eCoMove

Cooperative Mobility Systems and Services for Energy Efficiency

D244.210 (D2.10)	ecoStrategic Model

SubProject No.	SP2	SubProject Title	Core technology integration
Workpackage No.	WP4	Workpackage Title	Technical development of eco models
Task No.	2.4.4.2	Task Title	ecoStrategic Model (eStraM)
Authors		Isabel Wilmink, Paul van den Haak, Yusen Chen (TNO), Paul Mathias (MAT Traffic), Jaap Vreeswijk (Peek Traffic), Matthias Mann (PTV), Philipp Themann (ika)	
Dissemination level PU/PP/RE/CO		PU	
File Name		130717-DEL-SP2-WP4-D244.210-eStraM-V1.2.doc	
Due date		01-02-2012	
Delivery date		01-02-2012	
Abstract	The eCoMove project is targeting at reducing fuel consumption and CO2-emissions with 20% by supporting drivers of vehicles and trucks to drive more efficient, to plan their routes more efficient and traffic managers to manage traffic more efficiently. Information from road side units, traffic management centres, ecoMaps and also from other vehicles are exchanged to determine the best route, the most efficient driving strategy and the optimal		

Information Society Technologies	Project supported by European Union DG INFSO ICT-2009-6.1, ICT for Clean and Efficient mobility
Project reference	FP7-ICT-2009-4 IP Proposal - 247908
IP Manager	Jean Charles Pandazis, ERTICO – ITS Europe Tel: +32 2 400 0714, E-mail: jc.pandazis@mail.ertico.com

	traffic management and control strategy and settings. As essential components the ecoSituational and the ecoStrategic Model are developed within eCoMove as core technologies in SP2. They are deployed in in-vehicle and traffic management applications in SP3, SP4 and SP5. These models are developed to predict the behaviour of drivers and their vehicles in order to derive recommendations to improve fuel efficiency of single vehicles and of the whole traffic in a traffic network. The ecoSituational Model is a microscopic model aiming to describe the longitudinal behaviour of a vehicle for a short to medium time horizon. As this behaviour is heavily dependent on the traffic participants and the given road infrastructure, various sources of information need to be exploited. The model considers for example information available in the ecoMap such as speed limits or curvature and slope information to predict the velocity profile of a vehicle. The mesoscopic ecoStrategic Model simulates the whole traffic in a traffic network and is mainly designed to optimise traffic management control strategies. It takes the predictions of the multiple ecoSituational Models (available in the ecoMap at the test site traffic management centre) into account and provides information about predicted average velocities and flow back to the ecoMap. Hence, the ecoStrategic and ecoSituational Model are linked with each other. This document focuses on the ecoStrategic Model (eStraM) and summarises requirements, in- and outputs, related components and applications. It describes the major modules of eStraM and gives information about the implementation necessary to deploy the ecoStrategic Model at test sites. As not all information about test site implementation and other eCoMove core technologies, applications and components is yet available, the deliverable needs to be updated at a later stage in the project.

Control sheet

Version history			
Version	**Date**	**Main author**	**Summary of changes**
0.1	05.08.2011	Isabel Wilmink (TNO)	Main structure of document
0.2		Yusen Chen (TNO)	Added sections and content
0.3		Paul van den Haak (TNO)	Added remaining content
0.4	26.09.2011	Paul van den Haak and Yusen Chen (TNO)	Finalized draft
0.4.1	06.09.2011	Paul van den Haak (TNO)	Added content provided by IKA
0.4.2	10.10.2011	Paul van den Haak (TNO)	Added draft version of eStraM software design
0.4.3	02.11.2011	Paul van den Haak (TNO)	Added comments Peek (Jaap, Nuno), TUM (Jonas)
0.5	09.11.2011	Isabel Wilmink, Paul van den Haak, Yusen Chen (TNO), Paul Mathias (MAT Traffic)	Chapters restructured, text from Paul Matthias included, text added in several chapters, comments processed
0.6	11.11.2011	Isabel Wilmink (TNO)	Introductory chapters added, comments from partners processed
0.7	25.11.2011	Isabel Wilmink, Yusen Chen, Paul van den Haak (TNO), Paul Mathias (MAT) Matthias Mann (PTV), Jaap Vreeswijk (Imtech), Philipp Themann (IKA)	All chapters completed
1.1	01.12.2011	Isabel Wilmink, Paul van den Haak (TNO, Paul Mathias (MAT)	Peer review feedback processed
1.2	17.07.2013	Isabel Wilmink (TNO)	Minor comments from reviewers processed.
	Name		**Date**

Prepared	Isabel Wilmink (TNO), Paul Matthias (MAT), Paul van den Haak (TNO)	01.12.2011
Reviewed	Rob Olsthoorn, Jonas Lüssmann	
Authorized	Jean-Charles Pandazis	08.08.2013
Verified	Manuela Flachi	08.08.2013

Circulation

Recipient	**Date of submission**
Project partners	01.02.2012
European Commission	01.02.2012

TABLE OF CONTENTS

1. GENERAL INTRODUCTION 9

1.1. SYSTEM OVERVIEW 9
1.2. NAMING OF THE ECOSTRATEGIC MODEL 9
1.3. DOCUMENT OVERVIEW 9
1.4. FUTURE UPDATE OF D2.10 10

2. REFERENCED DOCUMENTS 11

2.1. ECOMOVE DELIVERABLES 11
2.2. ECOMOVE REFERENCE DOCUMENTS 11
2.3. ECOMOVE ONLINE DEVELOPMENT TOOLS 12
2.4. OTHER REFERENCES 12

3. INTRODUCTION TO THE ECOSTRATEGIC MODEL (ESTRAM) 14

3.1. ROLE OF ECOMODELS IN ECOMOVE 14
3.2. PURPOSE OF ESTRAM 14
3.3. NETWORK VIEWS 15
3.4. HOTSPOTS 16
3.5. APPLICATIONS 16
3.6. EXPECTED INPUTS TO THE ECOSTRATEGIC MODEL 16
3.7. EXPECTED OUTPUTS OF THE ECOSTRATEGIC MODEL 18
3.8. AIM, CONSTRAINTS AND REQUIREMENTS FOR ESTRAM 21
3.9. MAJOR MODULES OF THE ECOSTRATEGIC MODEL 22
3.10. RELATIONSHIP WITH ESIM 25

4. DATA PROCESSING AND FUSION MODULE 27

4.1. DATA SOURCES 27
4.2. PROCESSING AND FUSION 27

5. ON-LINE DYNAMIC O-D ESTIMATION MODULE 30

5.1. BASIC PRINCIPLE OF O-D ESTIMATION 30
5.2. ON-LINE O-D ESTIMATION 30

6. PREDICTION MODULE 32

7. SIMULATION MODULE 33

7.1. DYNASMART APPROACH 33
7.2. NEURAL NETWORK APPROACH (»MAT.NETSTATE«) 34

8. HOTSPOT IDENTIFICATION MODULE 39

8.1. HOTSPOT DEFINITION 39
8.2. GEOMETRY INDUCED HOTSPOTS 39
8.3. TRAFFIC INDUCED HOTSPOTS 39
8.4. HOW TO IDENTIFY A HOTSPOT 39
8.5. CALCULATION / IDENTIFICATION OF HOTSPOTS 40

9. OUTPUT MODULE 43

10. COMMUNICATION WITH THE ECOEMISSION ESTIMATION COMPONENT 44

10.1. THE ECOEMISSION ESTIMATION COMPONENT 44

10.2. USE OF THE ECOEMISSION ESTIMATION COMPONENT OUTPUT 45

11. COMMUNICATION WITH THE ECOMAP 46

11.1. INPUT FROM THE ECOMAP 46
11.2. OUTPUT TO THE ECOMAP 46
11.3. TIMESTAMP 46
11.4. POINTS FOR FURTHER DISCUSSION 46

12. IMPLEMENTATION ON THE TEST SITES 48

12.1. MUNICH TEST SITE IMPLEMENTATION 48
12.2. HELMOND TEST SITE IMPLEMENTATION 49

13. INTEGRATION, VERIFICATION, VALIDATION OF ESTRAM 51

13.1. INTEGRATION 51
13.2. VERIFICATION 52
13.3. VALIDATION 53

FIGURES

Figure 1: Visualisation of eStraM output (Dynasmart version) 20
Figure 2: Visualisation of output of the ecoEmission Estimation component............ 20
Figure 3: Modular structure of the ecoStrategic Model 22
Figure 4: An impression of the Dynasmart Software Package 34
Figure 5: The two cooperating networks: Simulation network and the error-propagation network as the controller of the first subsystem 35
Figure 6: The model consists of two cooperating subsystems: the simulation network, which is a nonlinear-layered Σ-Π-RNN, and the linear error-propagation network. .. 37
Figure 7: Draft system architecture test site Munich 48
Figure 8: Draft system architecture test site Helmond 49
Figure 9: Further illustration of the Helmond test site architecture 50
Figure 10: eStraM software design.. 52

TABLES

Table 1: Finalised eCoMove Deliverables 11
Table 2: eCoMove reference documents... 11
Table 3: Data available at the SVN server 12
Table 4: Other documents referenced 12
Table 5: Overview of the use of eStraM.. 16
Table 6: Inputs to the ecoStrategic Model 17
Table 7: Outputs of the ecoStrategic Model... 19
Table 8: High level requirements on the ecoStrategic Model 22

TERMS AND ABBREVIATIONS

Abbreviation	**Definition**
ADAS	Advanced Driver Assistance System
ANPR	Automated Number-Plate Recognition
DTA	Dynamic Traffic Assignment
DOW	Day Of the Week
eCH	ecoCooperativeHorizon
ecoTSD	eco TrafficStateData
ecoFVD	ecoFloatingVehicleData
eSiM	ecoSituational Model
eStraM	ecoStrategic Model
GPS	Global Positioning System
RSU	Road Side Unit
SPT	Signal Phase and Timing
TOD	Time Of the Day

Terms	**Definition**
hotspot	Hotspots are locations in the network (road segments or intersections) where the fuel consumption and CO2 emissions are higher than they could be. That means that traffic is not driving as efficiently as possible, due to geometry induced inefficiencies, or traffic induced inefficiencies
traffic state view	Traffic parameters (such as flows, speeds) on the links in a network, in a given time period
O-D matrix	Origin-Destination matrix, a matrix that displays the number of trips going from each origin to each destination

1. General introduction

Deliverable D2.10 is part of the subproject 2 “core technology integration” and deals with the component ecoStrategic Model. This model is developed in Task 2.4.4.2 as a core technology, to be used by the different applications, based on previous results of the project which provided use cases, requirements as well as system architecture [D21.21][D53.52].

1.1. System Overview

The eCoMove system is designed to tackle the problem of energy efficiency in road transport by applying the latest vehicle-to-infrastructure and vehicle-to-vehicle communication technologies to create an integrated solution comprising cooperative eco-driving support and eco-traffic management. The project aims to demonstrate that the combination of these new intelligent communication technologies can potentially lead to overall fuel savings and CO_2 emission reductions of up to 20 %.

1.2. Naming of the ecoStrategic Model

In the first version of the description of work for the project eCoMove the ecoStrategic Model is referred to as the “Strategic model”. Deliverable D2.1 defines the term ecoStrategic Model for this component and introduces the abbreviation “eStraM” [D21.21].

1.3. Document Overview

This document describes the general functionality of the eStraM, its in- and outputs as well as its modular structure and the relevant links to other components of the eCoMove system.

1.3.1. Intended Audience

The intended audience of this document are all stakeholders interested in the eCoMove project and especially in the ecoStrategic Model. Four main audiences can be distinguished:

- System developers in the eCoMove project who need to identify how the output of the ecoStrategic Model can be used in their applications, and what input to provide to the model and how,
- the European Commission who is supporting the eCoMove project,
- the correlated projects in the area of Cooperative ITS, and
- external stakeholders who would like to understand the usage of the ecoStrategic model in the eCoMove project.

1.3.2. Document Structure

This section shortly summarizes the different chapters in order to create an overview of the document.

Chapter 2 summarizes the referenced documents: deliverables, external documents, and available online development tools.

Chapter 3 gives a general introduction to the ecoStrategic Model in the project eCoMove and an overview of the main modules of the eStraM. Additionally the in- and outputs are summarised.

Chapters 4-9 describe the various modules in more detail: the data processing and fusion module (chapter 4), the on-line dynamic O-D estimation module (chapter 5), the prediction module (chapter 6), the simulation module (chapter 7), the hotspot identification module (chapter 8) and the output module (chapter 9).

Chapters 10 and 11 describe the communication with other eCoMove system components: the ecoEmission Estimation component (chapter 10) and the ecoMap (chapter 11).

Chaper 12 describes the implementation of the ecoStrategic Model at the test sites (Munich and Helmond).

Chapter 13 discusses integration, verification and validation of the ecoStrategic Model.

1.4. Future update of D2.10

A future update of this document is foreseen, in which some information will be included that is not available at the time of writing this version. It concerns for instance information about links to components and applications of the eCoMove system that are still under development, the implementation on test sites or parameters of which the values still need to be determined (because they need to be tuned to local conditions at a test site).

2. Referenced Documents

This chapter provides a listing of all documents referenced by this deliverable, including details known at the time of writing.

2.1. eCoMove Deliverables

This section contains deliverables (to be) produced within the eCoMove project. All public deliverables will be available for download on the eCoMove project website http://www.ecomove-project.eu/publications/deliverables/. Non public deliverables are available at the eCoMove project collaboration portal on ProjectPlace: https://secure.projectplace.com/en/Log-in. All partners in the consortium have access to the portal, account management is owned by ERTICO.

Table 1: Finalised eCoMove Deliverables

Ref	Doc	Version, Date
[D21.21]	System concept, functionality, requirements and use case description (101110-DEL-D2.1-System-concept-requirements-and-use-case-description-v06-FINAL.pdf)	V0.6, 2010-11
[D244.29]	ecoSituational Model (110930-DEL-SP2-WP4-D2_9-eSiM-V2x.pdf)	V1, 2011-10
[D243.26]	ecoMap specification - DRAFT (110511_DEL_SP2-WP2.4.3-D2.6-v0.1.doc)	V0.1, 2011-05
[D53.52]	SP5 Architecture and System Specification (DOC_SP5_WP3_Deliverable_5.2_final.doc)	V2.0, 2011-03
[D242.213]	Final report ecoMessages, DOC_SP2-WP2.4.2-D2.13-Final Report ecomessages-v0.1.doc	V0.1, 2011-11

2.2. eCoMove Reference Documents

This section contains internal documents (to be) produced within the eCoMove project. All documents are or will be available for download on the eCoMove project collaboration portal on ProjectPlace.

Table 2: eCoMove reference documents

Ref	Doc	Version, Date
[DoW]	Description of Work (DoW-PartB_eCoMove_v1.2 final-NEF.pdf)	V1.2, 2010-06
[eDH]	eCoMove Developer's Handbook (110118_DEL_SP2_D2 16_eCoMove-Dev-Handbook-v41.docx)	V41, 2011-05
[WD6.3.3]	Test Sites Planning (Draft) (20111108-DOC_SP6_WD6.3.3_WP3_TestSitePlanning-v06.doc)	V0.6, 2011-11

Ref	Doc	Version, Date
[M2.3]	SP2 Verification Plan Working Document, 110402-DOC-SP2-WP5-Verfication Plan-Final.doc	2011-04

2.3. eCoMove online development tools

To ensure collaboration between eCoMove partners multiple tools have been set in place to support software development: documentation sharing, issue management, source control and build system recommendations. Additional information is summarized in [eDH]

2.3.1. TRAC

Trac is an enhanced wiki and issue tracking system for software development projects. Trac uses a minimalistic approach to web-based software project management. It provides an interface to Subversion (see next section in this document), an integrated Wiki and convenient reporting facilities. The eCoMove TRAC is hosted and maintained by "Institut für Kraftfahrzeuge der RWTH Aachen University" (ika), and can be found here:

https://ecomove.ika.rwth-aachen.de/

2.3.2. Subversion

Subversion is a source version control system that manages files and directories and the changes made to them over time. This allows recovering older versions of data or examining the history of how data changed.

The eCoMove subversion is also hosted and maintained by ika and is split up into the different subprojects of eCoMove**.** Table 3 summarizes the SVN servers referenced in this document.

Table 3: Data available at the SVN server

Ref	Doc
[SP2_SVN]	https://svn.ika.rwth-aachen.de/trac-ecomove/ecomove
[SP3_SVN]	https://svn.ika.rwth-aachen.de/trac-ecosmartdriving/ecosmartdriving

2.4. Other References

Other documents referenced in this deliverable can be found in Table 4.

Table 4: Other documents referenced

Ref	Doc	Version, Date
[Mahmassani]	Mahmassani Hani S. and Sbayti Hayssam, Dynasmart User Manual.	May 2006

Ref	Doc	Version, Date
[Chen]	Chen, Y, REMODE – Estimation of a dynamic origin-destination matrix by traffic counts and camera data Proceeding of World Conference on Transportation Research, Lyon,2011.	1992
[Zhou]	Zhou X., List G, An Information-Theoretical Sensor Location model for traffic Origin-destination demand estimation applications	2010
[Ben-Akiva]	Ben-Akiva M. et al, Network State Estimation and Prediction for Real-Time Traffic Management	2001
[Ashok]	Ashok K., Ben-Akiva M, Estimation and Prediction of Time-dependent Origin-Destination Flows with a Stochastic Mapping to Path Flows and Link Flows	2002
[Mathias, 1999a]	Statische und dynamische Verkehrsumlegung mit Rekurrenten Neuronalen Netzen. Shaker Verlag (ISBN 3-8265-6720-X), Herzogenrath.	1999
[Mathias, 1999b]	Static and Dynamic Traffic Assignment with recurrent Neural Networks. 14th ISTTT, Jerusalem, Israel.	1999

3. Introduction to the ecoStrategic Model (eStraM)

3.1. Role of ecoModels in eCoMove

In the project eCoMove, core technologies are developed in Subproject 2 (SP2) and will be used by applications aiming to improve the energy efficiency of traffic by applying cooperative technologies. The ecoSituational Model (eSiM) and the ecoStrategic Model (eStraM) are highly important core technologies as the information from these models is the basis for most in-vehicle and roadside infrastructure applications.

The purpose of the ecoSituational Model is to determine a prospective velocity profile for the vehicle for the very near future (seconds ahead). This is essential for in-vehicle applications to determine the optimal driving strategy based on the current and predicted traffic and driving situation. The ecoSituational Model is thus a microscopic model focusing on individual vehicles and their detailed movements [D244.29].
The ecoStrategic Model is a macroscopic model. By "macroscopic" we mean that it focuses on traffic flows in the network rather than the individual vehicles.
The ecoStrategic Model will be designed in such a way that (mainly infrastructure) applications receive the information they need and can put in place traffic management strategies. This document describes the ecoStrategic Model and its interfaces with the eCoMove system.

3.2. Purpose of eStraM

The eStraM is a network model that provides information about hotspot events for CO_2 emissions or locations that have a major impact on CO_2 emissions. It is the basis for the SP5 applications (traffic management & control), but may also be used by SP3 and SP4 (in-vehicle and back-office) applications.

The eStraM is needed as a basis for the traffic management and control strategies in SP5. Where the eSiM works on the microscopic level, the eStraM translates the knowledge (included in the situational level) about what causes CO_2 emissions to be high or low to the macroscopic level (a route or a network). The eStraM will thus provide information about hotspot events that have a major impact on CO_2 emissions. These events can be related to the topography/topology of the network (such as steep inclines, areas with many traffic lights in a row etc.) or to the traffic and driving conditions (e.g. stop & go traffic, inefficient merging, etc.). The eStraM will be designed in such a way that SP5 applications receive the information they need from it through the ecoMap, and can put in place the strategies (e.g. for routing of traffic in a network or the settings of control systems) necessary to reduce CO_2 emissions. The eStraM can then determine what the effect of the hotspot events is on the CO_2 emissions, and whether hotspots in the network have been alleviated.

The main difference with eSiM is that the focus of eStraM is on traffic (i.e. a number of vehicles) in a network, rather than the individual movement of vehicles or the movement of a small number of vehicles or small platoons in a small area, e.g. an intersection. For example, eSiM focuses on the prediction of a velocity profile for one single vehicle along its most probable path, while eStraM aims at alternative paths that consist of a string of junctions. Both should be consistent in various aspects. The eStraM will build on the knowledge generated with eSiM (predicted velocity profiles of vehicles); it can also use other sources of information such as loop detector data. The traffic data and hotspots generated by eStraM are forwarded to the ecoMap. All other applications then can access the ecoMap to get information about the state of traffic and hotspots in the network. For the implementation of eStraM on a test site, we will need to determine whether there will be a local and central ecoMap running within a TMC or whether we will have multiple distributed ecoMaps running which are synchronized when necessary. See also chapter 12 on the implementation at test sites.

3.3. Network views

The eStraM has multiple views which are updated every x minutes (x to be determined, e.g. 5-15 minutes, based on constraints dictated by the traffic network size, the required information update interval, etc. The views are updated in cycles, so every cycle of length x minutes will provide an updated traffic state view which makes use of data gathered in the last x minutes. The used information is likely to be delayed a few minutes due to communication and computation times. The updates will be delivered as fixed blocks of x minutes (not as a sliding window).

1. *Current state view*
   The current state view gives an overview of the current state of the network, based on observation and modelling. For each link, the current state is known such as: speed, volumes, density, rain, temperature, road works, lane closures, etc.
2. *Predicted state view*
   This view gives a complete overview of the predicted traffic state of the current network based on historical profiles, current data, and a simulation model (given that it is unlikely that there are enough observations, i.e. enough road sections continuously monitored). These historical profiles can be generalized for homogeneous/representative subsets for different situations. For example, the average speed/density/volume on a link for a Monday has a different profile than on a Saturday. These profiles are location and time dependent. Therefore, the predicted state of the network is calculated for each location and time, using historical and current data. This process is repeated every x minutes to yield an up to date overview of the network state.
3. *Predicted state with measures view*
   The “predicted state view” only yields a prediction based on the fact that no extra eCoMove measures but only default intervention/control or management measures have been applied (e.g. standard traffic signal plans). The “predicted state with measures” should calculate the effect of new or intended measures based on predicted information. A set of measures can be called a scenario and different predictions can be computed for different scenarios. These scenarios come from the application ecoStrategies in SP5.
4. *Desired state view*
   This view will include the ideal distribution of traffic over the different routes in the network. “Ideal” means here that an optimization has taken place, in which fuel consumption in the network has been minimized, and/or hotspots have been alleviated, subject to constraints regarding e.g. travel time.

3.4. Hotspots

The eStraM is going to identify hotspots. A hotspot is a location where the fuel consumption and CO_2 emissions are higher than they could be, and could be reduced by removing inefficiencies in traffic. eStraM focuses on geometry and traffic induced hotspots caused by inefficiencies such as inefficient routing and unnecessary stops (see [D21.21] for a full list of inefficiencies addressed in the eCoMove project).

To define whether a situation is a hotspot or not, we will compare the CO_2 emissions with a certain base line for reference purposes. This baseline can be chosen beforehand as a fixed threshold, or location/time dependent from historical data. Hotspots can be defined in current and predicted views on the network. Detailed specifications can be found in chapter 8.

3.5. Applications

Table 5 indicates which SP5 Systems, Applications and components will use the ecoStrategic model (eStraM). SP3 and SP4 applications only use eStraM indirectly, as they will use recommendations from SP5 applications.

Table 5: Overview of the use of eStraM

Application	Use eStraM
ecoRoute Advice	YES
ecoPark Advice	YES - through ecoRoute Advice
ecoBalancedPriority	NO
ecoGreen Wave	YES
ecoApproach Advice	NO
ecoRamp Metering	YES
ecoSupport Merging	NO
ecoSpeed and Headway Management	NO
ecoTruck Parking	NO
ecoTolling	NO
ecoNetwork State (eStraM)	YES, is part of eStraM
ecoTraffic Strategies	YES

3.6. Expected inputs to the ecoStrategic Model

Table 6 presents the expected inputs to eStraM.

There are 2 different ways to specify the input:

- what data eStraM needs
- how it obtains these data from other applications

We will address both items in this document. First, in this section we list the types of data that can be used to feed the eStraM. Chapter 4 discusses how eStraM obtains these data sources. Also, chapter 11 discusses the connection to the ecoMap. And chapter 12 discusses how the data collection process is organised at the test sites

Table 6: Inputs to the ecoStrategic Model

Item		Resolution	Historical	Real time	Remarks
Observation	**Source**				
	Loop	1-5 minutes	minimum 6 months to 1 year, most recent data	live feed	best resolution possible, location specific, location should represent a linear position on corresponding link with linkID corresponding to ecoMap links.
	GPS, FVD	1-30 seconds	minimum 6 months to 1 year, most recent data	live feed	best resolution possible, map matched to ecoMap linear positions
	Incident/event data	instant	minimum 6 months to 1 year, most recent data	live feed	This should include accidents, road works, lane closures, events, weather, etc
	Camera	instant	minimum 6 months to 1 year, most recent data	instant	Processed data: vehicle ID, timestamp, camera location including matches with cameras on other locations to extract travel time
	Emissions	instant	Can be calculated from simulation output	instant	Every time the current view refreshes, there is a need for an up to date overview of emissions in the network
Network					

Item		Resolution	Historical	Real time	Remarks
	Zone data	-	-	-	This is geographical zone
	Node data	-	-	-	X,Y position
	Link data	-	-	-	with number of lanes, lane width, link length, capacity
Demand					
	O-D matrices	5-10 minutes	Typical O-D for different days, time sliced	-	different days are days of the week, weekends, etc.
Control data					
	Junction control	Control type, whether area-coordinated or not, offset	-	-	-
	Signal plan	Phase control, frame signal plan or other	-	-	-
	VMS location	Location, message log	-	-	-
	Ramp Metering location	Location, log and criteria/algorithm-	-	-	-

3.7. Expected outputs of the ecoStrategic Model

Table 7 presents the expected outputs of eStraM. These will be available to eCoMove applications via the ecoMap.

This output does not include emissions. eStraM provides data to the ecoMap with which the ecoEmission Estimation component (macro version) can calculate the CO_2 (and other) emissions. eStraM can use this information to identify hotspots. eStraM will provide information about the hotspots to the ecoMap (links that are identified as having a hotspot, including how severe the hotspot is). Details about the ecoEmission Estimation macro version component can be found in [D53.52].

N.B. The fuel consumption reduction that we aim to achieve is expressed in eStraM as the reduction of CO_2 emissions (when looking at the percentage change this is basically the same as looking at the percentage change in fuel consumption).

Table 7: Outputs of the ecoStrategic Model

Output name	Information object
Current State View	Flow (veh/h), Speed (km/h) and Hot Spot severity for every link in the network
Predicted State View	Predicted Flow (veh/h), Speed (km/h) and Hot Spot severity for every link in the network
Predicted State View with Measures View	Flow (veh/h), Speed (km/h) and Hot Spot severity for every link in the network after applying a measure
Desired State view	TNO approach: Ideal traffic distribution based on emissions (flows, speeds per link, using a cost function optimising predominantly for fuel consumption/CO_2 emissions) / MAT.NetState approach: Ideal traffic origin-destination route distribution for the whole network distribution based on minimized overall emissions

In addition, eStraM provides several other variables that may be relevant for applications and components (e.g. variation in speeds, number of stops). See chapter 9.

Figure 1 visualises the output of the Dynasmart part of eStraM – a view of the network with desired, current and predicted traffic flow characteristics (here, density is displayed).

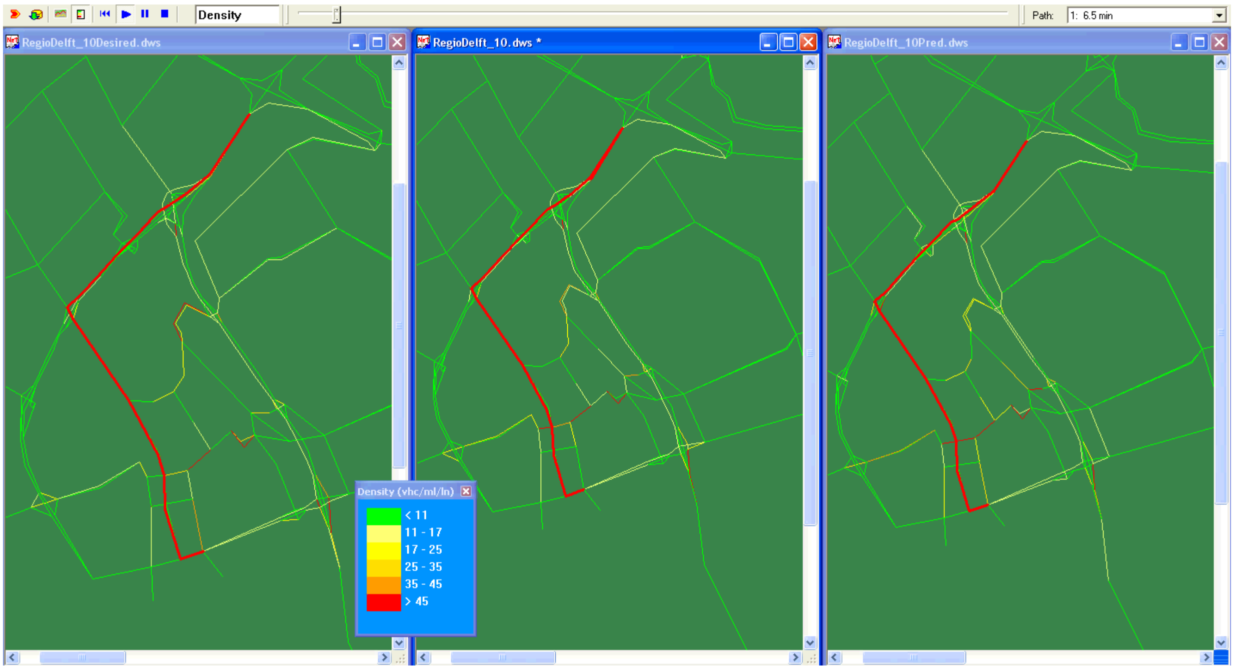

Figure 1: Visualisation of eStraM output (Dynasmart version)

The differences in densities shown for the desired, current and predicted state of the network are small in Figure 1 – a hypothetical situation, but not unrealistic. The difference between the current densities and those predicted for 15 minutes ahead are only likely to be large in transitions between peak and off-peak hours. As for the desired state, often only a small share of all traffic needs to be rerouted to achieve better results overall.

Figure 2 shows how the emission in the network are visualised by the ecoEmission Estimation component.

Figure 2: Visualisation of output of the ecoEmission Estimation component

Any other information that is output by eStraM can be visualised using the data stored in the ecoMap; eStraM does not provide this visualisation.

3.8. Aim, constraints and requirements for eStraM

The aim is to deliver a version of eStraM that can be implemented on a test site and can work on-line, delivering real-time estimations of the current and predicted state, as well as the desired state (i.e. more energy efficient distribution of traffic over the network). Depending on the data availability and the wishes of the test site operator, it can run automatically, providing output to applications and also on screen for an operator to check visually what the state of traffic (and emissions) is.

Several conditions need to be met to enable an eStraM to run on-line:

- Traffic data need to be available (from infrastructure based sensors and floating vehicle data, which are communicated to the ecoMap via ecoMessages [D242.213]).
    - eStraM needs historical data (useful for the predicted view) – we are in the process of figuring out for each test site what is really available, from which time periods, and for which kind of intervals (e.g. 1- or 5- or 15-minute data; preferably, the data should be of a higher resolution than the prediction horizon).
    - eStraM needs real-time data (useful for the current and the predicted view).
    - The real-time data should be available for eStraM without much delay. If the latency is too high (delay in transmission) this means that the predicted view cannot be provided fast enough.
- It needs to be decided whether the traffic data will be available to eStraM through the ecoMap or directly (and what this means for how much time is needed to obtain measurements).

The eStraM needs to run fast enough to make the output available in time for other core technologies, applications and components. “In time” means that the information is still useful for them. Whether this is feasible depends on several things:

- the size of the network (number of links, nodes and zones);
- the computing power of the hardware used;
- the amount of pre-processing of traffic data needed (data fusion; as described in chapter 4);
- the model used (Dynasmart is pretty fast but it depends on the network size and the computer used whether it is fast enough);
- the time applications need to determine what measures to apply and what effect they have locally, in order to include them in, for example, the traffic state view with measures.

If it is impossible to build an eStraM and interfaces that are fast enough to timely deliver current and predicted views, the alternative is to deliver an off-line version (that could be tested with real world data). Plan B would be to still use Dynasmart, but off-line and for certain test scenarios only. Plan C is to use the VISSIM model (off-line) made available in task 5.4.3 (development of simulation environment) or by the Simulation Task Force.

Table 8 summarises the high level requirements on the ecoStrategic Model. N.B. These were not included in [D21.21], therefore the numbering is not consistent with requirements included there.

Table 8: High level requirements on the ecoStrategic Model

Requirement	Description
eStraM01	The ecoStrategic Model shall describe and forecast the state of traffic on the links of a road network, and indicate whether hotspots are present.
eStraM02	The ecoStrategic Model shall base the state estimation on real-time measurements of traffic data (e.g. loop data, camera data, floating vehicle data) and historical traffic patterns.
eStraM03	The ecoStrategic Model shall interface with the ecoMap to obtain real-time and historical traffic data as provided on the test sites and to provide output data that can be used by eCoMove applications.
eStraM04	The ecoStrategic Model shall interface with the ecoEmission Estimation component either via the ecoMap or directly.

3.9. Major modules of the ecoStrategic Model

As seen in the earlier parts of this document, eStraM is going to deliver different output (views) of the traffic situations (current, predicted, desired). To compute these views, there is a need for data interpretation, prediction, fusion, etc. For this, eStraM has a modular approach in which certain modules have specific tasks which could be useful for one or more views. The modules are set up as general as possible for the purpose of re-usability. Before we explain these modules in the next chapters, we first give a process flow overview of eStraM in Figure 3.

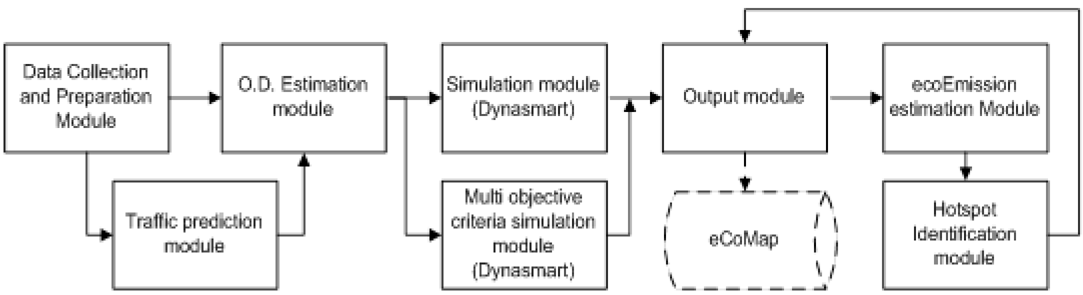

Figure 3: Modular structure of the ecoStrategic Model

For further elaboration we have divided eStraM into the following modules:

- data collection and preparation module

- dynamic O-D estimation and prediction module
- traffic prediction module
- simulation module: (mesoscopic or macroscopic) traffic model at network level
- hotspot identification module
- output module

The following sections provide brief descriptions of the different modules. The following chapters describe the modules and their functionalities in more detail and, in a separate chapter a description of how eStraM will be implemented at the test sites is given.

3.9.1. Data collection and preparation module

The eStraM will make use of several types of data sources, e.g. loop, camera and GPS data. Infrastructure based sensors provide data about specific locations (e.g. junction controller loop detectors); probe vehicles can provide data about location and time in the network (this is only a sample of the whole vehicle population).

Loop data at junctions provide mainly flow and speed and turning movements, with an interval of 1 to 2 minutes. ANPR camera (automated number plate recognition) data provide mainly travel times for a given trajectory, with an interval detailed from a few seconds to 1 minute. Probe vehicle's GPS data provide position and speed for those equipped vehicles (a small sample of total vehicle population, e.g. from 1 to 5% sampling rate), at intervals of 1, 5 to 60 seconds. These different data sources cover different geographical and time dimensions, and can be converted or fused to generate traffic states with either traffic flow or travel time. The module generates links flows that are used in the on-line dynamic O-D estimation module.

eStraM will use both historical data and real-time data (N.B. the latest data to have come in will be called real-time data, if it is less than x minutes old (e.g. 5 minutes); all the latest data are added to the database to become historical data). These data are used to calibrate and validate traffic models (and could be used for both simulation and statistical approach based modelling).

3.9.2. On-line dynamic O-D estimation module

As we want to know the traffic state of the entire network (not just links where traffic is continuously monitored), we would need to monitor all roads if we depend solely on measurements. In reality, monitoring systems cover only a very small part of the road sections in the network. Therefore, we need to obtain the unobserved locations by using the information from those monitored locations.

Either a statistical approach or a simulation model can be used to perform this task. A statistical approach builds on observed data, to establish the co-relationship between main determinant factors, to deduce unobserved traffic situations. In this case, there is no need to compute O-D matrices. But there should be sufficient observation data to be able to represent all potential combinations, to enable the calibration of statistical models. Such data needs may be a challenge given all possible combinations of needs.

A simulation model can produce the traffic states for the entire network but it needs a well-calibrated O-D matrix. An O-D matrix is mostly calibrated against the observed data and the simulation model will run to obtain the unobserved link situations. The relationship is:

- an initial O-D matrix is used to generate the mapping between O-D matrix and flow, via network use;
- this mapping is used to update the O-D matrix using traffic observations;
- the newly estimated O-D matrix is assigned back to the network and compared with the previously estimated O-D matrix and network traffic states. This is an iterative process and it will stop at a given convergence criterion.

3.9.3. Prediction module

In order to be able to obtain an O-D matrix for some time ahead (for example 15-30 minutes from now), we need a prediction module. For the current O-D matrix, we use currently available real-time data. For the predicted O-D matrix, we use real-time data as well as historical data (which show the normal patterns over the day and over the week in terms of flows and speeds) to generate predicted link flows. Then, the O-D estimation module is run with the predicted flows. The mathematical approach for this is given in chapter 6.

3.9.4. Simulation module

When the O-D matrix is available (either current or predicted), the trips can be assigned to the network. In a sense, this is a completion of the data: the estimation of the traffic state on all links in the network (not just the ones that are monitored). For this, different traffic simulation models can be used. Two approaches are elaborated: one with the Dynasmart model and the MAT.NetState approach. Obviously, the computation time of the models needs to be faster than real-time.

The simulation models can be run with the current and predicted O-D matrices (producing the **current** and **predicted** state views). All traffic control and incidents that affect link capacity values and that are expected to be valid for the prediction horizon are taken into account.

To obtain a more fuel efficient assignment (the **desired** state view), the cost function can be adapted by introducing a perceived weight for environmental components. Most current cost functions include some term for the cost of fuel, and pollution, based merely on convertible and statistical economic values. Computation with conventional cost functions show that travel time costs are far more dominant than all environmental terms combined (examples have been found of a 90% to 10% share level). If the perceived weight, which takes into account travellers' preference and perception of environmental costs when on-route, is used, the resulting distribution of the traffic reflects a traffic situation where the average emission or fuel consumption of the network (or of prioritised routes of the network) is minimal, rather than the situation as it probably can be found in the real network.

Dynasmart approach
Given the need for dynamic modelling (as needed for eco-traffic management and control concepts) and consistency with the microscopic model eSiM, a mesoscopic model approach was chosen for eStraM. The model in question is Dynasmart [Mahmassani et al, 2006]. Mesoscopic traffic models, like macroscopic traffic models, focus on the characteristics of traffic flows rather than the movements and

interactions of individual vehicles, but they allow for more detailed modelling of the dynamics of the traffic flow than macroscopic models.
Dynasmart is a mesoscopic simulation model based on individual vehicle, similar to micro-simulation but without detailed car following and lane-changing model. The model core simulates individual vehicle movement trajectory per 6 seconds of interval, considering traffic control strategy at junction, roadside and guidance information. Dynamic interaction of vehicles in different time-slices is implemented, considering a multi-criteria routing such as travel time and emission.

MAT.NetState approach
eStraM comprises a second, macroscopic, modelling approach, “MAT.NetState”, that mainly refers to the online determination of ideal network route distributions (in terms of total fuel consumption). It is a fully distributed procedure which is based on cooperating recurrent neural networks and adopts learning algorithms from “Recurrent Error-Back-Propagation”. MAT.NetState is a dynamic model that can consider spill back effects caused by incidents as well as multi-path routing. It uses any objective function for the dynamic route choice and is therefore able to take into account current fuel consumption values per link in order to optimise route choice related parameters [Mathias, 1999].

3.9.5. Hotspot identification module

The hotspot identification module looks at the state of traffic and emissions in the network (as provided by the simulation module and the ecoEmission Estimation component) and determines at which locations the emissions are higher than would be expected and/or desired, and what causes emissions to be that high. This information can be used by other eCoMove applications to trigger measures.

As eCoMove does not look at measures to reduce travel demand, hotspots in this case are related to on-trip inefficiencies (see [D21.21] for a description of inefficiencies addressed in the eCoMove project). For instance: high emissions caused by congestion on a route, high usage of a route with many intersections necessitating frequent slowing down and stopping or high usage of a route with a steep gradient.

Besides an explicit identification of hotspots (and their severity) by means of sophisticated algorithms, the CO_2 emissions will also be visualised on top of a network presentation, where regions of unusual high consumption are marked with eye-catching colours.

3.10. Relationship with eSiM

As defined, the ecoSituational Model is dealing with detail at control and vehicle level, while the ecoStrategic Model provides information at corridor and routing level.

Both models will feed each other with information through the ecoMap. eSiM stores the predicted velocities of each equipped vehicle in the ecoMap. eStraM can access the ecoMap and get all information. Vice versa: eSiM uses data provided by eStraM, such as predicted traffic flow for a link in the ecoMap which then can be accessed by the eSiM.

The data needed and provided by eSiM is already specified in the first draft document D2.44.29 [D244.29]. The reader is referred to this document for a description of eSiM.

4. Data processing and preparation module

4.1. Data sources

There are several sources of dynamic data that serve as input for the eStraM models. Available data sources include loop detectors, cameras, ecoFVD messages (vehicle generated data) and traffic light controller states. There are special interfaces to the eCoMove system foreseen in order to receive the data (see also SP5 deliverable 53.52 [D53.52]). These data need to be collected and processed for both traffic state and O-D estimation and prediction.

Each data source needs to be mapped to the network and contribute to a given purpose:

- Loop detectors: provide flow and speed at a given location (e.g. minute based)
- Traffic Light Controllers: provide traffic light control states (e.g. every second)
- Cameras: provide speed and travel time between two given points on given lanes (e.g. seconds based)
- ecoFVD: provides location, speed and actual fuel consumption of passing vehicles at any given moment (e.g. seconds based)

Loop data is point data at one given link, while ANPR camera data are measured point to point at two distinct links in the network. The common ground of these two different data sources is that they are location based.

When using these two datasets for model estimation, we use mainly observed flow. We define consistently the mapping of the percentage of the O-D pair using a given link or using a trajectory (link to link), and we are able to combine both data sources. The mathematic formulation for this is given in the next section.

For point-to-point (camera) observation, the identification rate needs to be taken into account. This may vary from location to location and from time-period to another. For the same ANPR camera, we could assume that it is stable and constant (this can be verified using historical observations).

4.2. Processing and fusion

The pre-processing and fusion of the dynamic data comprises the following:

- Aggregation of data (time): Detector values must be time averaged over equal time periods. If these periods are different, a correction must take place online.
- Aggregation of data (road topology): First of all it might be that there are detectors on single lanes that need to be combined to one cross section value for the link, as the mesoscopic and macroscopic models usually process link related values. Another important point is that the resolution of the modelling network is much lower than the real one. In order to map detector values from the real network on the model it is necessary to combine certain values.

- Plausibility of data: This first of all means a plausibility check of possible value ranges. Values outside those ranges are either corrected or omitted. We will not perform plausibility checks that analyses spatial relationships (e.g. the dependencies of neighbouring detectors are not analysed).
- Completion of data: In case of missing data, plausible values are determined on the basis of historic data (as a rule by approximation) that substitute the missing data.
- Comparison and prioritisation of redundant data: If data from two or more sources refer to the same information (e.g. speed values for the same location from ecoFVD and detector loops), it must be checked if the information concurs. If there is a contradiction there must be a rule to weigh the different data points.

Though there are several issues related to data fusion, our approach proposes a consistent framework methodology for both traffic state estimation and O-D estimation, using these data sources:

- We estimate the O-D using all these sources; Loop flow and ANPR's average speed provide consistent input to the O-D estimation. Further details are given in the section on O-D estimation (chapter 5).
- We assign the O-D to the network, by DTA technique, to generate assignment mapping (the proportion of O-D pairs using a link, for a given time slice).
- We compare the simulated flow/speed/travel times to the observed values. We iterate to make sure that the required convergence is reached regarding the simulated and observed values.

We can formulate a generic approach fusing both loop detector and ANPR camera observations:

$$c_{(l,h)} = a_{(l,h)} \sum_{t,i,j} p_{(l,h)(t,i,j)} * d_{(t,i,j)} + \varepsilon_{(l,h)}$$

where

- p = link flow proportion/assignment map, for departure time t, origin i and destination j, at link l (or ANPR-point to point trajectory) and observation interval h
- d = estimated traffic demand
- c = measured traffic flows on l for a given period h
- ε = observation error term
- a = identification rate on link l (for ANPR mainly), equal 1 if it is a single link observation

We use the Maximum Likelihood Method to estimate $d_{(t,i,j)}$ by using measurements $c_{(l,h)}$. In this, we minimize the error term.

This equation unifies both link (point) observation and trajectory (link to link) observation (ANPR). The major definition here is the mapping that should take into account both point to point with ANPR and one point for loop detector data.
The above equation also has consequences for how each data set is organised and how an assignment map is built. In the whole network representation, both loop and camera use the same link/network structure, so that they are consistent.

The module outputs link flows per time slice, which are used in the O-D estimation.

5. On-line dynamic O-D estimation module

5.1. Basic principle of O-D estimation

The basic principle of O-D estimation is to keep the original O-D matrix structure and to fit the observations (traffic counts, floating vehicle data or camera). Applications to various static and dynamic O-D estimations also provide some feedback for the fine-tuning of the algorithm. The objective function is as follows:

$$\min Z = \left\{ (1-w)\sum_{l,h}[\sum_{t,i,j} p_{(l,h),(t,i,j)} \cdot d_{(t,i,j)} / c_{(l,h)} - 1.0]^2 + w\sum_{i,j}[\sum_{t} d_{(t,i,j)} / g_{(i,j)} - 1.0]^2 \right\}$$

where

- w = a positive weight between 0 and 1
- p = link flow proportion, for departure time t, origin i and destination j, at link l and observation interval h (link can be also an ANPR point to point trajectory)
- d = estimated traffic demand
- c = measured traffic flows per loop or ANPR monitoring camera
- g = historical static demand

This is implemented in REMODE [Chen, 2011], which will be applied to estimate dynamic O-D matrix from traffic counts and FVD (floating vehicle data) as well as camera data. It takes an assignment map from an external program (static or DTA), and provides the updated matrix based on its original one.

With O-D matrices and network traffic state, we are able to guide traffic on alternative routes, as we know the flow per origin and destination and we are able to distribute them into various paths. In case of a statistical approach, there is no O-D information thus no such routing information, albeit turning fractions are provided externally.

5.2. On-line O-D estimation

For eStraM, we need on-line O-D estimation (and prediction, see chapter 6). This requires several steps. A precondition for the operation of the eStraM network models is a data basis that contains O-D matrices for certain day types (e.g. normal working day) that are temporarily resolved to at least one hour slides. These one hour matrices serve as the initial O-D matrices that need to be adapted. The online correction of the O-D flows in a traffic network is then performed on two different levels, each of which represents another time horizon.

- The more elementary level is the adaptation of the network inflows of the simulation part of the model according to dynamic data from loop detectors, assuming that all (or almost all) network inflows are measured by detectors. This allows one to adapt to short-term varying inflows (e.g. every minute) while maintaining the relative weights of the O-D relations.

- A more sophisticated level is the correction of the actual basis O-D matrix according to all current detector values of the network, which can be gained through the use of measurement equations / objective functions. The O-D estimation applied in the eCoMove context follows the known measurement equation [Ben-Akiva]:

$$Y_h - Y_h^H = \sum_{p=h-p'}^{h} a_h^p * (X_p - X_p^H) + \zeta_h$$

where

- H is index for historical data
- h is the time interval (current time)
- Y is traffic measurement (flow)
- X is O-D matrix (demand)
- p' is the number of time intervals corresponding to the longest trip
- a_h^p is the assignment mapping of X to Y
- ζ_h is error term

The equation above is a measurement equation, referring to traffic state versus traffic measurement, and p' is the number of time intervals corresponding to the longest trip. This implies that the equation is using multiple observation intervals to cover the whole journey and to make the prediction more stable.
The computation is such that the correction will not lead to a new O-D matrix that is too far away from the original one (in terms of a given matrix), so as to avoid ending up with an abnormal solution.

The on-line O-D estimation yields O-D matrices that are used in the Simulation Module (see chapter 7).

6. Prediction module

For the prediction, we take into account both the latest measurements and the historical profiles of traffic over the day.

The prediction is based on a standard Kalman Filter (see the first equation below), referring to the prediction of a step ahead [Ashok].

$$X_{h+1} - X_{h+1}^{H} = \sum_{p=h-p'}^{h} f_h^p * (X_p - X_p^H) + w_h$$

Where

- f_h^p is the transition matrix of effect $(X_p - X_p^H)$ on $(X_{h+1} - X_{h+1}^H)$
- w_h is the random error term
- h is the time interval (current time)

One-step prediction (h+1) is then given by

$$\widehat{X}_{h+1|h} = \Phi \widehat{X}_{h|h}$$

Φ (= f_h^h) in (in the above equation), is the transition factor that is first estimated for the current state (time interval h) with available observation data, and then updated and cumulated (using multiple points, e.g. 5, 10 and 15 minutes ago) during the rolling horizons (real-time), meaning that it takes both effects of offline (historical data, giving the normal profile of link flows over the day) and online calibration (using the latest available data). Further specification is needed; this will be done based on the available data at the test sites. At the moment, the plan is to use polynomial functions to capture the daily profile.

The equation above is a spatial and temporal equation, capturing trip-making and their variations over time and space. One simple way of incorporating structural relationships (the historical profile over the day) is to include all the prior information and estimation into the consideration. A Kalman filter fits well in this context.

Why use the Kalman filter?

Kalman filter is an algorithm for efficient inference in a linear dynamic system, in which the state space of the latent variables is, continuous. Its purpose is to use measurements observed over time, containing random variations and other inaccuracies and to produce values that tend to be closer to the true values of the measurements and their associated calculated values.

The output of this module is estimated link flows for the future time period, which can be used to predict the future O-D matrix.

7. Simulation module

This chapter discusses the network models that will be used. A network model needs to be made for each test site on which eStraM will be deployed. The road network (geometry, speed limits etc.) and traffic control on this network need to be modelled. Also, the network model will include a traffic assignment module which can be run to obtain the current/future/desired traffic state.

7.1. Dynasmart approach

Dynasmart [Mahmassani] is a simulation tool for dynamic network analysis and evaluation. This simulation environment models the evolution of traffic flows in a traffic network resulting from the travel decisions of individual drivers. Dynasmart is a mesoscopic model which uses a detailed network representation as a microscopic model. This includes the node and link structure. A node represents a junction, with coordinates and traffic light settings (type of control and control phases). A link is a road section between two adjacent nodes, indicating the type, length, number of lanes, capacity, etc. On these links, control measures (speed limit, ramp-metering) as well as incident and road-work, route guidance (VMS), and toll can be defined.

Dynasmart has a number of features. The micro-simulation simulates individual trip-maker decisions such as route, departure time and mode including user responses to varying types of information as well as higher order activity participation and sequencing decisions. Dynasmart is mesoscopic in the sense that it moves individual particles according to robust macroscopic traffic flow relations. The particles (users) can be represented in multiple classes in terms of (1) operational performance (trucks, busses, passenger cars), (2) information availability and type, (3) user behaviour rules and response to information. Traffic processes at signalized junctions can be represented under a variety of operational controls (this is critical for urban congestion and ITS). These features make Dynasmart applicable for determining network congestion pricing schemes and evaluating ITS deployment alternatives and their geographic coverage (VMS locations, information strategies, etc). Figure 4 shows an impression of the Dynasmart simulation software.

Multi-criteria routing
The multi-criteria routing module will be used to change route choice in the network, in order to reduce inefficient route choice (in terms of fuel consumption and CO_2 emissions).
The literature shows that travellers consider eco-routing explicitly (and therefore do not only base their route on expected travel times). This is not purely based on economic calculation (which would emphasise travel time), but on the traveller's perceptions and preferences, which means that perceived values of time and eco-routing are larger than computed ones. We incorporate eco-routing behaviour in Dynasmart by modifying the cost function that is used in the traffic assignment. Simply put, in the assignment the "cheapest" routes (in terms of generalised costs, of which fuel costs may already be a part) are chosen. We will test varying costs of fuel consumption and CO_2 emissions, with the intent of having fuel consumption and emission costs influence the routes. By doing this, we aim to decrease the total fuel consumption in the network.

This module should provide information to applications that give route advice. It is up to these applications to decide how to translate changes in routes to route advice for a particular vehicle.

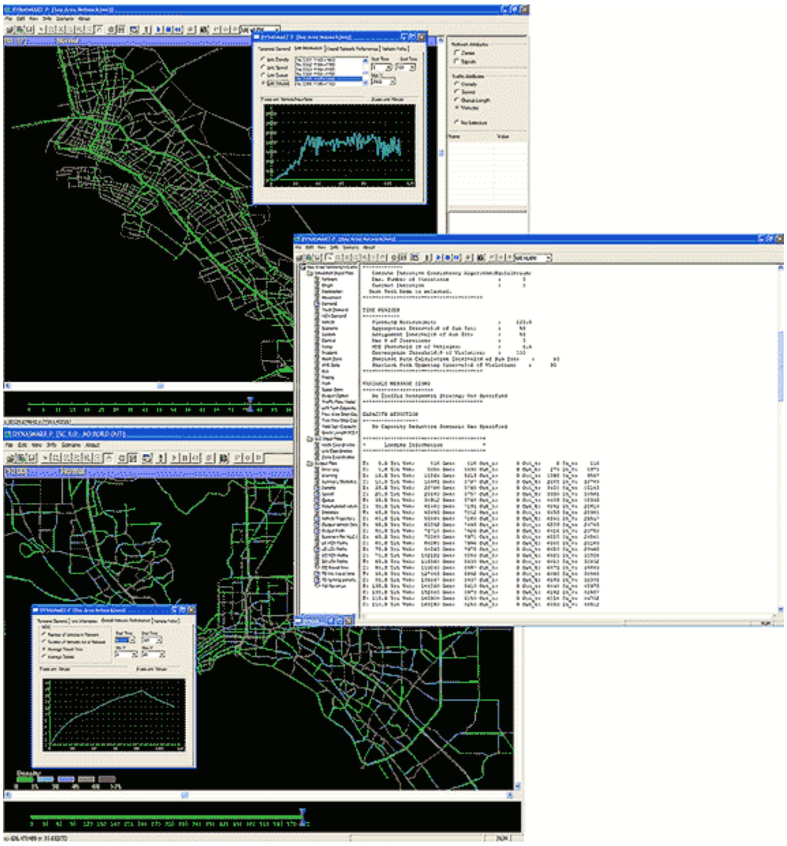

Figure 4: An impression of the Dynasmart Software Package [Mahmassani]

7.2. Neural Network approach (»Mat.NetState«)

The Neural Network approach for integrated dynamic traffic assignment of road networks is based on an interpretation of the road network as a special recurrent neural network (RNN) consisting of Σ-Π-neurons, i.e. computational units which perform sum and product operations. The dynamic traffic assignment part of the model, which is realized through the cooperation of a nonlinear recurrent neural network and an enhanced version of a linear recurrent error-propagation network, works with any given objective function and is capable to reproduce congestion phenomena like spill back effects. The traffic control part cares for optimal determination of link capacity values. This is obtained by another linear recurrent error-propagation network that realizes gradient descents at certain intersection capacity surfaces. Because of the neural network structure, the model can easily be implemented on a multi-processor computer in order to handle very large networks.

General structure of the model

The dynamic traffic assignment task is interpreted as a control problem. The road network being regarded as the network to be controlled (simulation network), and the optimal route choice (error-propagation network) as the controller of the first subsystem (see Figure 5).

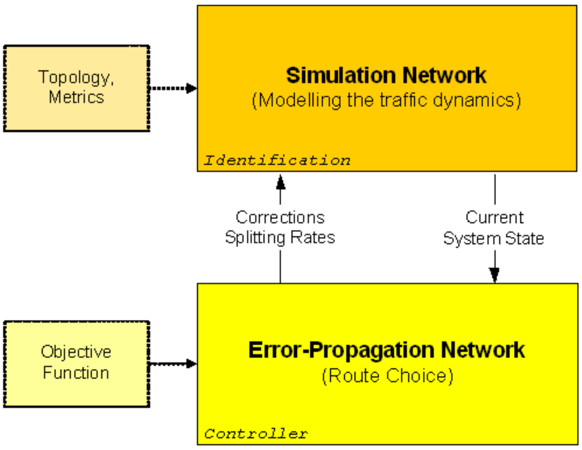

Figure 5: The two cooperating networks: Simulation network and the error-propagation network as the controller of the first subsystem

The Simulation Network

The simulation network is a nonlinear structured recurrent neural network. Its construction is based on the observation that in fact road traffic systems and artificial neural networks exhibit many similarities, as they both are some kind of "transportation system". The general approach therefore is to map topology and dynamics of the traffic system as directly as possible onto a neural network model. Following this point of view, the neurons of the simulation network are identified with the links of the road network. The neuron activation states correspond to the road network's traffic load and their weight values can be viewed as the intersection flow splitting rates. It works on the link load vector and determines the link in- and outflows with appropriate load dependent functions. Its temporal evolution equation can be interpreted as a Σ-Π-network.
With the above definitions and assumptions we now define a discrete-time model for road traffic networks. Since the model shall be link oriented, it focuses on the mathematical formulation of the link dynamics. The dynamical equation of link *k* for vehicles with destination *d* is described as follows:

$$x_k^d(t+\tau_x) = x_k^d(t) + \tau_x \left(r_k(t) \sum_{i \in \mathcal{L}_U(k)} w_{ik}^d(t)\, s_i^d(t) - s_k^d(t) \sum_{i \in \mathcal{L}_D(k)} w_{ki}^d(t)\, r_i(t) + q_k^d(t) \right)$$

where

- τ is the discrete time step of the evolution,
- the *x*-values are the traffic densities of the vehicles with destination *d* on link *i*,
- the *w*-values are the splitting rates at time *t* for the traffic flow with destination *d*,
- the *s*-values are the exit flows from link *i* with destination *d* at time *t*,
- the *r*-values are the link restrictions of link *k* at time *t*, and
- the *q*-values are the source traffic inflows into link *k* with destination *d* at time *t*.

The Error-Propagation Network

The error-propagation network is a linear Recurrent Neural Network that is transposed to the simulation network. The role of this network is to serve as a controller for the former network: it receives the first network's actual state, calculates gradient information of a given objective function, and eventually gives back coupling weight corrections (i.e. corrections of the flow splitting rates) to the first network. A modified version of the recurrent back-propagation algorithm (RBP) is employed here that takes into account the Σ-Π-structure of the simulation network.

The objective function has the following form (a sum of link local functions depending on the state of that link):

$$E(t) = \sum_{k} E_k(x_k)$$

From this and the use of gradient descent it is possible to construct the error-propagation network that follows the following dynamical rule:

$$v_{ik}^{le} := \frac{1}{K_k^e}\left[\left(w_{ki}^l F_{ki}^l - w_{ik}^l H_{ik}^l\right) - \delta_{ik} \sum_{j \in \mathcal{L}} \left(w_{kj}^l F_{kj}^l - w_{jk}^l H_{jk}^l\right) + \delta_{el} w_{ki}^e G_{ki}\right]$$

where the *F, H, G* and *K* values can easily be computed out of the states of the simulation network.

Eventually, the adjustments of the weights of the simulation network can be made by using the following formula:

$$\Delta w_{pq}^d = \varepsilon \left(y_q^d - y_p^d\right) s_p^d \, r_q$$

Operation Mode

The whole procedure is defined as follows:

1. Simulation network: Develop the simulation subsystem a certain number of discrete time steps according to its dynamical rules
2. Simulation network: Transmit the actual state to the error propagation network
3. Simulation network: Wait for the error propagation results
4. Error-propagation network: Compute the error-propagation network weights by using the current state of the simulation network
5. Error-propagation network: Develop the error propagation subsystem discrete time steps according to its dynamical rules
6. Error-propagation network: Transmit the states to the simulation network
7. Simulation network: Update the simulation network weights using the states of the error-propagation network

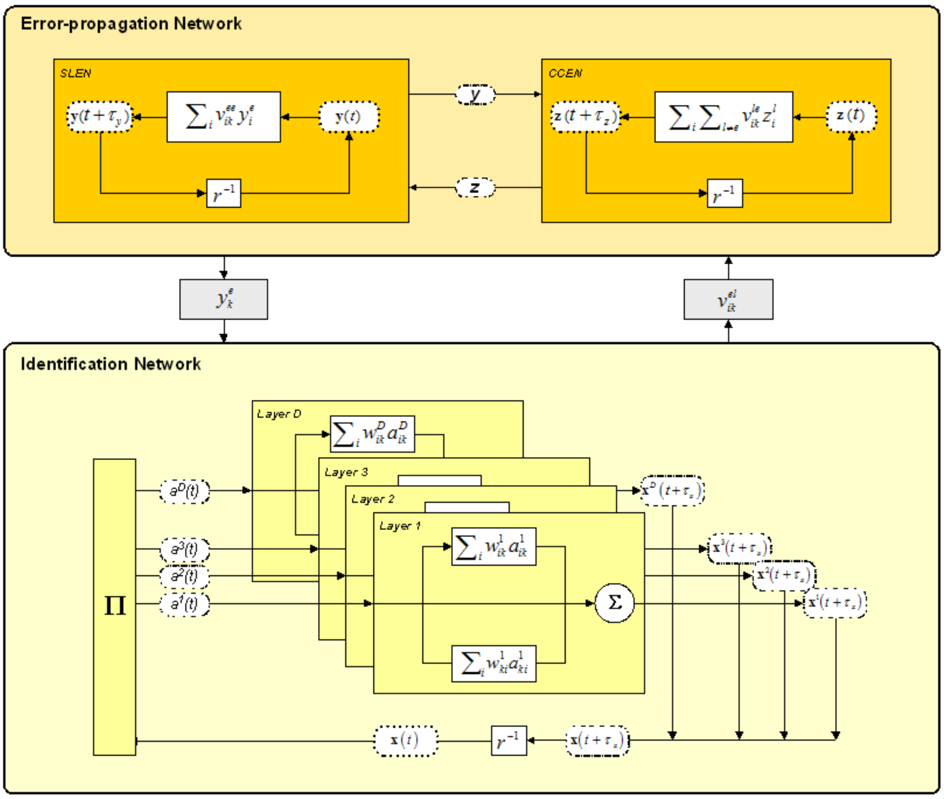

Figure 6: The model consists of two cooperating subsystems: the simulation network, which is a nonlinear-layered Σ-Π-RNN, and the linear error-propagation network.

Multi-criteria routing

The NetState model uses a linear error-propagation network, which is transposed to the simulation network, to dynamically determine the optimal o-d-routes in the network. The routing part of the model utilises a gradient descent algorithm on an objective function that expresses the overall goal to minimise the totality of the fuel consumption / CO_2 emission in the network.

The role of the error-propagation network is to serve as a controller for the simulation network: it receives the first network's actual state, calculates gradient information of a given objective function, and eventually gives back coupling weight corrections (i.e. corrections of the flow splitting rates) to the first network.

The objective function has the following form (a sum of link local functions depending on the state of that link):

$$E(t) = \sum_k \alpha_k FC_k(x_k)$$

The α_k are dynamic weights that reflect the macroscopic fuel consumption characteristics of the road sections / links indicated by the k-indices. For instance, a road section with an active ecoGreenWave or ecoBalancedPriority would have smaller weights assigned to it than it would have without these applications. The functions FC_k are realised by the ecoEmissionEstimation component that derives estimations of fuel consumption / CO2 emissions on the basis of traffic states, road geometries, current traffic control states and detailed vehicle type related consumption values.

8. Hotspot identification module

8.1. Hotspot definition

Hotspots are locations in the network (road segments or intersections) where the fuel consumption and CO_2 emissions are higher than they could be. That means that traffic is not driving as efficiently as possible, due to:

- geometry induced inefficiencies, or
- traffic induced inefficiencies

For CO_2, there is no regulation like there is for air quality. There are overall targets (mostly at the national level) but there are no limit values that should not be exceeded locally (along a road). This means that there are no set maximum values to which the emissions on a road section can be compared. Therefore, the aim is to determine whether the emissions per km driven are higher than what they would be under favourable circumstances, and what causes the emissions per km driven to be higher than this. This should be done separately for passenger cars and goods vehicles (because goods vehicles emit much more than passenger cars, so a higher share of goods vehicles on a link will automatically increase the total emissions).

A choice has to be made as to when emissions are deemed to be too high, and what could be defined as “favourable circumstances”. This is discussed in section 8.5, which discusses how to identify a hotspot.

In the following sections, where it says “emissions”, we mean “fuel consumption and CO_2 emissions”.

8.2. Geometry induced hotspots

There are parts of the road network where fuel consumption and CO_2 emissions will be relatively high due to the geometry (irrespective of how much traffic there is). Inefficient route choice could lead to traffic using routes with (steep) gradients, many sharp curves or a high number of intersections where traffic needs to slow down or frequently come to a complete stop. Also, the speed limit or rather the realised speeds are relevant: driving very slow or very fast is emission inefficient.

8.3. Traffic induced hotspots

Traffic induced hotspots occur for various reasons. In uncongested networks, badly maintained or unsynchronised traffic lights can cause a larger-than-necessary number of stops. In congested networks, traffic jams and long queues at intersections (signalised or not) can lead to substantially higher fuel consumption and CO_2 emissions.

8.4. How to identify a hotspot

If a hotspot is defined as a location where the fuel consumption and CO_2 emissions are higher than they could be, then that implies that we need to define what they could be in an optimal situation. This is needed to compare fuel consumption and CO_2 emissions in a current or predicted situation to how low they could be if traffic followed the most efficient route and intersections function (i.e. only a small part of traffic needs to slow down considerably, to stop completely or to alter a route). It is also interesting to compare the current situation with the "average" situation (in that season of the year, that day of the week, that time of the day), but one has to bear in mind that the average situation could constitute a hotspot.

Geometry induced:

- compare emissions on a road segment to what they would be on an "ideal road section", e.g. straight road section with no inclination, speed limit 70-80 km/h (and no intersections or other features that slow traffic down). .

Considerations:

- the "ideal road section" with very little traffic (only a few vehicles, hardly any interaction between them) on it would give the bare minimum for emissions (in g/km, for the vehicle categories distinguished).
- the "lowest" fuel consumption and CO_2 emissions in a network can be calculated by running the model with very little traffic in it. That way, it can be seen how high the fuel consumption and CO_2 emissions are in different locations (in g/km, or g/s) when vehicles can drive as they wish (are not hindered by other vehicles). Specific routes may have to be specified for this.
- In some cases, these routes will not be chosen by traffic because they are slower routes. But this is probably not always the case. It is interesting to see what the "penalty" is of using hilly routes or routes with many sharp curves. Also, it is interesting to see the difference between different types of intersections (e.g. signalised, right of way, roundabout) and between routes with and without at-grade intersections.
- This needs to be done per vehicle type distinguished in the emission model (at least passenger cars, heavy goods vehicles).

Traffic induced:

- compare fuel consumption and CO_2 emissions per vehicle per km with what they are in an uncongested situation
  - o traffic jams compared to free flow driving
  - o long queues /waiting times at traffic lights compared to normal queues / waiting times.
- compare emissions to what they are on average on that link in that season, on that day of the week and in that period if the day (e.g. per hour or per 15 minutes).

We want to know if the emissions are higher than they could be given the local geometry. If the emissions per vehicle are higher, this means that the traffic conditions are less than ideal. If the emissions are only a little bit higher, you may not want to activate measures. However, in the end you might want to do something even in the "average" situation, therefore you need to compare to an uncongested situation. In addition to that, we can check if the emissions are higher than in the "average" situation, to get more insight into just how bad the situation is.

8.5. Calculation / identification of hotspots

Because Dynasmart is a model with link based outputs (traffic characteristics for road sections of various lengths), a hotspot will also be defined at this level. Short links can be used, but it is advised to use a link length of more than 6 seconds of travel time because the output of Dynasmart is written per 6 seconds.

There are several approaches to defining quantitatively what constitutes a hotspot. For each link, the following can be calculated (based on the vehicle fleet composition and traffic parameters found, and possibly the link type):

- Emissions in g/km for the average vehicle (to be calculated per vehicle category) on a link.
- The total emission in kg/km for all traffic on the link in the period considered.

The comparison to a baseline can be done as follows:

- Compare the current/predicted emissions to the lowest possible emissions for the average vehicle (to be calculated per vehicle category) – for the lowest emissions we can model vehicles driving 70 km/h on an empty, straight road with no inclination. Comparison to this baseline would show how high, relatively, the emissions per vehicle are, irrespective of how many vehicles were present on the link. It shows action may be needed because vehicles are driving inefficiently. It does not show whether this is because of the amount of traffic or because of geometry induced inefficiencies.
  - o we need to define the threshold for when it becomes a hotspot
  - o e.g. percentage change <2% is no problem, 5-10% increase is moderate hotspot, >10% increase is hotspot (thresholds to be determined based on findings from simulation study used to determine emission factors for different situations, thresholds presented here may be too low)
- Compare the current/predicted emissions to the emissions under uncongested (free flow) conditions and to the normal or average emissions for that season of the year, that day of the week (DOW) and the time of the day (TOD) – for a vehicle of certain vehicle category or for all traffic on the link. The comparison to the average emissions could be used to show that emissions on a link are considerably higher than is normal for that period. E.g. a situation where on Tuesday between 8:15-8:30h total emissions are 25% higher than on an average Tuesday morning between 8:15-8:30h.
  - o thresholds need to be defined here as well.
  - o to be considered: only comparing it to the "average situation" means that there is no incentive to improve on the average situation. Normal peak hour total emissions would then be called acceptable. This is why we also need to compare with the emissions under uncongested conditions (emissions may then be higher than the lowest possible emissions, as traffic may choose fuel inefficient routes).
- The absolute difference can also be determined. In that case, we could look at:
  - o the change in emissions in g/km for a vehicle of a certain category on a link (increase of more than 25g/km is a hotspot? This would mean that we need to determine what a considerable increase is).
  - o the change in total emissions in kg/km for a link. It needs to be determined what would be a substantial change.

8.5.1. Representation of the hotspots

We can represent the severity of hotspots in several ways:

- percentage difference between the current/predicted situation and the "lowest possible emissions" and/or uncongested situation
- absolute difference between the current/predicted situation and the "lowest possible emissions" and/or uncongested situation
- percentage difference between the current/predicted situation and the average situation on that link in that season, on that day of the week and in that time of the day
- absolute difference between the current/predicted situation and the average situation on that link in that season, on that day of the week and in that time of the day
- only locations (links/nodes) with a high difference between the current/predicted situation and the "ideal" and/or uncongested situation get marked (e.g. yellow-orange-red for increasing "hotspotness")

When applications choose to visualise a hotspot, they can, for instance, show them in one colour, i.e. all hotspot links are coloured red (regardless of how much of a hotspot they are). Or the colour can be varied according to how much worse than normal or than the ideal situation the emissions are (see for an example Figure 7).

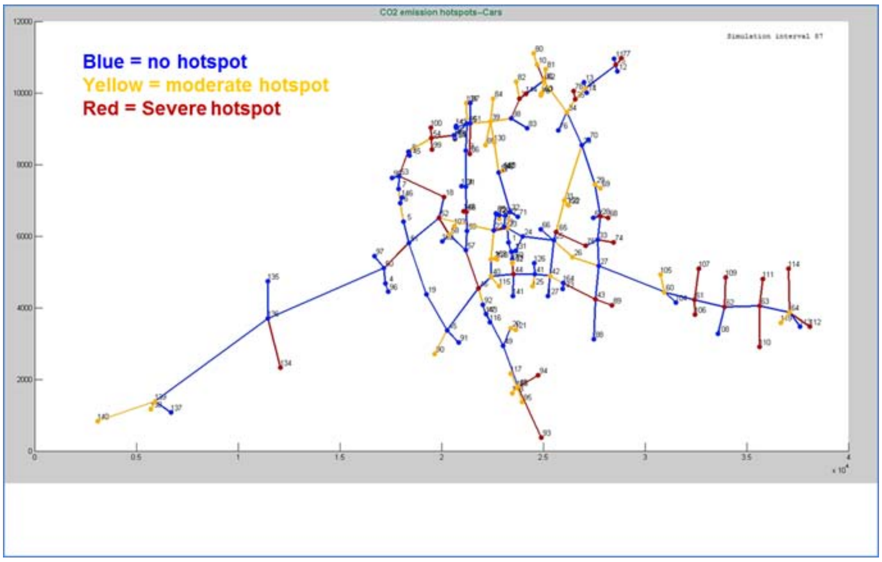

Figure 7: Visualisation of the hotspot categorisation

9. Output module

After defining the hot spots in the hot spot identification module for every traffic state view, eStraM is ready to output the traffic state views (current view, predicted view, desired view). This process is done by the output module, which is actually planned to be an interface between the eStraM and the ecoMap. Currently, we assume that the ecoMap contains the input data, which is described in section 3.6, and that we can output the traffic state views in the ecoMap.

The output module will output the following data every refreshment cycle:

- average speed per link in km/h,
- average intensity per link in veh/h,
- average densities per link in veh/km,
- hotspots (for every link there is an hot spot attribute which contains a value which represents the severity of the hot spot (if there is any),
- standard deviation of the speed per link based on the measurements taken in that cycle,
- number of stops or queue length per link.

This data will be output for all traffic state views.

The way eStraM delivers its output depends on the implementation at the test sites. The plan is to let eStraM communicate with the eCoMap to deliver its traffic state views and receive its input data. More information about the communication with the ecoMap can be found in Chapter 11.

10. Communication with the ecoEmission Estimation component

10.1. The ecoEmission Estimation component

10.1.1. Description of its use with eStraM

The eCoMove system comprises a component ‘ecoEmission Estimation’ to compute emissions of either individual vehicles or of traffic flows. For eStraM, the emphasis lies on calculating the emissions of traffic flows on the different links that comprise the traffic network that is considered in eStraM.

This means that the meso/macroscopic version of the ecoEmission Estimation component is used. It calculates the emissions on each link of the network on the basis of traffic parameters, such as traffic flow and speed data (and/or level of service – is traffic congested or free flow?), assumed distributions of vehicle types and road geometry elements (such as intersection type, gradient, curviness). All these data need to be communicated to the ecoEmission Estimation component.

10.1.2. Variables to be exchanged

The precise variables to be communicated will be determined once the ecoEmission Estimation component has been defined to the level of input data needed.

Examples of link data (for the current/ predicted/predicted with measures/desired state views, per time interval) communicated to the ecoEmission Estimation component include:

- average speed,
- standard deviation of speed (using the averages of each time step in the time interval),
- flow,
- shares of vehicle types,
- number of stops or queue length,
- link capacity (available in ecoMap),
- link length (available in ecoMap),
- speed limit or free flow speed (speed limit available in ecoMap).

The ecoEmission Estimation component will produce (per link, per time interval) total emissions (at least CO_2, also probably NO_x, PM_{10}) for the link (distinguishing vehicle types), with which also the average emissions per vehicle per km can be computed.

10.1.3. Communication between eStraM and the ecoEmission Estimation component

The communication between eStraM and the ecoEmission Estimation component can be done directly or via the ecoMap. It will depend on the implementation on the test site which route will be taken.

10.2. Use of the ecoEmission Estimation component output

The ecoEmission Estimation component will be used in the context of the hot spot identification module and the multi-criteria routing module.

For the hot spot identification module, it is necessary to compare the CO_2 emissions per link with a baseline, as explained in section 3.9.5. The CO_2 emissions per link are therefore retrieved from the ecoEmission Estimation component.

The multi-criteria routing module incorporates the actual CO_2 emission values into the link dependent objective functions. With this we can put more emphasis on emissions in the objective functions (compared to the regular functions that emphasises travel times). The simulation module of eStraM takes the fuel consumption per link as input and computes the desired route distribution.

11. Communication with the ecoMap

eStraM needs input data, as described in section 3.6, to be able to compute and provide traffic state views. These traffic state views are the output of eStraM, as described in section 3.7, and are used by SP5 applications. Currently, we assume that the input data as well as the output data are communicated with SP5 applications (or other parts of the eCoMove system) through the ecoMap. The ecoMap contains static information (link lengths, link capacity, etc) as well as dynamic information, allowing applications / functions to write information into the map for other applications / functions to retrieve. Both static and dynamic data is available via a single common interface [D243.26].

11.1. *Input from the ecoMap*

From the ecoMap, we plan on using the following observation data:

- Loop data
- FVD, GPS
- Incident/event data
- Camera data
- Emissions

The first four data elements should be stored per link in the ecoMap, so that eStraM can extract these data elements and use it in the DynaSmart simulation environment. It is not yet decided how the emissions from the ecoEmission Estimation Component will be communicated. Currently, we assume that the ecoEmission Estimation Component will feed the ecoMap with emissions per link, given that there is an implemented ecoMap available. If this is not the case, there is a need for a direct connection from the eStraM to the ecoEmission Estimation Component.

11.2. *Output to the ecoMap*

The output module of eStraM will output the traffic state views in the ecoMap, given that there is an ecoMap available and it contains functionality to save the data attributes speed (km/h), density (veh/km), intensity (veh/h) and hotspot severity per link in the network. The ecoMap is expected to contain functionality to store and retrieve Java Generic Types in a list per link as an attribute. This means that eStraM could define its own class structure to store traffic state views. If this is done, we need to make a specification of the structure of this class. A more elegant approach would be to define a Traffic State View class and include this in the API of the eCoMap. Both approaches are currently under discussion.

11.3. *Timestamp*

eStraM will use the common timestamp defined for the eCoMove system.

11.4. *Points for further discussion*

In order to decide whether eStraM is going to receive input from and send output to the ecoMap or directly to other components/applications, there are several topics that need further discussion and finalization.

For instance, sending/receiving data to/from other components/applications via the ecoMap could introduce a delay. This delay could make it difficult for eStraM to deliver the traffic state views or receive input data in time.
Furthermore, since we have no implementation of the ecoMap yet, this part of the development of eStraM is halted. This means that we are not capable of sharing data via the ecoMap with other partners, which causes problems for the calibration process of our models with test site data (which is a time consuming task).
Finally, if there is not going to be an implementation of the ecoMap on the test site or if the data needed by eStaM are not available in the ecoMap in the end, the data need to be collected via other partners. These data need to be read in, processed and structured, which includes the time consuming task of programming data parsers for different input files. Since there is a shared need of different partners, it is highly desirable to have an ecoMap implementation available, which contains all necessary data elements of the test site Munich and Helmond.

12. Implementation on the test sites

In the following a first version of the physical integration of the ecoStrategicModel into the test sites is presented. As the implementation of eStraM on the test sites progresses, the descriptions will be updated. This includes decisions on which parts of eStraM be will tested at which test site.

12.1. Munich test site implementation

Figure 8 describes the draft system architecture from TS Munich from the perspective of the ecoStrategic Model. Thus only the relevant information (upper part of the figure) and the interaction with the central turntable for dynamic information – ecoMap – as well as the functional aspects of eStraM are considered. The box “Applications” is a place holder for any applications using the outcome of eStraM processing components (e.g. “Current traffic state”) which can be found within the ecoMap. It is still under discussion if some of the applications will run on separate servers, but this is in the end a test site and partner specific design decision.

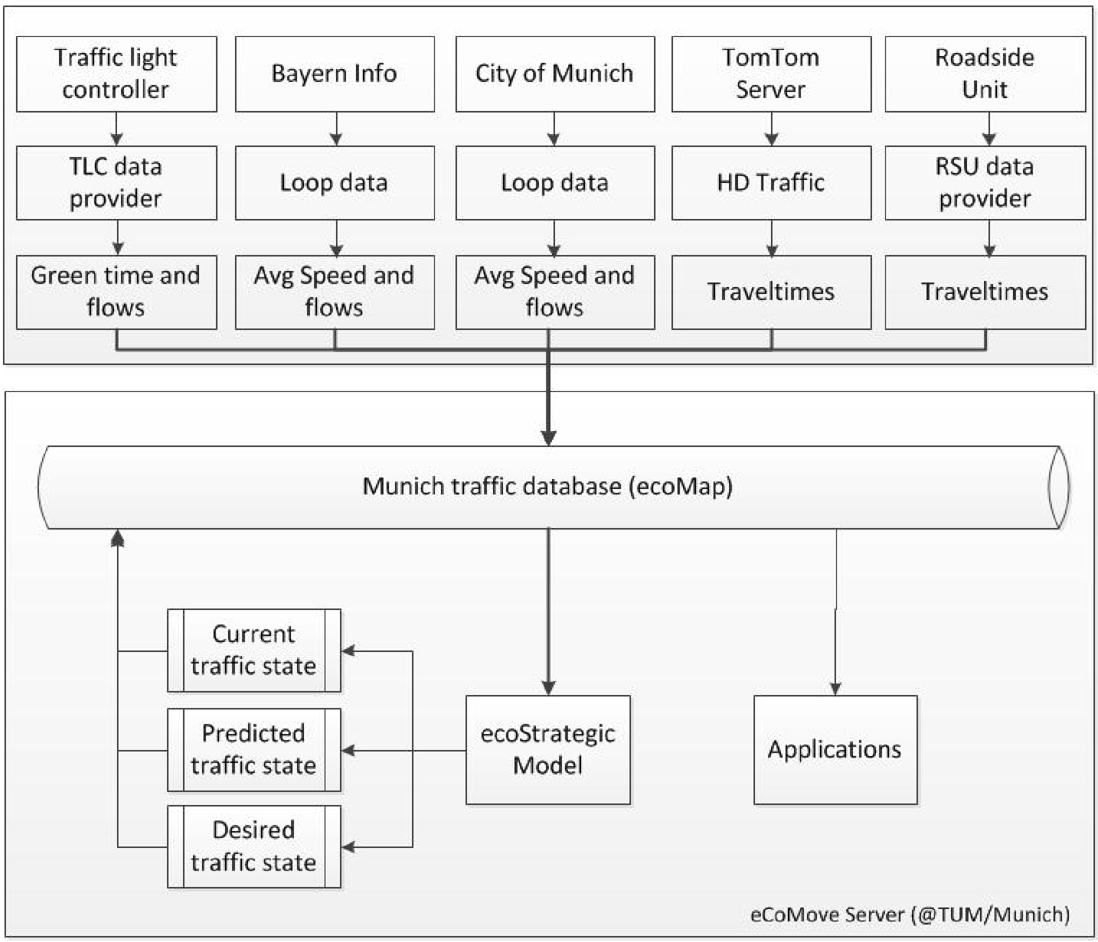

Figure 8: Draft system architecture test site Munich

The required information such as loop data or traffic light control data will be collected in real-time from various systems, pre-processed and converted to an ecoMap format and written into the ecoMap of the central eCoMove Server.

As the applications development is still in progress, the decision about which views will be implemented and what data will be provided to applications active in Munich (via the ecoMap) has not been made yet.

12.2. Helmond test site implementation

Figure 9 describes the system architecture for the Helmond test site. The architecture is slightly different than for Munich. The data sources are similar but the specific data that can be provided and the format of it differs from the Munich test site. This is a preliminary picture; the precise implementation will be decided on later.

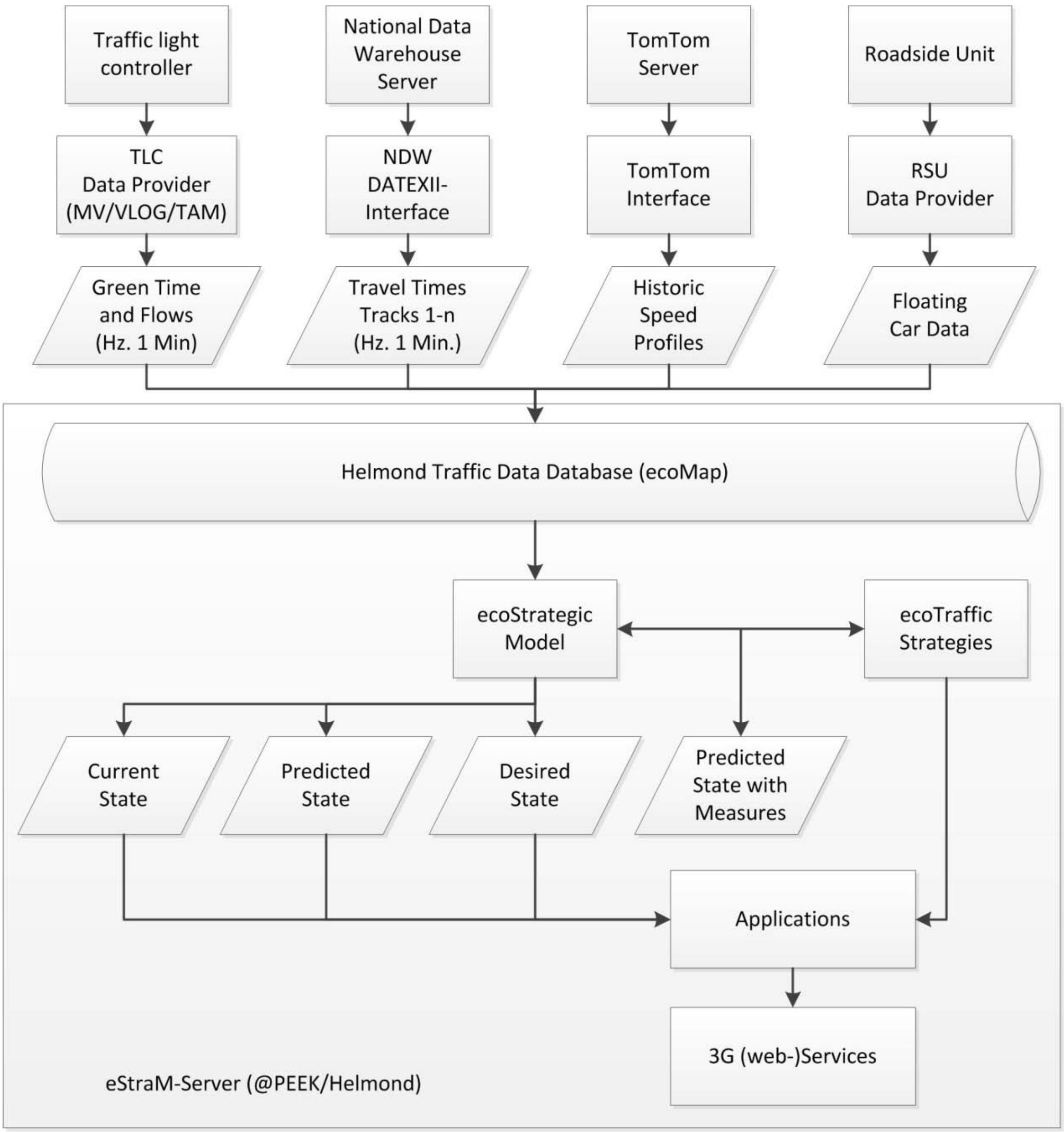

Figure 9: Draft system architecture test site Helmond

Figure 10 gives a further illustration of the architecture of the Helmond test site.

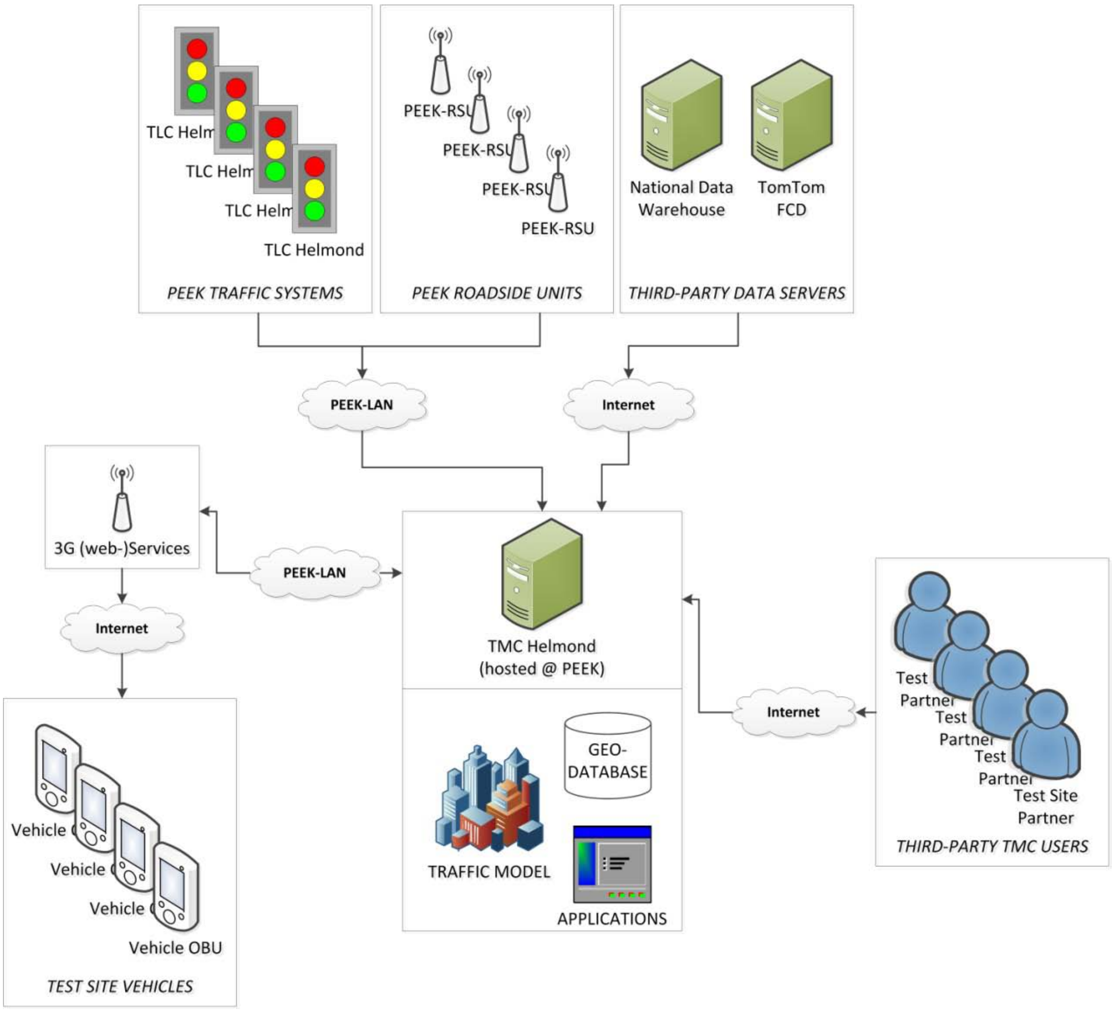

Figure 10: Further illustration of the Helmond test site architecture

As with the Munich test site, the applications development is still in progress. The decision about which views will be implemented and what data will be provided to the applications active in Helmond (most likely via the ecoMap) has not been made yet.

13. Integration, verification, validation of eStraM

13.1. Integration

This document describes the ecoStrategic Model and its relations to other eCoMove core technologies, applications and components. The integration of eStraM will take place at the test sites in the first half of 2012.

In this section, we discuss some software issues that need to be addressed for a successful integration of eStraM in the eCoMove system.

The ecoStrategic Model will consist of an OSGi bundle of software dependencies and packages. In the current plans, there are three design demands to the software of eStraM to integrate eStraM in the ecoMove system:

(1) It should deliver a current traffic state, predicted traffic state and a desired traffic state for each refreshment cycle
(2) It should run on a test site
(3) It should run online

These design issues will be further explained in this section

13.1.1. Delivering of traffic state views

As described earlier in this document, eStraM is going to deliver different traffic state views. These views are completely separate and therefore use different input and generate different outputs. These different views are computed by using different sets of eStraM modules. The following design issues are important for the view generation:

- The input of eStraM consists of multiple data sources. The software design needs to take into account that the right data from the ecoMap needs to be parsed in the right format for eStraM to generate the right view. Selection, copying and moving data is an extensive process in eStraM.
- The software design of eStraM should be able to handle multiple traffic state views at the same time, keeping their corresponding input/output ready for processing. Time stamping the data for these views is very important as well.
- Some of the traffic state views depend on the output of the ecoEmission Estimation Component. eStraM needs to communicate with this component, probably through the ecoMap, and feed it the correct input data.

13.1.2. Implementation on test sites

In the current plan, eStraM will be implemented on a test site. This means, that eStraM will be taken out of its safe lab environment and put in the real world. This real world is continuously changing and sends input and asks for output. The following design issues are important for the implementation on the test site

- The input/output to/for eStraM could be present on time, too late, contain errors, contain missing data or could not become available for some time slices at all. This means, that eStraM needs to check for data availability and correctness. Furthermore, running the model should be timed exactly with the data availability

- Running eStraM on a test site means that there is a need for a computer on site. This machine has, like every other machine, limited working memory, processing power and disk storage. These machine dependencies should be taken into account in the software design of eStraM. There is currently a discussion whether there will be a physical machine at the test site on which eStraM and the ecoMap will run. PTV proposes a more 'distributed' system in which eStraM could run locally with a local ecoMap and synchronise its data with the test site server in Munich which is maintained by PTV. For the Helmond test site, there will be a physical machine running at the TMC from which eStraM could be run as well.

13.1.3. On-line and real time

In the current plan, eStraM needs to run on-line in order to support the SP5 applications. The following software design issues regarding online and real time issues are important for eStraM:

- Some or all of the input data could be delayed so that an update of the traffic state view might be infeasible to be ready for the next update cycle. eStraM should be able to control these situations. Furthermore, running the model could, in some cases, exceed the deadline for an update cycle. eStraM should be able to control these situations as well.
- Running eStraM online means that it is likely that we will encounter some unexpected practical problems. A real test could introduce exceptions which should not introduce a system crash. Therefore, the eStraM implementation needs to be robust, so that it can handle errors and failures gracefully.

13.1.4. eStraM Sofware Design

In this initial phase of the eStraM implementation, it is planned to use the Model View Control (MVC) design pattern, in which the controller, the model and the view are separated to provide a loose coupling between the different aspects of the application (see Figure 11). The next step is to elaborate the MVC design pattern and to make the design choices needed.

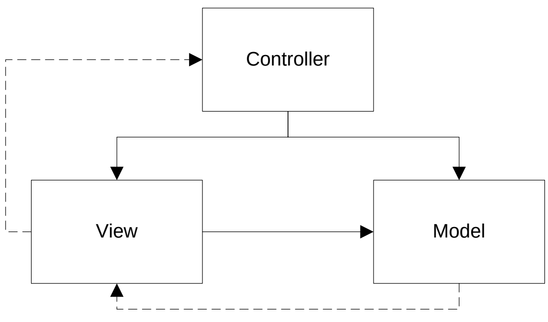

Figure 11: eStraM software design

13.2. Verification

There are two parts of verification (see [M2.3]):

- Verify if the developed component/systems/applications meet requirements identified in the use cases and requirements.

- Verify if the developed component/systems/applications are able to provide all functionalities specified in the architecture and specifications.

For eStraM, this can be translated into the following verification parts:

- Verify if eStraM provides the required (current, predicted) traffic state views correctly.
- Verify if eStraM provides the output data correctly to the ecoMap or to any other parts of the eCoMove system as specified.

As the implementation progresses, more specific plans for verification can be made, in accordance with the overall eCoMove verification approach.

13.3. Validation

There are currently no plans for validation of core technologies, However, there are several ways in which the eStrategic Model can be validated.

Using real-time data, we can compare the generated traffic states (current, predicted) against traffic states measured on the road.

Another possibility is to use data from the simulation environment. VISSIM data can be used to emulate data from road side sensors as well as from vehicles. This makes it possible to validate eStraM (mainly the O-D estimation and prediction) under conditions that are more favourable than can be achieved on the test sites, e.g. with high penetration rates of vehicles communicating floating vehicle data, and sensors in places that offer the most information for the O-D estimation.

Plans for validation of eStraM need to be elaborated further and checked for feasibility in the next few months.