distance from the starting node of the section with the starting node defined by the map and not by the driving direction of the vehicle. Having access to an own GIS component the simulated GPS device is now able to use this data to localize the street section and after that the exact coordinate of the position on this section.

The current course is then approximated using the run of the road curve. To do this the WGS84 coordinate system is simplified into a two dimensional coordinate system. The coordinates of multiple points before and after the current position of the vehicle (+/-5 meters; +/-10 meters) are calculated and used together with the current positions coordinates as input values for a linear regression algorithm (Fahrmeir, Kneib, & Lang, 2007) calculating the slope of the result. The slope is then transformed into the current course of the vehicle respecting the current driving direction on the street section (Beginning to End; End to Beginning).

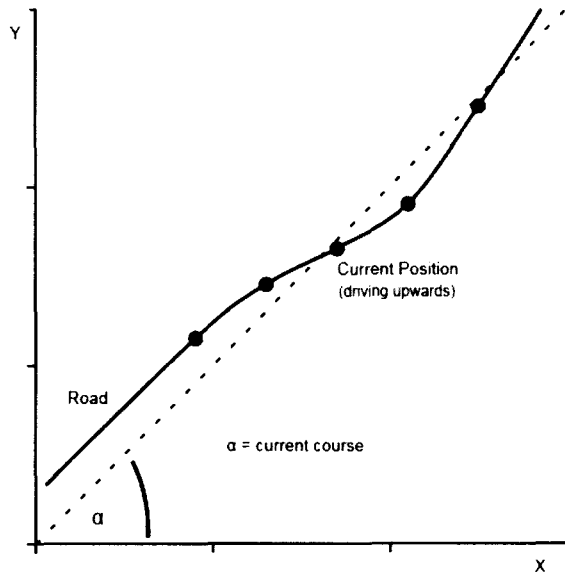


Figure 16: Calculation of the current course using regression

Despite the fact that a normal GPS connection has some dependency on the location of the receiver like the number of satellites visible (Modsching, Kramer, & ten Hagen, 2006) the simulated GPS device does not emulate any environmental influence.

Radio

Radio transmissions in TFAS are used to transfer the FCD message from a sending vehicle to all receiving vehicles for further evaluation. Any radio transmission equipment on the market can theoretically achieve this. Multiple transceivers ARF35 by Adeunis RF (France)¹⁵ for the license-free frequency of 868 MHz were used for this task during the research period. The software component integrating this device provides functionality to send a single FCD message and to signal the reception of a new FCD message.

The requirements of the simulation software simplified the real situation in the way that it is assumed that there are many sending vehicles – the simulated ones; called ‘Module 1A’ in the TFAS specification – and just one receiving vehicle – the simulation software itself. This is the reason that the simulated vehicles like ‘Module 1A’ only have

access to a sending device and are not able to receive any messages. The simulation software - in this case the Simulated Radio Receiver (see Chapter 4.1) only receives FCD messages but does not send any data.

The emulation of the radio transmission in the simulation is done by virtualizing the transmission medium through the construct of a 'virtual ether'. Every Simulated Transmitter (see Chapter 4.1) transmits its FCD messages into the 'virtual ether' where the simulated radio receiver can then pick up the messages sent from all vehicles. Periodically a support process cleans up the 'virtual ether' to remove messages that are deemed too old. Using this construct is more complex than necessary because of the encoding and decoding of the transmitted FCD messages by transmitter and receiver but it allows the use of the pre-developed components. After further tests it was also found to have no significant negative effect on overall simulation performance.

The quality of the radio transmissions in reality is heavily influenced by the environmental conditions (Herter & Lörcher, 2004) like the impact of e.g. city buildings and hills. To respect these influences the simulated radio transmission in the simulation software is also impacted by a (theoretically) non-deterministic effect. This effect is controlled by two parameters:

- Probability of a message to be sent in per cent and
- Maximum variance (+/-) of the probability in percent.

The actual used variance is then randomly set separate for each vehicle and period by the simulation engine. This means that with a probability of 50% and a variance of the probability of 50% a radio message has an effective probability to be sent ranging from 25% to 75%. The values for these parameters are set in the configuration of the simulation.

6. IMPLEMENTATION

This thesis does not describe the software system implementation in detail, as its field of work does not directly concern the application of modern software development techniques and methodologies. A description of common and reliable approaches for the theoretical as well as the practical aspects can be found in the recognized works of many authors, e.g. (Knuth, 1998), (Pomberger & Dobler, 2008) or (Martin, 2008).

This chapter concentrates on two facets of the software development that had a distinctive influence on how the developed simulation model was implemented to allow the effective evaluation of the Traffic Flow Analysis System while providing sufficient performance.

Simulation Runner

The Simulation Runner – a term taken over from (Himmelspach, Ewald, & Uhrmacher, 2008) – is the core component for the computational execution of the simulation model. It is responsible for internal organization of the used data containers as well as the controlling of all activities of the simulation participants, e.g. the simulated vehicles. The point of interest in this case are not the applied data structures for the current values of the vehicle properties - like linked-lists or generic collections - nor the communication using an eventing infrastructure but the fundamental task execution engine which is single handedly responsible for allocation of processor time to each of the simulation model instances.

There are basically two types of models for the execution of tasks: sequential and parallel computation. This differentiation is a direct consequence of the number of processors available (Herold, Lurz, & Wohlrab, 2007) as anything else like the multi-tasking by operating systems (e.g. Microsoft Windows or Linux) on single processor computers is not real parallel processing but just a simulation thereof using time sharing systems to simulate the simultaneous execution of processes (Tanenbaum, 2007). Only with the introduction of multi-core processors into the mass-market by Intel in 2006 (Intel Corporation, 2006) did become true multi-tasking real.

Nevertheless was for the Simulation Runner of the evaluation environment the decision made to implement it using a traditional sequential processing model. This was done for a number reasons:

- *Simplicity*: The sequential processing model is well known and can be realized using any given toolset. The support for well maintainable true parallel processing routines for the Microsoft .NET Framework was not released to production before the April 2010 (Microsoft Corporation, 2010) significantly after the start of the development.
- *Reliability*: The evaluation of sequential software algorithms is controllable because their execution is predictable and therefore repeatable. The project restrictions regarding time and resources encouraged the use of well known but eventually slower sequential programming techniques due to the fact that only sufficient not best performance was required.
- *Reusability*: The objective for the simulation environment asked for the reuse of already pre-developed software components of TFAS. These components were not developed with multiple instance execution as a requirement. This resulted in the lack of thread safety, i.e. the inability to

safely run simultaneously producing correct result in any situation.

Missing thread-safety can lead to not reproducible errors that are difficult to detect and analyze (Herlihy & Shavit, 2008) thereby increasing the projects risks of failure or delayed completion.

It was however respected that there could be need to a shift from the sequential to the parallel execution model at a later time, i.e. the software components outside the Simulation Runner are designed to support either model. The introduction of a new execution model implementation would require a nearly complete rewrite of the Simulation Runner and some of the data structures as the algorithms and data structures differ for sequential and parallel execution.

Random Number Generator

The generation of pseudo random numbers is a key feature in the simulation. The calculation of the routes as well as the simulation of the environmental influences is depending on it. It is however well known, that the random number generators of many standard libraries and so also the one contained in the .NET Framework do not met normal stochastic requirements regarding randomness (Matsumoto, Wada, Kuramoto, & Ashihara, 2007) using their default settings.

This issue was addressed through the use of special seed values for the random number generator that were retrieved by the combination of several often-changing values of the computing environment (e.g. time, temperature, mouse movement).

If the random number generation had been of utter importance for this simulation purpose a better solution would have been the implementation of a framework that allowed the use of different random number generators for different simulation runs like the one described in (Ewald, Rössel, Himmelspach, & Uhmacher, 2008).

7. EVALUATION AND RESULTS

The performance of the simulation system largely depends on the available computing capacity especially the main memory. Using a state-of-the-art computer with the Intel i7 processor and 8 Gigabytes of main memory it is possible to simulate about 1.000 vehicles at a time on the road network of the city of Fürstenwalde.

The component that uses the most computing power during a simulation run is not the simulation model or the simulation runner but the graphical visualization by the map component. This performance need of the map component does not rise linear with the number of vehicles simulated but more quadratically or even exponentially. The simulation software can control the load consumed by the map component by changing the map update interval. The map updates become more and more visible when the value is increased but the necessary processor usage decreases freeing resources for additional vehicles.

The maximum capacity of the simulated radio component was not reached during any simulation run there not restricting the maximum amount of data collected for the TFAS evaluation.

The manual inspection of the collected data did not reveal any severe inaccuracies or differences to data collected by real vehicles. The main reasons for differences were deviations of the TeleAtlas road maps from the real world. Especially roundabouts and junctions, which are particularly simplified, created slightly inaccurate coordinates that

sometimes led to difficulties for the TFAS congestion service to map the vehicle to the roads again.

The simulation of the environmental influences to the radio transmission provided a good average result in comparison with real radio transmissions. The deviation largely depends on the geographic location of the vehicle, with a larger positive divergence in the city center. This effect was to be expected and does not influence the data quality from the viewpoint of the simulation purpose.

8. DISCUSSION

The developed simulation environment was an invaluable tool for the evaluation of TFAS. The project resources would have never allowed the evaluation using only real world data. Neither the prototype devices nor the necessary cars could have been manufactured or provided.

The conception and the development effort also consumed valuable resources but these were – on one hand – planned from the start for the use of an existing simulation system and on the other hand could be freed up for the creation of an individual solution that really fulfills the requirements.

The development of the simulation model certainly was the key point for the success of this effort. Selecting the wrong model would have had a significant influence on the outcome. A model too complicated could have slowed the performance of the simulation and possibly required additional preparation of the data basis. A model too simple would have deviated too much from the real world effectively removing the credibility from the generated data. The use of cellular automaton was the best choice for this because the vehicle-follower-model depends on the vehicle following each other, which was explicitly excluded as a requirement for this simulation. Additionally did the concept of discrete time provide a very efficient way to implement this model using a programming environment.

The implementation of the simulation model incorporates the use of fundamental programming patterns. The sequential processing of the vehicles is a reliable and sufficient procedure but it leaves room for improvement as will be described in Chapter 10. The graphic visualization provides a basic interface for the setup and control of the simulation. It also provides connectivity to the other TFAS components and proved to be an efficient tool. The tool is not self-explanatory but the usability is good enough to require only a brief training for the operators.

The simulation software – eventually called ‘Grasshopper’ – evolved into an integral part of TFAS as it can be used to demonstrate the whole system, which wasn’t initially planned. Grasshopper was successfully demonstrated to the research group.

The work on Grasshopper provided me valuable insights into the field of simulation and significantly improved my knowledge and experience in the conception and planning of complex software systems.

9. CONCLUSIONS

9.1 System Capabilities

The successful integration of the simulation engine as well as the pre-developed components of the Traffic Flow Analysis System established a simulation system that creates the FCD needed to evaluate the Traffic Flow Analysis System and displays the returned results on the screen of a Windows application. The behavior of vehicles used in this simulation is graphically presented on the road network, while their movement will follow the dynamic traffic assignment approach. The state of simulations can be saved and retrieved to deterministically evaluate the Traffic Flow Analysis System in different conditions.

9.2 System Limitations

The developed simulation environment fulfills the stated requirements of the project. Nevertheless are there a number of limitations that are restricting the simulation in its flexibility and applicability:

- The simulation of the participating vehicles as mass points is a major restriction of the current version. It strongly influences the ability to simulate traffic congestions at a real scale spanning multiple streets. While this has no effect on the overall recognition of traffic congestions it limits the ability to evaluate the advanced possibilities of TFAS.

- The created simulation model was stripped down to just the at least necessary properties. The level of simulated realism could be improved by the introduction of further properties like the ones in Table 1 or the implementation of a more controllable routing mechanism. The gained advantages however should be carefully weighted against the possible decrease in performance and the required time/costs for their development.
- There is no variance in the simulated types of vehicles as there is just one simulation model implemented. This could also be improved by the introduction of more than one simulation model instance representing multiple types of vehicles, e.g. passenger cars, truck, busses etc., but again the purpose of the simulation is not targeted at the realism of the simulated traffic but on the creation of a collection of data for the evaluation of the Traffic Congestion Analyzer without immediate regard for the process how this data was collected therefore restricting the usefulness of the availability of multiple vehicle types.
- In the current state of the system it is only possible to block complete street sections for any traffic thereby creating the desired traffic congestions. Traffic congestion therefore always starts at the beginning or the end of a street section limiting the possible variance of the data. The evaluation of TFAS would be more realistic and comprehensive when it would be possible to block a street section at any point.
- The available road network is currently limited to single local area. A larger road network could provide a greater variability for the location of traffic congestions and/or the routes of the vehicles. The usefulness of such an extension however remains questionable.

The listed limitations – in decreasing order of importance – have to be known to correctly utilize the simulation environment and assess the created results while additionally presenting the leads for its further improvement.

10. FUTURE WORK

As shown in the discussion of the results in Chapter 8 and the list of the systems limitations there are capabilities of the simulation environment that could be improved or should be the objective of further research.

Further research is required for cases where the usefulness from the viewpoint of the simulation purpose can be disputed. This especially includes the introduction of more complex simulation models, e.g. supporting passing, turning and/or controllable routing, and the availability of multiple vehicle types for the simulation entities where the change in results and its decisiveness after the implementation cannot be easily overseen.

Improvements with respect to performance could be achieved by an overall optimization of the Simulation Runner employing an architecture change for the utilization of the full capabilities of modern multi-core processors. These scalable simulation engines (Fujimoto, 2007) increase the speed of operation by partitioning the work to be done and assigning the so created work packages between the available processors applying load-balancing methods for optimal efficiency. An approach that would be easily applicable to the collection of simulated vehicles in the developed simulation environment.

Additionally has to be noted that the realization of a new revision for the software development itself could be used to address some minor usability issues and stability problems as well.

REFERENCES

- Bartelme, N. (2005). *Geoinformatik: Modelle, Strukturen, Funktionen*. Berlin: Springer.
- Benkner, T. (2007). *Grundlagen des Mobilfunks*. Wilburgstetten: J. Schlembach Fachverlag.
- Bloomberg, L., & Dale, J. (2000, August). *A Comparison of the VISSIM and CORSIM Traffic Simulation Models*. Retrieved June 1, 2010, from PTV America: http://www.ptvamerica.com/fileadmin/files_ptvamerica.com/library/2000%20ITE%20VISSIM%20+%20CORSIM%20Safeco.pdf
- Braess, H.-H., & Seiffert, U. (2007). *Vieweg Handbuch Kraftfahrzeugtechnik*. Wiesbaden: Vieweg+Teubner Verlag .
- Broadbent, N. (2008, Junly 15). *Dijkstra's Algorithm code in C++*. Retrieved June 1, 2010, from reviewmylife - Technology, programming, travel and general banter: <http://www.reviewmylife.co.uk/data/2008/0715/dijkstras-graph.gif>
- Business Geomatics. (2010, January). Virtuelle Weg ins Stadion. *Business Geomatics 1/10*, p. 15.
- Carbee, J., & Trueblood, M. (2002). *GIS Applications within Traffic Simulation Projects*. Retrieved June 1, 2010, from Missouri Valley Section of the Institute of Transportation Engineers (MOVITE) Homepage: http://www.movite.org/PDFs/Articles/02ITE_Papers_Carbee_Trueblood.pdf
- Cellier, F. (1991). *Continuous system modeling*. Secaucus, NJ, USA: Springer Verlag New York, Inc.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms (3rd ed.)*. Cambridge: MIT Press.

- Ewald, R., Rössel, J., Himmelspach, J., & Uhmacher, A. (2008). A plug-in based architecture for random number generation in simulation systems. *WSC '08: Proceedings of the 40th conference on Winter simulation*.
- Fahrmeir, L., Kneib, T., & Lang, S. (2007). *Regression: Modelle, Methoden und Anwendungen*. Berlin: Springer Verlag.
- Federal Highway Administration. (2004, June). *Categories of Traffic Analysis Tools*. Retrieved June 1, 2010, from Traffic Analysis Toolbox Volume II: Decision Support Methodology for Selecting Traffic Analysis Tools:
http://ops.fhwa.dot.gov/trafficanalysistools/tat_vol2/sect1.htm#sect1_3
- Federal Highway Administration. (2004, June). *Traffic Analysis Tools by Category*. Retrieved June 1, 2010, from Traffic Analysis Toolbox Volume II: Decision Support Methodology for Selecting Traffic Analysis Tools:
http://ops.fhwa.dot.gov/trafficanalysistools/tat_vol2/sectapp_e.htm
- Feinstein, A. H., & Cannon, H. M. (2002, December). Constructs of simulation evaluation. *SIMULATION & GAMING, Vol. 33 No. 4*, pp. 425-440.
- Fujimoto, R. (2007). Keynote: Modeling, simulation and parallel computation: The future is now. *Proceedings of the 2007 Spring simulation multiconference, Volume 1*, (pp. vii-viii: SCS).
- Halcrow. (2009). *Transportation Planning*. Retrieved June 1, 2010, from Halcrow Homepage:
http://www.halcrow.com/Documents/transport_planning/Microscopic_traffic_simulation.pdf
- Herlihy, M., & Shavit, N. (2008). *The Art of Multiprocessor Programming*. Burlington, MA: Morgan Kaufmann Publishers.
- Herold, H., Lurz, B., & Wohlrab, J. (2007). *Grundlagen der Informatik*. Munich: Pearson Studium.

- Herter, E., & Lörcher, W. (2004). *Nachrichtentechnik: Übertragung, Vermittlung und Verarbeitung (9. Auflage)*. Munich: Hanser Fachbuchverlag.
- Himmelspach, J., Ewald, R., & Uhrmacher, A. M. (2008). A flexible and scalable experimentation layer. *Proceedings of the 40th Conference on Winter Simulation (Miami, Florida, December 07 - 10, 2008)* (pp. 827-835). Miami, FL: S. Mason, R. Hill, L. Mönch, and O. Rose, Eds.
- Hohpe, G., & Woolf, B. (2003). *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Amsterdam: Addison-Wesley Longman.
- Intel Corporation. (2006). *Intel® Core™ Duo Processor Family*. Retrieved 2010 йил 01-06 from Intel Homepage:
<http://ark.intel.com/ProductCollection.aspx?familyID=22731>
- Klügl, F. (2001). *Multiagentensimulation, Konzepte Werkzeuge, Anwendungen*. München: Addison-Wesley Verlag.
- Knuth, D. E. (1998). *Art of Computer Programming, The, 2d ed.* New York: Addison-Wesley Professional.
- Martin, R. C. (2008). *Clean Code: A Handbook of Agile Software Craftsmanship, 1st ed.* Upper Saddle River, NJ: Prentice Hall.
- Matsumoto, M., Wada, I., Kuramoto, A., & Ashihara, A. (2007). Common defects in initialization of pseudorandom number generators. *ACM Trans. Model. Comput. Simul.* , 15 (1-20).
- McTrans Center. (2010, March 2). *TSIS: Traffic Software Integrated System*. Retrieved June 1, 2010, from McTrans Center Homepage:
<http://mctrans.ce.ufl.edu/featured/tsis/>
- Microsoft Corporation. (2010). *Parallel Computing*. Retrieved 2010 йил 01-June from Microsoft Developer Network: <http://msdn.microsoft.com/en-us/concurrency/default.aspx>

- Modsching, M., Kramer, R., & ten Hagen, K. (2006). Field trial on GPS Accuracy in a medium size city: The influence of built-up. *PROCEEDINGS OF THE 3rd WORKSHOP ON POSITIONING, NAVIGATION AND COMMUNICATION (WPNC'06)* (pp. 209-218). Aachen: Shaker Verlag.
- Mott MacDonald. (2009, July 17). *CONTRAM Homepage*. Retrieved June 1, 2010, from <http://www.contram.com/>
- Nagel, K., & Schreckenberg, M. (1992, December). A cellular automaton model for freeway traffic. *J. Phys. I France 2 (12)* , pp. 2221-2229.
- PNT, N. E.-B. (2009, May 28). *The Global Positioning System*. Retrieved June 1, 2010, from Space-Based Positioning, Navigation & Timing: <http://pnt.gov/101/gps.html>
- Pomberger, G., & Dobler, H. (2008). *Algorithmen und Datenstrukturen: Eine systematische Einführung in die Programmierung*. Munich: Pearson Studium.
- PTV AG. (2010). *"VISSIM – State-of-the-Art Multi-Modal Simulation" Brochure*. Retrieved June 1, 2010, from VISSIM - Mikroskopische Simulation: http://www.ptv.de/fileadmin/files_ptv.de/download/traffic/Vissim-Flyer_PTV_E_High_1_.pdf
- PTV AG. (2010). *VISSIM - Mikroskopische Simulation*. Retrieved June 1, 2010, from PTV Homepage: <http://www.ptv.de/software/verkehrsplanung-verkehrstechnik/software-und-system-solutions/vissim/>
- Rech, J. (2008). *802.11-WLAN-Technologie und praktische Umsetzung im Detail*. Hannover: Heise Verlag.
- Reuter, U. (1998, June 22). *Berechnung der Fahrleistungen am Beispiel des VW1200*. Retrieved June 1, 2010, from flat4: <http://www.flat4.de/fahrleis.htm>
- Schoder, G. (2009). *Floating Car Data: Konzepte und Nutzung in der Logistik* . Saarbrücken: VDM Verlag.
- Szyperski, C. (2002). *Component Software: Beyond Object-Oriented Programming (2nd Edition)*. New York: Addison-Wesley Professional.

- Tanenbaum, A. S. (2007). *Modern Operating Systems (3rd Edition)*. Upper Saddle river, NJ: Prentice Hall.
- Treiber, M., & Helbing, D. (2002, September). *Realistische Mikrosimulation von Straßenverkehr mit einem einfachen Modell*. Retrieved June 1, 2010, from Publications - Dr. Treiber: <http://www.mtreiber.de/publ.html>
- TSS-Transport Simulation Systems. (2010). *Aimsun*. Retrieved June 1, 2010, from TSS-Transport Simulation Systems Homepage: <http://www.aimsun.com/site/content/category/1/32/53/>
- Vista Transport Group, Inc. (2006). *VISTA Homepage*. Retrieved June 1, 2010, from <http://www.vistatransport.com/products/>
- Wiedmann, R. (1974). Simulation des Straßenverkehrsflusses. In *Schriftenreihe am Institut für Verkehrswesen, Heft 8*. Karlsruhe: Institut für Verkehrswesen der Universität Karlsruhe.
- Wikipedia. (2010, June 15). *Traffic Message Channel*. Retrieved June 17, 2010, from Wikipedia.com: http://en.wikipedia.org/wiki/Traffic_Message_Channel

CURRICULUM VITAE

NAME: Peter Hans Bernhardt
ADDRESS: Am Weinberg 30
15711 Königs Wusterhausen
Germany
DOB: Berlin - 11/25/1973

EDUCATION

1995 - 2000 University of Applied Sciences Wildau
Finished with the "Diplom-Wirtschaftsinformatiker (FH)" degree in
business information technologies with a specialization in databases and
information & communication management.

TEACHING EXPERIENCE

since 2007 University of Applied Sciences Wildau
Freelance Lecturer, Faculty of Business Administration / Business
Computing, Department of Business Computing
2000 - 2003 University of Applied Sciences Wildau
Lecturer, Faculty of Engineering/ Industrial Engineering, Department of
Telematics

WORK EXPERIENCE

since 2005 Hendrix Informationstechnik GmbH, Wildau
Project Manager and Lead Programmer
since 2003 Freelance Software Systems Engineer
1999 - 2000 Brose North America Inc., Auburn Hills, MI
Intern and Assistant to the IT-Manager

ENDNOTES

- 1 Project funded by the German federal government to exchange data using multiphop-routes between moving vehicles in the timeframe from September 2000 to September 2003
- 2 Technische Hochschule Wildau [FH] (www.tfh-wildau.de)
- 3 See Homepage of Fürstenwalde (www.fuerstenwalde-spree.de)
- 4 See TeleAtlas Homepage (www.teleatlas.com)
- 5 See Navtec Homepage (www.navteq.com)
- 6 See e.g. <http://www.spatialinsights.com/catalog/product.aspx?product=95>
- 7 See Oracle Sun Developer Network (java.sun.com)
- 8 See Microsoft .NET Framework Developer Center (msdn.microsoft.com/en-us/netframework)
- 9 See Microsoft Visual Studio Developer Center (msdn.microsoft.com/en-us/vstudio)
- 10 See Microsoft Windows Homepage (www.microsoft.com/windows)
- 11 See ThinkGeo MapSuite Destop Edition (gis.thinkgeo.com)
- 12 For more information see the NIMA Technical Report TR8350.2 by the Department of Defense, World Geodetic System 1984, Its Definition and Relationships With Local Geodetic Systems, Third Edition, National Geospatial-Intelligence Agency (earth-info.nga.mil/GandG/publications/tr8350.2/tr8350_2.html)
- 13 For more information see the website of the National Executive Committee for Space-Based Positioning, Navigation, and Timing (PNT, pnt.gov/101/gps.html)
- 14 Like SharpMap (sharpmap.codeplex.com) or MapWindow (www.mapwindow.org)
- 15 Adeunis RF Homepage (www.adeunis-rf.com)