Cooperative Mobility Systems and Services for Energy Efficiency

D550.58(D5.8)	**Setup, manual and experiences with the traffic management and control applications**

SubProject No.	**SP5**	**SubProject Title**	**ecoTraffic Management and Control**
Workpackage No.	**WP5**	**Workpackage Title**	**Integration and verification**
Task No.	**T5.5.1**	**Task Title**	**Integrate the applications for specific test sites**
Authors		C. Bresser, W.P. van den Haak, R. van Katwijk, J. Lüssmann, M. Mann, B. Netten, P. Matthias, C. Sanchez, J. Vreeswijk, F. van Waes, I. Wilmink	
Dissemination level PU/PP/RE/CO		PU	
File Name		D550.58(D5.8) Installation Instructions-CLEAN	
Due date		31.07.2013	
Delivery date		19.03.2014	

European Commission Information Society and Media	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

Control sheet

Version history			
Version	Date	Main author	Summary of changes
0.1	22-08-2012	J. Vreeswijk	Template
0.2	10-09-2012	F. Van Waes	Review of template
0.4	13-12-2012	J. Vreeswijk	Update of template
0.5	27-02-2013	F. Van Waes	Contributions partners added
0.6	03-04-2013	F. Van Waes	Last contributions partners added
0.7	26-06-2013	F. Van Waes	Remarks from reviewer processed
1.0	28-06-2013	J. Vreeswijk	Final version
1.1	13-11-2013	F. Van Waes	Remarks from Y3 Review processed
1.2	28-02-2014	F. Van Waes	Remarks from PreFinal Review processed
	Name		Date
Prepared	F. van Waes		03-04 & 26-06-2013
Reviewed	D. Kuck		29-04-2013
Authorized	Jean-Charles Pandazis		19.03.2014
Verified	Manuela Flachi		19.03.2014

Circulation	
Recipient	Date of submission
Project partners	04.03.2014
European Commission	19.03.2014

Table of Contents

FIGURES

TABLES

TERMS AND ABBREVIATIONS

Abbreviation	Definition
API	Application Programming Interface
CACC	Cooperative Adaptive Cruise Control
CAM	Cooperative Awareness Message
CIS	Central ITS Station
CO_2	Carbon dioxide
DLL	Dynamic Link Libraries
ecoABC	ecoAdaptive Balancing and Control
ecoATS	ecoAdaptive Traveller Support
ecoMM	ecoMotorway Management
eStraM	ecoStrategic Model
ETIP	External Traffic Information Processor
FCD	Floating Car Data
FVD	Floating Vehicle Data
GUI	Graphical User Interface
LDM	Local Dynamic Map
NOx	Nitro oxide
OBU	On-Board Unit
O-D	Origin-Destination
OSGi	Open Service Gateway initiative framework
PKI	Parking Information
REQ	Requirement
RIS	Roadside ITS Station
RMR	Roads and Multimodal Routes
RSU	Roadside Unit
SLAM	Speed and Lane Advice Message
SP	Sub-project
TEC	Traffic Event Compact
TFP	Traffic Flow Prediction
TLC	Traffic Light Control
TMC	Traffic Management Centre
TMDB	Traffic Information Database
TP	Truck Priority
TPA	Truck Parking Area
TPEG	Transport Protocol Expert Group
UC	Use Case
VIS	Vehicle ITS Station
VSL	Virtual Stop Line
XML	Extensible Markup Language

1 Introduction

This deliverable deals with a detailed overview of simulation and test sites activities of the ecoTraffic Management and Control sub-project undertaken, as well as lessons learned from these activities.

1.1. Project overview

The goal of the eCoMove project is 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 %.

The description in this document especially focus on applications and components related to traffic management and traffic control. In the context of ecoTraffic Management and Control in relation with the eCoMove system, this document often refers to ‘infrastructure’. Infrastructure refers to the combination of traffic management and control measures independent of their roadside or central physical location. Roadside infrastructure related to roadside units connected to legacy traffic systems such as traffic lights, ramp metering installations, variable message signs, toll gates and others. On the other hand, ‘central’ generally refers to a traffic management centre or elements which are physically stored at or originated from a back-office location. In total 9 applications and 4 components are described in this document.

1.2. Background

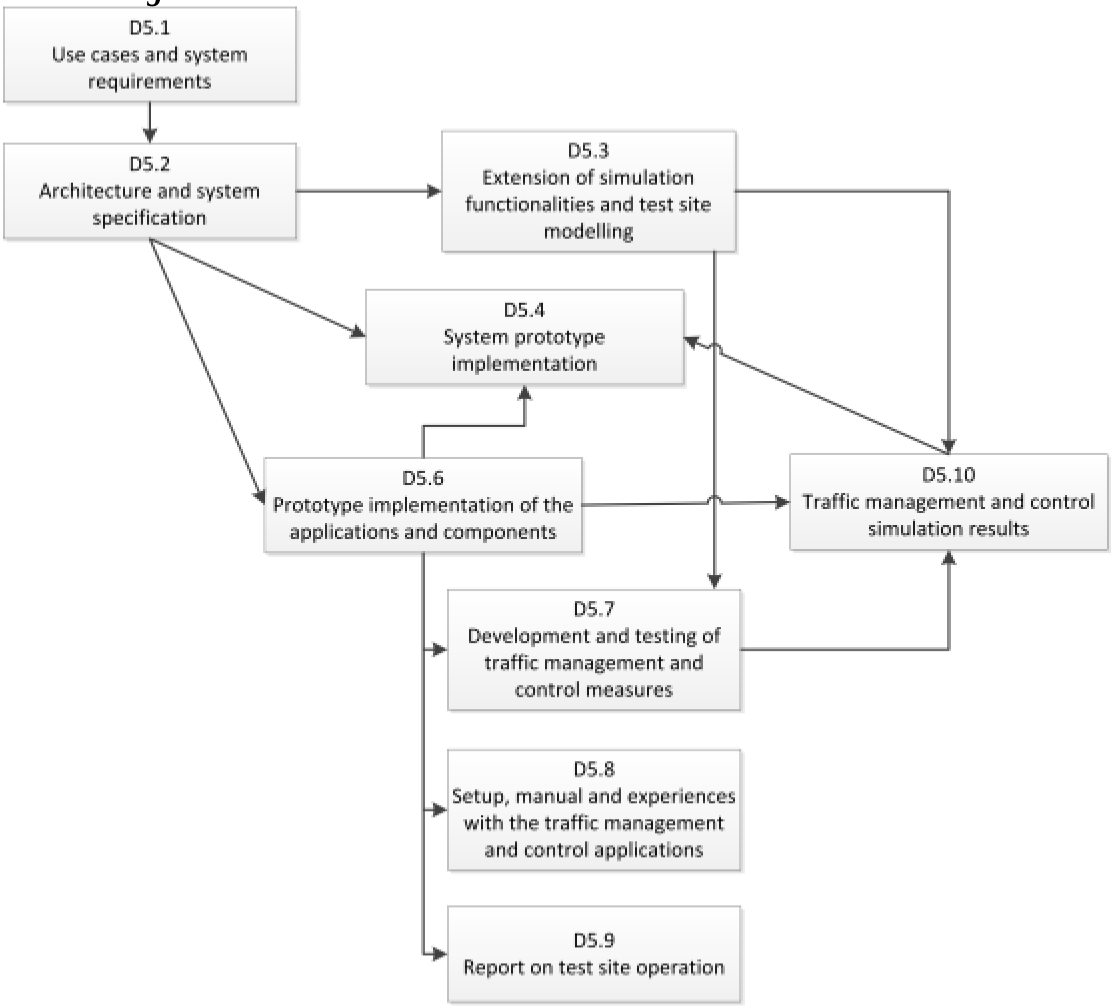

Figure 1 – Deliverables flow diagram

System concepts are described in D5.1 and further detailed in D5.2 towards system designs. The setup of the simulation environment and required extension are discussed in D5.3, while the actual implementation is described in D5.7. Following D5.2, there are two deliverables describing the prototype implementations. D5.6 presents the prototypes of individual applications and components and also includes verification plans. Results from laboratory verification tests are presented in D5.7, while results from field verification are discussed in D5.9. General information about the setup of applications, operational instructions and practical experiences are reported in D5.8. Validation results are discussed in two deliverables. D5.9 includes the results from field validation, while D5.10 is dedicated to impact figures coming from simulation activities. Finally, D5.4 addresses the integration of applications, both in terms of technology as in terms of synergies where impact is concerned.

1.3. Document Overview

1.1.1 *Intended Audience*

This document has been prepared to give the reader some understanding of how the applications and components have been simulated and how they have been tested. The content of this document may be of interest to a wide audience: road operators, traffic engineers, vehicle manufacturers, fleet and logistics, etc. The main purpose however, is to enable developers to oversee the integration steps resulting from the work described in this document. It concerns integration in both field and simulation environments, with infrastructure legacy systems and vehicle systems and in traffic management and control systems. Practical experiences are also discussed.

1.1.2 *Document Structure*

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

Chapter 2 focuses on simulation activities. How were the applications setup for simulation, what preparations were made, how could someone else reproduce the work, and what are the most notable practical experiences?

Chapter 3 focuses on test site activities. How were the applications setup for the test sites, what preparations were made, how could someone else reproduce the work, and what are the most notable practical experiences?

Chapter 4 presents the final conclusions.

2 Integration in simulation

This chapter describes the simulation activities. How were the applications setup for simulation, what preparations were made regarding the user manual and the adaptability, and what are the most notable practical experiences? See the overview in Table 1.

	Simulation	
Application	**Verification**	**Validation**
Format	*Model – Network*	*Model – Network*
ecoRouting – Network routing	-	VISSIM – Munich
ecoRouting – Local level routing	-	VISSIM -Helmond
ecoParking Advice	-	-
ecoGreenwave – German variant	VISSIM - Munich	VISSIM-- Munich
ecoGreenwave – Dutch variant	VISSIM - Helmond	VISSIM - Helmond
Balanced Priority – Variant 1	VISSIM – Helmond VISSIM – Munich	VISSIM – Helmond VISSIM – Munich
Balanced Priority – Variant 2	VISSIM- Helmond	VISSIM – Helmond
ecoRamp-Metering – Green frequeny and speed advice	VISSIM - Badhoevedorp	VISSIM - Badhoevedorp
ecoRamp – Metering – Virtual Stop Lines	AIMSUN	AIMSUN
ecoRamp Metering - Truck Priority	AIMSUN	AIMSUN
ecoSpeed and Headway Management	VISSIM – France?	VISSIM – France?
ecoTruckparking	-	-
ecoTolling	VISSIM - France	VISSIM - France
ecoApproach Advice – Variant 1	VISSIM – Munich	VISSIM- – Munich
ecoApproach Advice – Variant 2	VISSIM - Helmond	VISSIM - Helmond
ecoApproach Advice – Variant 3	VISSIM - Helmond	VISSIM -Helmond
ecoNetwork State and Prediction - Variant 1	VISSIM - Helmond	VISSIM - Helmond
ecoNetwork State and Prediction - Variant 2	VISSIM - Munich VISSIM - Helmond	VISSIM - Munich VISSIM - Helmond

Table 1 –Overview of the applications related to the verification / validation in simulation.

2.1 Application: ecoRoute Advice

Two variants are implemented:

- Network routing
- Local level routing

2.1.1 *Application setup for simulation*

Network routing

The ecoApproach Advice has been developed as a dependent OSGi bundle: "The ecoRouteAdvice bundle" and it includes one java class.

The ecoRouteAdvice bundle requires the bundle eCoMap which is included in the eCoMove simulation environment bundle set.

In order to run the ecoRouteAdvice bundle, the dependencies must be configured in the bundle manifest of the bundle ecoRouteAdvice. Figure 3 indicates the required dependency configurations.

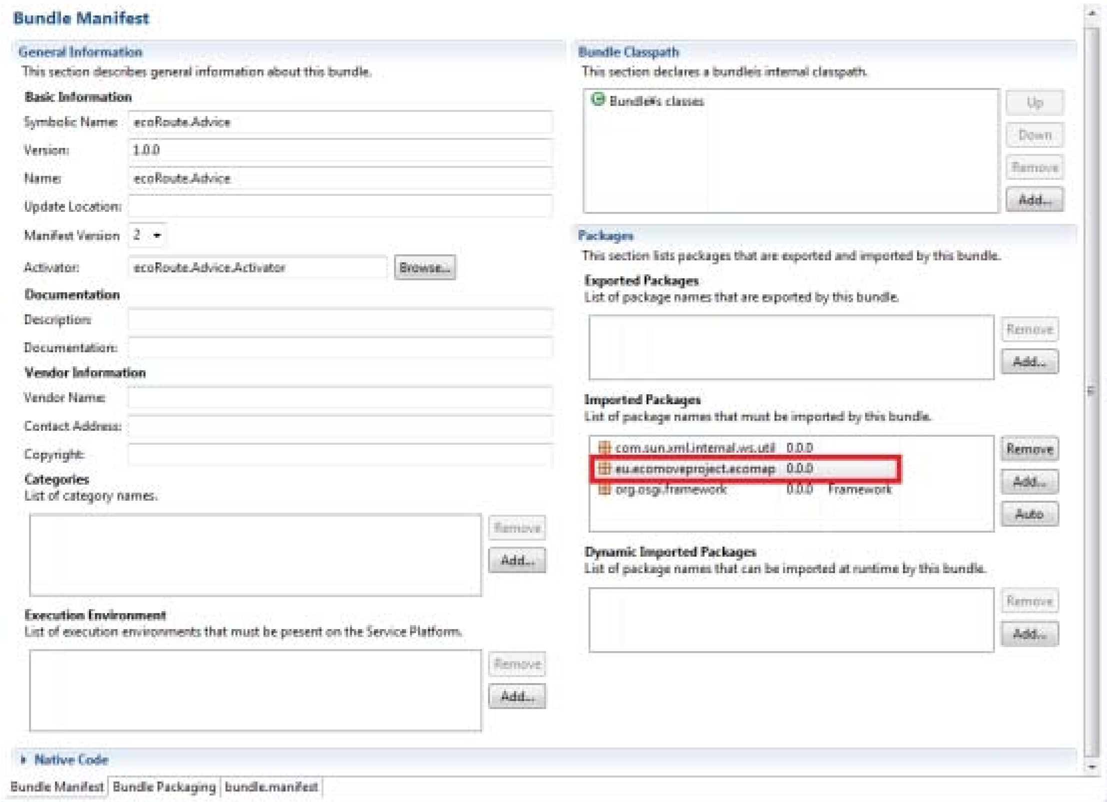

Figure 2 – Configuration of dependency

The ecoRouteAdvice bundle is programmed to work in 3 main steps

1. Strategic Route extraction
2. Acceptable Route extraction
3. Acceptable Route

Since the second step needs travel time consequently it requires dynamic data. Therefore the bundle gets active when the simulation is started by calling the batch-file.

The basic working principles of the bundle are:

- Accessing all routes defined in VISSIM by reading main VISSIM input file (.inp) which is located in [path-to-installation-folder]\Data\[Name of the city e.g. Munich].
- Extracting routes including broken links (Strategic Route extraction) in order to provide first shortlist of routes.
- Then accessing the first shortlist to determine the best route in a route groups having a same starting and the ending point as defined in Figure 2.

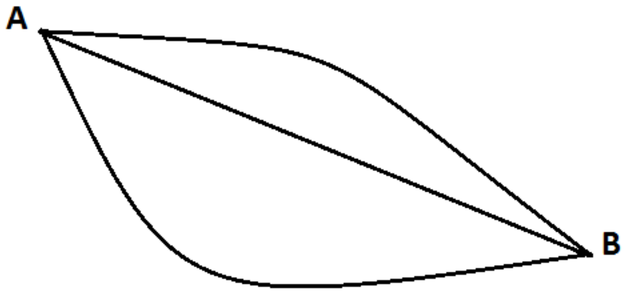

Figure 3 – Routes having the same starting and ending points

- The final step is to compare acceptable routes and desired routes.

Local level routing

A route diversion strategy was implemented for a situation involving a railway crossing in Helmond. Due to a high frequency of trains this railway crossing may be closed for a period up to 10 minutes. Local level routing guides drivers to the next intersection where they can pass the railway by means of a tunnel. The traffic situation was modelled in VISSIM (see Figure below and also section 4.1) and the dynamic route decision module of VISSIM was used to influence route choice. The eCoMove local level routing application monitored the status of the railway crossing (open/closed). If the railway crossing was closed it would overwrite the planned route of vehicle intending to pass the railway crossing. Both local vehicle actuated and network adaptive traffic light control were both used to see which of two performance best with route diversion strategies.

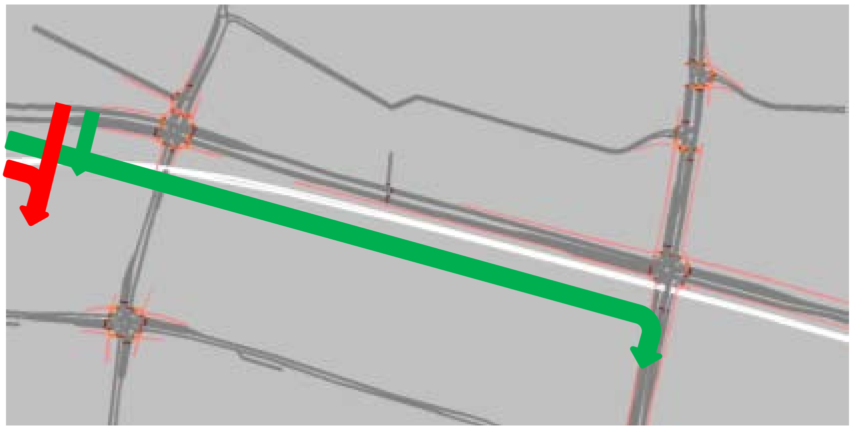

Figure 4 – Scenario local level routing Helmond

2.1.2 *User Manual and Adaptability*

Network routing

The ecoRouteAdvice is initialized by calling the batch file. The simulation network and the Knopflerfish with the integrated bundles open. The application is executed during runtime of the simulation. It is a precondition, that a certain percentage of equipped vehicles (C2X vehicle) are configured, as these are the only vehicles influenced by the application.

The application requires the setting the location of VISSIM input file (.inp) of related network. By default, path to and the name of VISSIM input file is set as "C:\SP5_Simulation_Environment\Data\Munich2012\EcoMove_05-21(withGlobalRoutingDecisions)". However the location and the name completely depend on path to installation folder and the name of VISSIM input file.

In order to run the application, this default path and name configuration must be changed with the actual path and the name of network VISSIM input (.inp) file in your computer. Figure 5 and Figure 6 explains how to change this default value more detailed. The new entry in the java class (Activator.java) located in the bundle ecoRouteAdvice should be "[in [path-to-installation-folder]\Data\[Name of the network folder]\[Name of the VISSIM input file].inp"

e.g.: "C:\SP5_Simulation_Environment\Data\Helmond\Network.inp"

```
public class Activator implements BundleActivator {

  /* (non-Javadoc)
   * @see org.osgi.framework.BundleActivator#start(org.osgi.framework.BundleContext)
   */
  public void start(BundleContext context) throws Exception {

      try {
          BufferedReader input =  new BufferedReader(new FileReader(
                  "C:/SP5_Simulation_Environment/Data/Munich2012/EcoMove_05-21(withGlobalRoutingDecisions)"));
          String line = null;
```

Figure 5 – Location of the line in Activator java Class (ecoRouteAdvice/src/ ecoRoute.Advice/Activator.java)

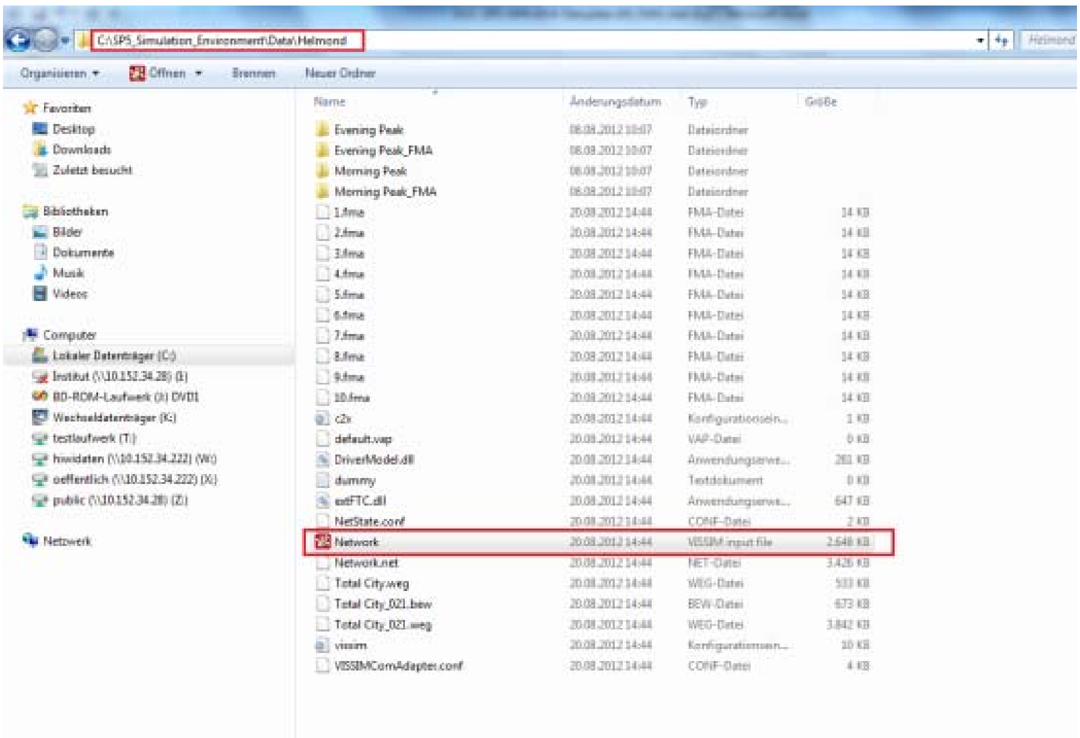

Figure 6 – Location and name of VISSIM input file

For the desired route matrix the ecoNetwork Prediction component is needed. The exchange of data is realised to a text file. The text file has a VISSIM INP data format for routes.

Local level routing

Local level routing in eCoMove uses pre-configured route diversions which are activated by a dynamic event. For each situation the user (e.g. the road operator) needs to determine the triggering event, the route(s) affected by the event and the recommended/desired alternative routes.

2.1.3 *Practical experience*

Network routing

Getting the output of the application may take some time up to 5 minutes depending on the performance of a computer.

Local level routing

Simulation results showed that in terms of CO2 the rerouted vehicles hardly benefit from local level routing. This can be explained by the extra distance they travel with an extra signalized intersection to pass. These vehicles did have significant time savings. There were clear CO2 improvements on a network level which can be best explained by increased and therefore more continuous vehicle throughput when the railway crossing is closed. As a consequence, compared to the baseline vehicle queues decreased and less vehicles had to be processed immediately after the railway closure. Adaptive traffic light control with queue estimation played a crucial role in this process.

2.2 Application: ecoParking Advice

Although the application is not used during simulation it is possible to do so. Below is a theoretical description on how this can be obtained. The actual application was not hooked to the simulation environment because of a lacking routing engine which is necessary to evaluate which is the most 'eco' route and destination for parking using the coordinate systems and network from the simulation environment.

It was not intended to simulate ecoParking Advice, it was a possible addition to the project to be able to address the effect. Unfortunately this addition proved impossible with the available means. The model which promised to be useful was limited in that it couldn't be used with dynamic destinations and the knowledge to change this was no longer available within the consortium.

2.2.1 *Application setup for simulation*

The ecoParking Advice application is provides a freely addressable webservice. This webservice uses the xRoute server from PTV which also provides a webservice.

To hook ecoParking Advice to the simulation environment a configuration for the simulation environment and a simulator for the routing engine must be created.

The configuration of the ecoParking advice consists of a series of geometries for the parking area's using a coordinate system. The simulation configuration must use the same coordinate system as the simulation environment and the simulated parking areas should be located at the correct location.

The routing engine simulator must use the coordinate system of the simulation environment and must contain the network of the simulation environment in order to be able to route to the correct locations.

The ecoParking advice application runs as a webservice within the eCoMove framework. The webservice URL is configurable in the configuration file of the webcontainer. This configuration is described in the manual of the webcontainer.

From the Traveler Support Manager in the simulation environment calls can be made to the webservice that is provided by the ecoParking advice service. The interface description can be found in the design, implementation and verification documentation.

2.2.2 *User Manual and Adaptability*

To get the ecoParking Advice application up and running the following steps should be performed:

1. Install xServer along with a working license using the supplied installation manual. For the simulation environment an xServer simulator should be created that can handle requests from within the simulation environment and the associated coordinate system.

2. Copy the application files of the ecoParking Advice application to the eCoMove simulation environment webcontainer.
3. Start the eCoMove container.

The ecoParking advice configuration contains the following settings which influence the behaviour of the external interface:

1. A KLM file containing the parkings and serving area's (currently called Helmond.KLM). This file is created using Google Earth.
2. An xml file containing logging settings for Log4J. See the Log4J manual for further configuration options.

The URL and port can be configured using the webcontainer configuration.

Next to the adaptations to these settings in the configuration, the application can be extended as well using standard Java build tools as described in the development environment documentation.

2.2.3 *Practical experience*

During development and testing the following usage and development issues were noted:

- xServer is a commercial product for which a license is necessary. There exist Open Source alternatives and free alternatives but a distinct disadvantage of these products is that there is no guarantee that the interface will remain the same. For production systems this is unacceptable so 'a' product should be chosen to provide the green routing algorithm as this is no longer part of the eCoMove toolkit. For test, development or demonstration systems a free and open routing engine is more suitable as this gives more flexibility in test site usage.
- The chosen routing engine is engineered to solve complex multiple destination routing problems that must overcome the traveling salesman paradigm. This means that the used engine is very accurate and very suitable for backoffice applications. The routes requested by ecoParking Advice are much less complex than the routing engine can handle. The routing algorithm therefor can be optimized to provide better performance.
- The ecoParking Advice application provides an easy to use and fast interface over which the 'eco' parking information can be retrieved. This information can be used by navigation manufacturers to provide a list of destinations from which the user can choose.

2.3 Application: ecoGreen Wave

2.3.1 *Application setup for simulation*

German variant

The ecoGreen Wave application was developed as one project in a C++ environment. It consists of 10 classes. Besides theses classes a DLL for emission calculation and VISSIM models (including the fixed time program is needed to run »ecoGreen Wave«.
Connected to the simulation the optimization starts automatically every optimization interval due to the simulation time stamp. Via a text file the ecoGreen Wave gets input about predicted traffic volumes and turning rates. The offset of the traffic light controllers in VISSIM is automatically changed via the external Fixed Time Controllers (»ext.FTC«).

In order to integrate the component ecoGreen Wave into the simulation environment, the following additional developments have been undertaken:
External control of VISSIM signal groups according to given fixed time plans → »ext. FTC«
The component »ext. FTC« has been realised that is via DLL calls connected with the micro-simulation VISSIM. It reads all within VISSIM defined fixed time signal plans and controls the VISSIM signal groups according to these plans remotely. It furthermore can receive new signal plan offsets from any remote partner (»ecoGreen Wave«) via socket, and processes these new offsets by rotating the signal plans. With these mechanisms it is possible to test the dynamic ecoGreen Wave online together with VISSIM.

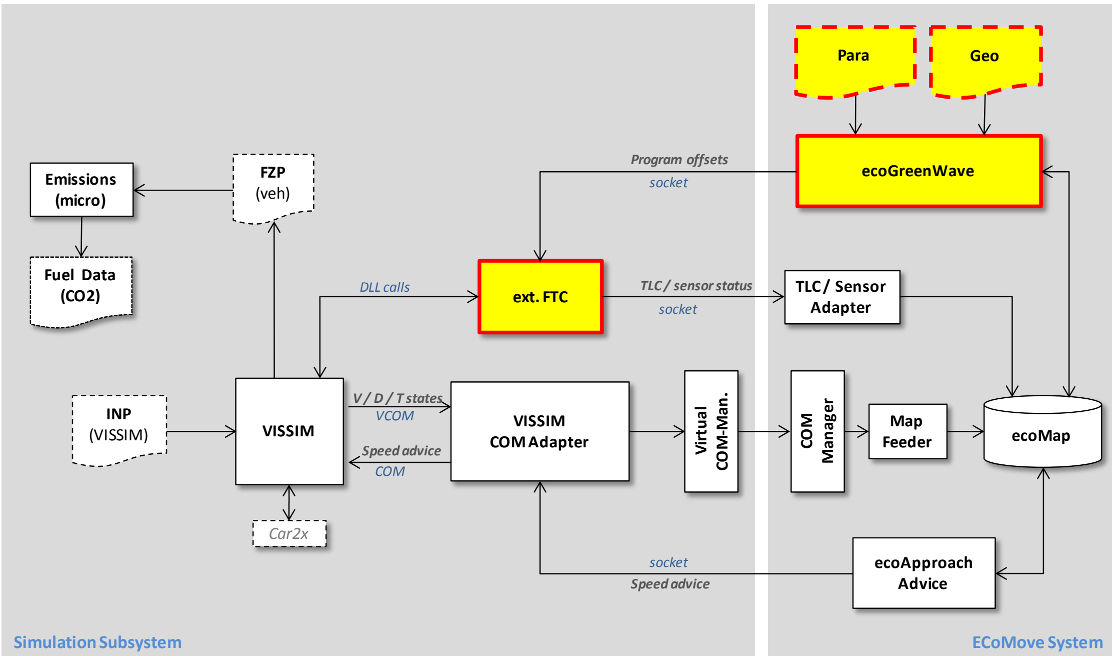

Figure 7 – Integration of application ecoGreen Wave in eCoMove simulation environment

Dutch variant
The Green Wave Application for Helmond is working with ImFlow. Therefore the architecture of a local controller looks like this:

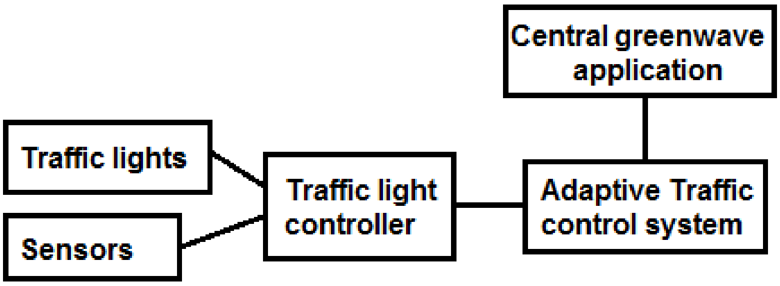

Figure 8 – Helmond Green Wave Architecture

In simulation this means the following for the blocks in the figure:

- Traffic lights are Signal Heads in VISSIM, they are sent to the traffic light controller through the SimInterface dll that couples Peek controllers to VISSIM.
- Sensors are inductive loops in real and modelled as such in VISSIM as well. Their information is sent through the same SimInterface to the traffic light controller.
- The traffic light controller is basically the same one as on the street, just a windows-executable version of it that is compatible with the SimInterface.
- The adaptive control system works the same as on the street as well, it takes over control of the timing and planning of the signal phases and its configuration can dynamically be adapted.
- The last block, the central Green Wave application is ecoGreenWave. It changes the configuration while the system is running.

Inside the application coordination and timing (length) for a centrally controlled green wave is calculated. This is injected in the adaptive control system following the architecture depicted below:

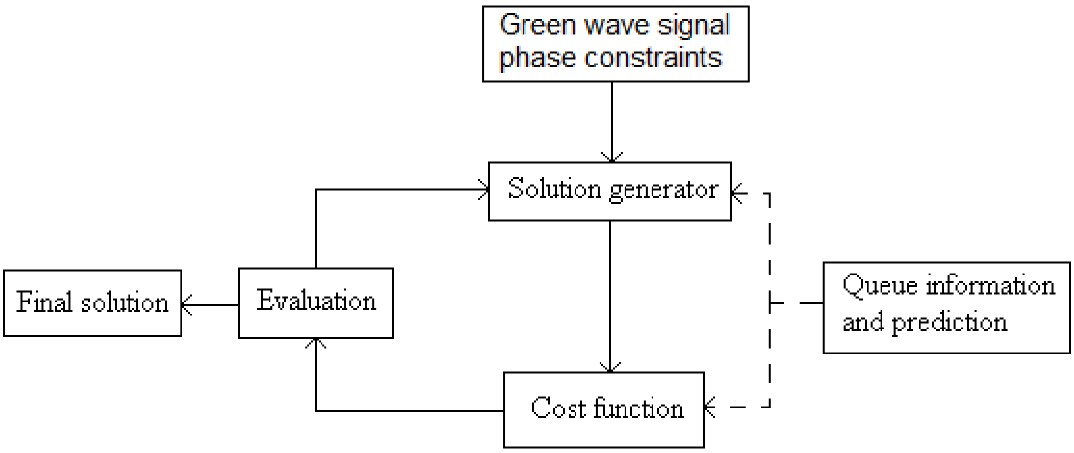

Figure 9 – Coordination architecture

The idea is that the adaptive signal controller gives away control over the main directions signal groups and the central green wave controls them. That way outside the green wave, there is still full flexibility for the controller, while the green waves are guaranteed. However, because there is no direct interface to inject green wave constraints into ImFlow yet, these constraints were modelled by using priority requests, while normal sensors from the main direction were removed in order not to get green on those directions outside the green wave. These requests are sent directly to the controller who sends them to ImFlow in the same way it would send a real bus priority request.

2.3.2 *User Manual and Adaptability*

German variant

ecoGreen Wave needs four different inputs:

- Configuration File
- Road File
- Demand File
- and VISSIM input files

In the Configuration File the following parameters can be set:

- Path of Road and Demand File
- IP address of corresponding VISSIM simulation
- Port and IP address of the ecoApproach Advice for data exchange
- Optimisation interval in seconds
- Number of progressive speed variants
- Maximal deviation from default speed in kilometres per hour
- Object Function: 1 – Platoon model, 2 – Simulation model (optimization on CO_2 emissions)
- Optimisation Algorithm: 0 – Iteration

In the road file the following parameters are configured. The Road File is split up into three sections. The configuration has to be done mostly according to the VISSIM INP files.

Intersection configuration (has to be configured for each intersection):

- Intersection Index
- Intersection name
- Signal program Index
- Cycle time
- Signal program offset

GreenWave configuration (has to be configured for each section to be optimised – a section consists of two neighbouring intersections):

- GreenWave Index
- GreenWave Path (Path of the VISSIM INP file)
- ID of VISSIM Input Links (beginning left, anti-clockwise), -1 if Input Link is not existing

GreenWaveNodes configuration (has to be configured for each node and driving direction; the order of the notes is relevant first one direction, then opposite direction):

- Intersection Index
- Green Wave Index
- Path to the corresponding VISSIM fixed time signal plan configuration file
- Direction: 1 – direction, -1 – opposite direction
- VISSIM approach links - link IDs of to the stop line
- VISSIM exit link – link ID of link after the stop line
- Signal group index (for signal groups belonging to last the approaching link)
- Begin and end of green time (for signal groups belonging to last the approaching link)
- Distance from last intersection
- Minimal number of lanes on approach
- Speed limit on approach

The Demand File includes information about predicted traffic volumes and turning rates for each section of the information. This information can come from the ecoNetwork Prediction. In simulation this information is part of the INP file and therefore known.

For each section a VISSIM INP file has to be created. Figure 3 shows a section including the traffic light configuration. The net model only contains the most important part of the network.

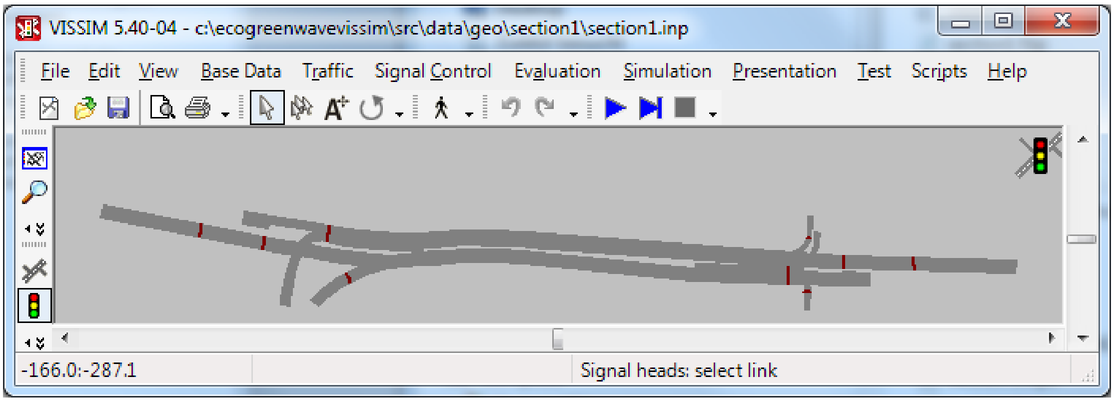

Figure 10 – Section of the Optimisation

The following elements have to be configured:

- Links and connectors (including lane information)
- Routes
- Signal heads
- Fixed time control
- Pre-signal for the main direction to model the platooning effect of the ecoApproach Advice
- Desired speed decision points (including desired speed distributions according to the Configuration File)
- Vehicle composition according to simulation environment
- Simulation settings
- Evaluation settings

Dutch variant

Running the application is quite straightforward. Just start up the code when a simulation has just been started. VISSIM can be paused right after starting up so that there is enough time to start up the java application for green wave. VISSIM should also be instructed not to collect evaluation data for the first 5 minutes, this is in order to fill the network with traffic and let all applications and systems initialize properly.

The application is solely made for the Helmond situation and no effort has been made to really productize the application for commercial exploitation. Therefore, all configurations are hard-coded in the application and it cannot directly be implemented in another network. A detailed manual is not really applicable either then, the code is well documented using javadoc and comments, so that changes of configuration are easy to make.

2.3.3 *Practical experience*

German variant

A detailed calibration of the vehicle driver models within the simulation will be performed after the driver behaviour assessment.
To run ecoGreen Wave it is necessary that all desired speed distributions included in the optimization process also exist within the INP files. VISSIM must be run in English. VISSIM 5.40-x is needed due to evaluation options needed.

Dutch variant

In practice it turned out that the priority requests were not always satisfied in exactly the same amount of time. Investigation is still going on what is causing this, but a back-up scenario has been made that implements the results of the green wave algorithm as a semi-fixed time program in ImFlow. The advantage of semi-fixed time is that the stages not in the green wave can be configured in such a way that they can be extended or shortened according to the demand of the traffic dynamically. The disadvantages are that the order of stages is fixed and that the green wave timing cannot be adapted dynamically according to the traffic demand.

2.4 Application: ecoBalanced Priority

As ecoBalanced Priority builds upon existing traffic light control algorithms there are various ways to implement this application. Two variants are distinguished in this section: one developed by a traffic systems vendor and one developed by a research institute.

2.4.1 *Application setup for simulation*

Variant 1

For simulation ecoBalanced Priority uses the following architecture:

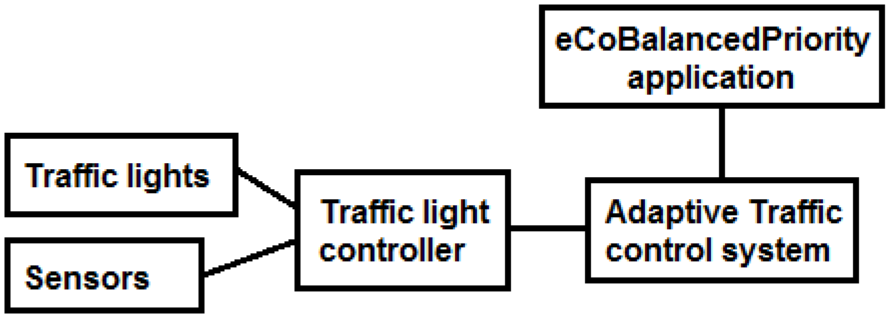

Figure 11 – ecoBalanced Priority Architecture

Again just like in the green wave application the traffic lights and sensors are connected to the traffic light controller through the SimInterface.dll. Additionally, priority vehicles from VISSIM now also send their PT-telegrams through this same interface. The adaptive traffic control system, ImFlow, works just like on the street, but gets configuration parameters dynamically through the connection with the ecoBalanced Priority application. These are determined according to the weight of the vehicle, emission class, time of the day and can also be influenced by strategies coming from the ecoTraffic Strategies application. However, the latter is not implemented, but technologically possible and not hard to integrate.

In simulation the vehicles do not send information about their emission class or weight, in the field this information should be in the ecoCAM message. Therefore, variation was simulated by using different line numbers in the PT telegram that indicate certain virtual properties about the vehicle.

Variant 2

The ecoBalancedPriority and ecoApproachAdvice applications are both programmed using the JAVA programming language and are connected to VISSIM using a combination of application programming interfaces or API's. The COM-interface and the Signal Control API provided by VISSIM and the JNI-API provided by the JAVA programming language are used to link the two components to VISSIM.

Communication with the signal controllers in VISSIM can be done in two ways: either through a library that uses shared memory for communication or via a library that uses a network socket for communication or substituting one dynamic link library (DLL) for the other switches between the two modes of communication without having to adapt any of the initialization or input-files. The DLL is named SignalDLL.dll in both cases and should be placed in the directory where the VISSIM input-files also reside. Using the VISSIM user interface the (to be) controlled intersections should be selected to be of type "external". For all of the externally controlled intersections the files "SignalDLL.dll" and "SignalGUI.dll" (essentially a stub-dll as the application does not offer any dialogs) should respectively be selected as "Program file" and "Dialog DLL file".

The library that uses a network socket for communication employs a protocol that is adapted from the VLOG-protocol (a Dutch communication standard for real-time communication with a traffic signal installation). As the network and the use of this protocol introduce a slight overhead this interface is not used for the batch simulation runs. The name of the first parameter file (Data file 1) is interpreted either as the name of the file to be used for shared memory communication or as the name of the socket

Figure 12 shows the configuration settings for an intersection.

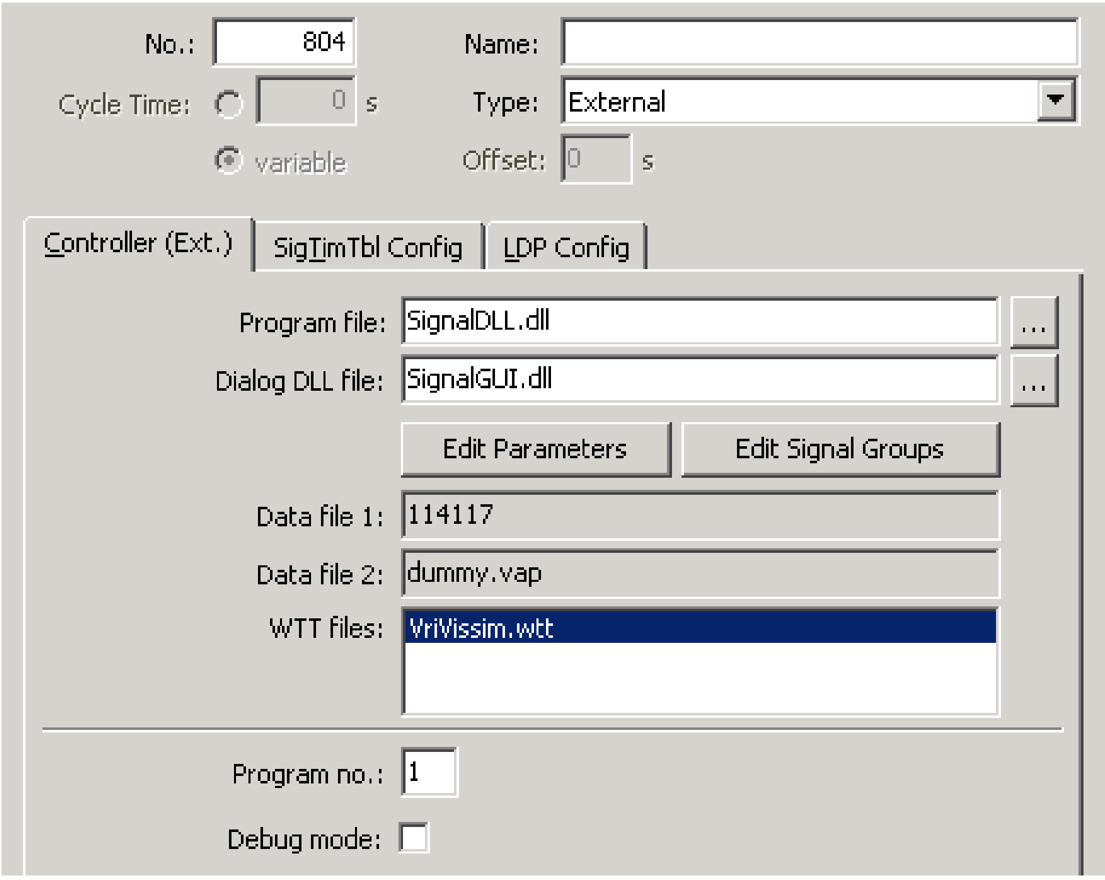

Figure 12 – Configuration settings for an intersection.

2.4.2 *User Manual and Adaptability*

Variant 1

The application is solely made for the Helmond situation and no effort has been made to really productize the application for commercial exploitation. Therefore, all configurations are hard-coded in the application and it cannot directly be implemented in another network. A detailed manual is not really applicable either then, the code is well documented using javadoc and comments, so that changes of configuration are easy to make. Special attention has to be given to the line numbers and what kind of vehicle a certain line number is simulating.

Variant 2

When optimizing the performance of a controlled intersection often only the stopped delay is taking into account. One of the reasons for this is that in order to finish the optimization in time some sacrifices are made with respect to the fidelity of the model. Typically either a vertical or horizontal queuing model is employed in which traffic assumes one of two states: free flow or idling. For eCoMove two additional models were developed. The first model takes into account additional traffic states, amongst which the traffic states intermediate to idling and free flow (i.e. decelerating, accelerating) , but also the fact that it is inevitable in some cases that vehicles are stopped twice. The second model builds upon the first model and associates the

different delay components with different levels of CO2 production (i.e. an idling vehicle emits significantly less CO per second than an accelerating vehicle does). This model furthermore takes into account the different emission characteristics of vehicles as a truck emits significantly more CO2 than a person car does.
In order to be able to choose between the different models the `<performance>value</performance>` element must be added to the `<Junction></Junction>` element of the xml-file used for initialization, otherwise the default model will be employed in the optimization module. Allowed values are 0 (= default model), 1 (high-fidelity delay model), and 2 (CO2 performance model)

2.4.3 *Practical experience*

Variant 1
The application worked well in simulation. Trucks get their priority and can be tracked nicely on the ImFlow web interface (see D5.7) to verify that the application is working well.

Variant 2
Simulations for ecoBalanced Priority were combined with those for ecoApproach Advice. Practical experience has shown that is difficult to give a static advice to vehicles approaching an adaptive intersection as this severely limits the flexibility of the adaptive intersection. Hence, there exists a trade-off between flexibility and predictability traffic lights.

2.5 Application: ecoRamp Metering

2.5.1 *Application setup for simulation*

Green frequency and speed advice

The application will be evaluated on the network of the Badhoevedorp test site, the A9 near Badhoevedorp. The network is modelled in VISSIM.
The ecoRamp Metering Application has been developed as one OSGi bundle. A second bundle has been built to link the VISSIM simulation environment.

Virtual stop lines

The virtual stop line (VLS) strategy has been implemented using the control plan functionality of Aimsun. Two control plans were used. One control plan to regulate the ramp metering installation based on the ALINEA algorithm and a second control plan to control a virtual stop line 50 meters upstream of the ramp metering installation.

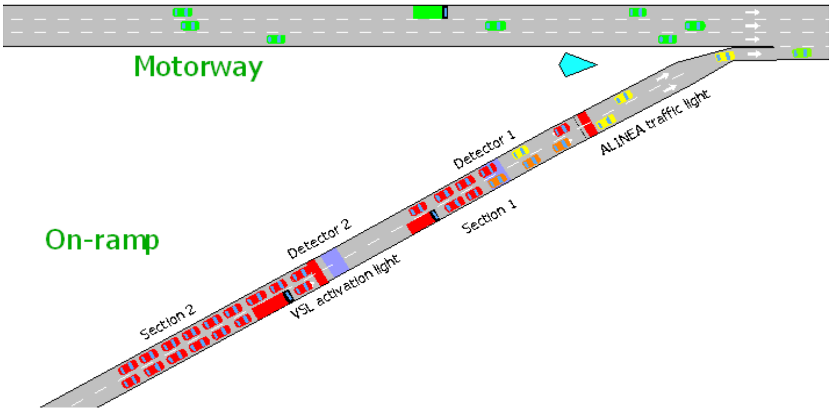

Figure 13 – Overview of on-ramp and VSL strategy activated

The on-ramp is divided into two sections. When the VSL strategy is active, vehicles in section one will anticipate to the ramp metering installation based on the ALINEA algorithm as they normally would, while the vehicles in section two remain idle until the VSL strategy is deactivated. Also see Figure 13. The activation and deactivation process of the VSL strategy is summarized in Figure 14.

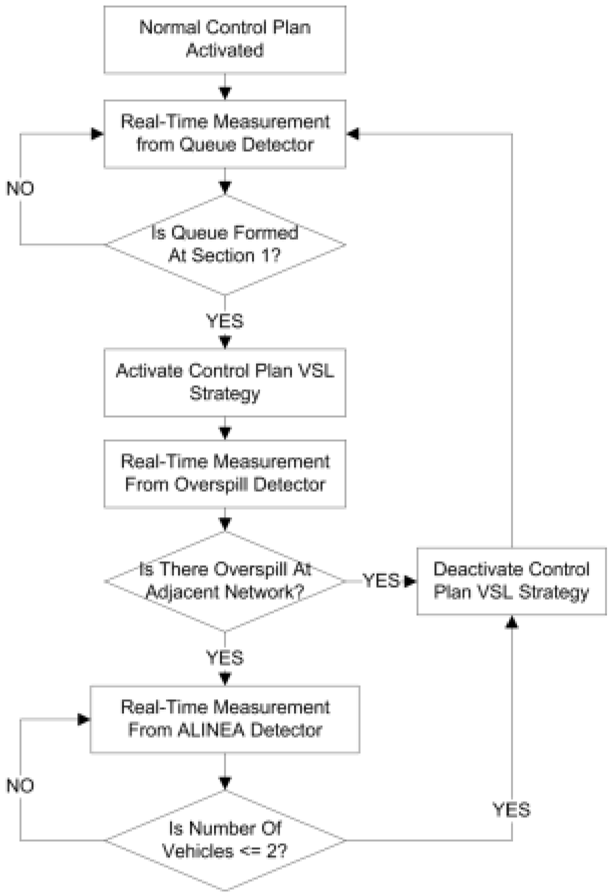

Figure 14 – Flowchart of virtual stop line strategy

Truck priority

The Truck Priority (TP) strategy has been implemented using the control plan functionality of Aimsun. Similar to the VSL strategy, the first control plan regulates on-ramp flow based on the ALINEA algorithm, i.e. without truck priority. Under the condition that a truck is present on the onramp, control plan two is activated. The flow chart of the TP strategy is shown in Figure 15.

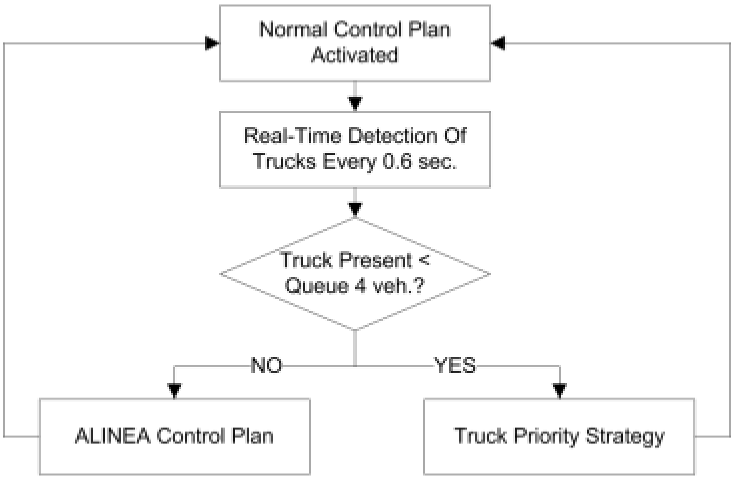

Figure 15 – Flowchart of truck priority strategy

2.5.2 *User Manual and Adaptability*

Green frequency and speed advice

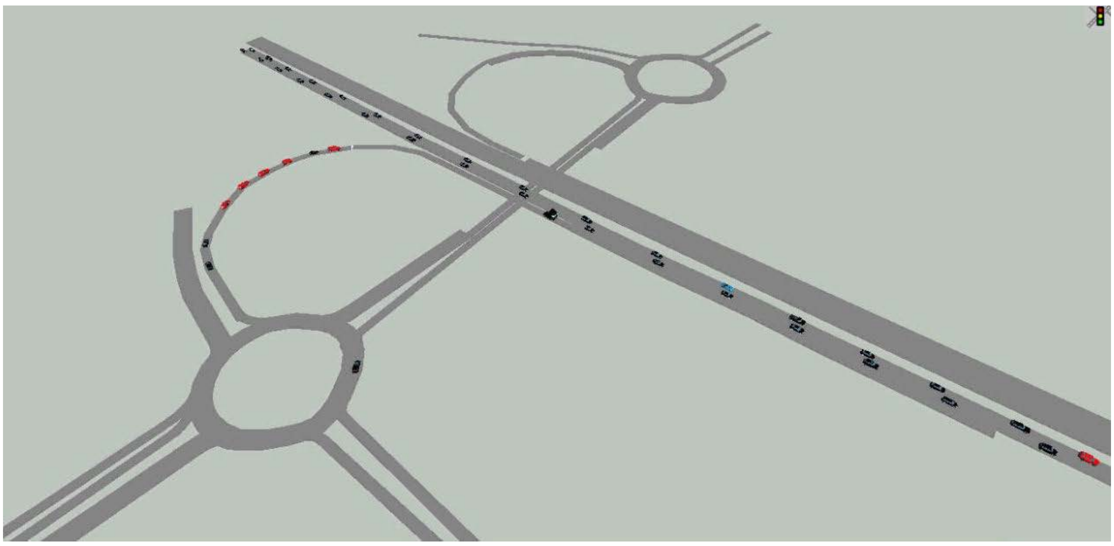

Figure 16. The VISSIM-model with the on ramp Badhoevedorp

The instructions for starting the simulation are:

1. Start VISSIM in the Quicklaunch
2. Start TestTerm in the Quicklaunch. Accept OK in the window popping up
3. Start Cygin Bash Shell in the Quicklaunch. A box appears.

- Changes the prompt from U:\ into C:\.
- Type: cd eCoMove_dev/runtime_env/osgi.
- Type: tail –f ecorampmetering.log

4. Start_osgi.bat in directory 2. OSGi desktop is popping up
5. Activate the ecoRamp Metering Bundle
6. Activate the Vissim connector Bundle

Virtual stop lines

The two control plans and AIMSUN are activated and deactivated based on measurements from detectors and sections which are evaluated by triggers. In AIMSUN, a trigger is a true or false expression based on a predefined criterion. Exploratory simulations teach that on-ramp traffic enters stop-and-go behaviour when the on-ramp occupancy exceeds 80%. Hence, 80% occupancy was selected as an indicator for stop-and-go traffic. The VSL strategy control plan is activated when three conditions are met:

1. The occupancy at detector 1 is greater than 80%.
2. The queue formed at section 1 is greater than or equal to 10.
3. The occupancy at detector 2 is less than 80%.

The first and second conditions are used to check if there is a queue formed at the on-ramp with more than 10 vehicles. The third condition checks if there is an overspill to the adjacent network. In short, the VSL strategy is only activated when there is no overspill to the adjacent network and there is a queue greater than or equal to 10 vehicles. The VSL strategy is deactivated again if one of two conditions is met:

1. The occupancy at detector 2 is less than 80%.
2. The queue formed at section 1 is less than or equal to 2.

Both conditions check if the vehicles in section 1 have been cleared to the motorway. If the section is nearly empty, the normal control plan is activated until the conditions for the VSL strategy are met again.

Truck priority

Similar to the VSL strategy, triggers are used for evaluation, this time using the occupancy measurements of the detectors on the on-ramp. Truck priority is only realized if the truck is the fourth vehicle in the queue. Preliminary simulations showed that giving priority to trucks at a bigger distance from the stop line disrupts the flow on the motorway disproportionally. In summary, the TP strategy control plan is activated when:

1. A truck is present on the on-ramp.
2. The truck is among the first 4 vehicles in the queue.

2.5.3 *Experience from simulation*

Green frequency and speed advice

Simulations were executed. The first results give the impression that the distance of 200 meter between entering the ramp and the stop line is too short for speed adaptation without stopping.

Virtual stop lines

Simulation results showed that the VSL strategy reduces fuel consumption on metered motorway on-ramps significantly. This can be explained by the reduction of stop-and-go behaviour while queuing. It is important to note that the VSL strategy is experimental to study what energy savings could ultimately be realized. Considering driver compliance the VSL strategy may not be very realistic, in particular not when the penetration rate of cooperative vehicles is low. Nonetheless, the basic principles have proven their potential and could possibly also be applied in over-saturated traffic conditions at signalized intersections.

Truck priority

Simulation results showed that the Truck Priority strategy can significantly reduce fuel consumption on metered motorway on-ramps without major negative side-effects on the mainstream flow. The basic priority principles could be applied in different ways for single and dual-lane on-ramps as well as single and dual-branch on-ramps. This simulation study examined a single-lane on-ramp and decreasing green intervals in combination with an increasing number of vehicles per green. When a dual-lane on-ramp is concerned the lane with the particular vehicle can be favoured over the other, releasing more vehicles till the heavy vehicle has passed. Such a way of lane discrimination could also be applied at dual-branch on-ramps where two traffic flows come together. Based on the composition of the traffic flow (e.g. share of heavy vehicle) or upstream influences of the traffic flows (e.g. spillback) they may be treated differently.

2.6 Application: ecoSpeed and Headway management

The application is evaluated on the network of the French test site, the A10 near Tours. The network is modelled in VISSIM

2.6.1 *Application setup for simulation*

The application is tested in the MOBYSIM simulation environment [ref ITSC2011, PreDrive], which consists of several simulators (see Figure 17). VISSIM simulates traffic on the A10 near Tours. ENVIVER is used for off-line evaluation of the VISSIM data (dotted arrow). The Road Side Sensors and System Simulator (RS4) simulate the Road side ITS Stations (RIS) running the eCoSpeed & Headway Management application. MOBYSTAT is communication simulator and simulates the delivery of the messages between the RIS and Vehicle ITS Stations (VIS). The CLOCK controls the timing of the simulation runs and controls the synchronization of the real time simulators with VISSIM time steps.
The MOBYSIM middleware (light blue channel) provides all communication between the simulators, including the simulator control (thin blue arrows), the G5 communication (thick green arrows) and the road side sensor input to the RIS (thick blue arrow). M2M interfaces VISSIM to the MOBYSIM environment and manages the time synchronization of VISSIM. M2M uses the COM interface to exchange data with vehicles in VISSIM. M2M reads vehicle positions and forwards these to the other simulators. M2M also contains the logic for the Vehicle ITS stations; it encodes, decodes, the G5 messages, sends and receives the messages via MOBYSTAT, interprets the speed and headway advice, and adjusts the speed of the equipped vehicles in VISSIM.

MOBYSTAT is a real time simulator for G5 vehicle ad-hoc networking. It uses a statistical model to predict the delivery of individual messages based on the locations, antenna and transmission power properties of the ITS stations. The positions and properties of VIS and RIS communication units are received from the other simulators. MOBYSTAT receives a byte encoded message from a sender, predicts which neighbouring ITS Stations receive the message and forwards the message to each receiver in the other simulators.

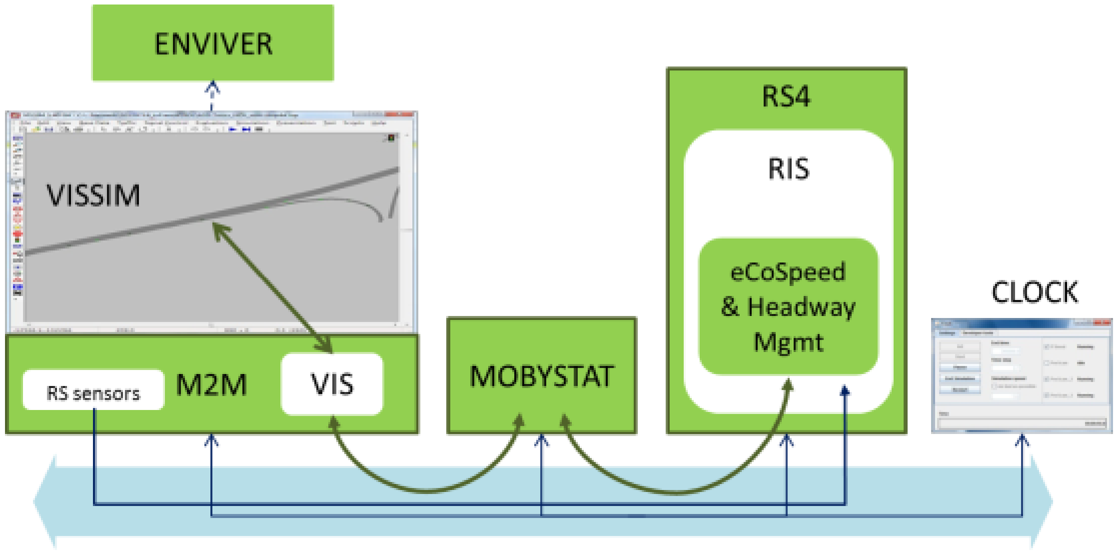

Figure 17 – MOBYSIM simulation environment with VISSIM and the RIS application

RS4 simulates the top layers of a RIS application unit (Figure 17). It runs an application unit with facilities like the message managers, a dynamic map, and interfaces to road side sensors. The message managers encode SLAMs for the speed profiles generated by the application, and decode CAMs from the VIS. Road side sensors like the video camera systems used in [Netten 2011] are simulated by transforming the vehicle positions received from M2M into sensor outputs. The road side sensor outputs and CAMs are fused and the detections of vehicle objects are updated in the world model. The fusion is handled by the Object Detection component (D4.7, section 6.7.1). The world model is a Local Dynamic Map equivalent to the ecoMap.

The application consists of several components (D4.7, section 6.7.1). The Shockwave detector reads vehicle positions from the map to detect perturbations and shock fronts. The shockwave monitor request the output from the detector to monitor the progress of perturbations, e.g. the wave speeds and magnitude of the perturbation. The Profile Generator computes the required speed and headway measure to damp the perturbation. Finally, the message generator encodes a SLAM for the speed profile. The application is decomposed in this way to allow the successive components to run at different time scales and request data at different abstraction levels from the map.

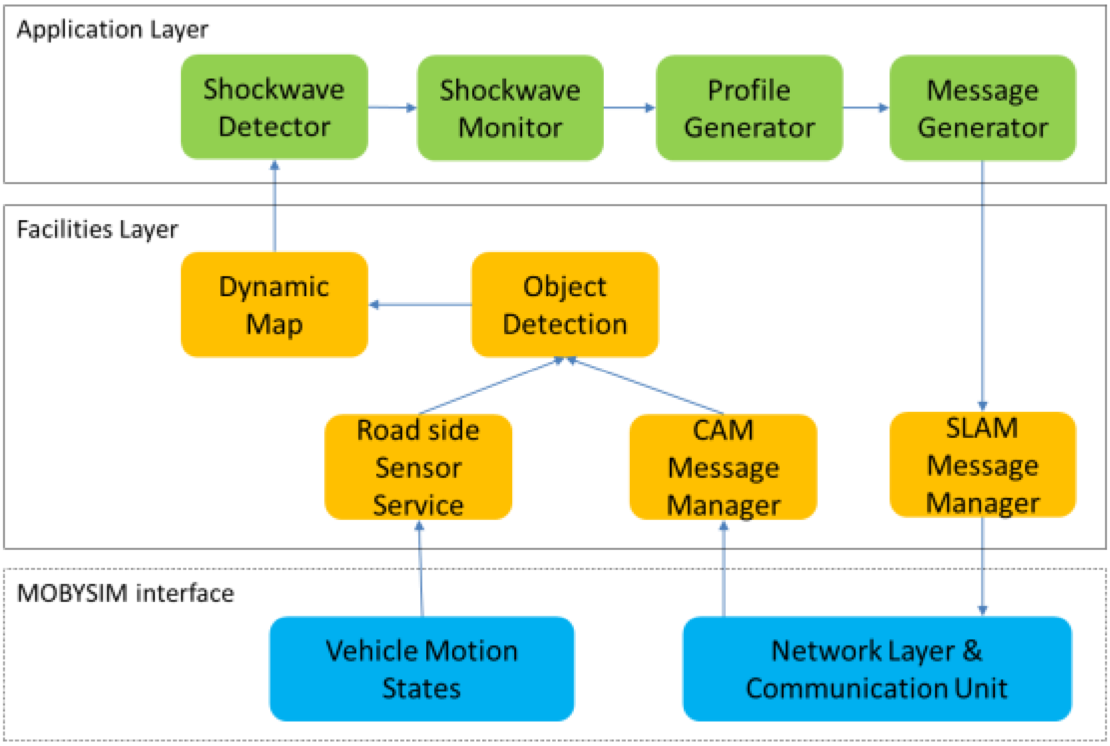

Figure 18 – RS4 environment to simulate the RIS applications and facilities and the interface to MOBYSIM

2.6.2 *User Manual and Adaptability*

The ecoSpeed and Headway Management application components are running on the RS4 simulator. The configuration properties of the application components can be set through the property file of the RS4 simulator (Table 1).

The monitored area for a single RIS is defined by a start and end location, measured as the distance from the beginning of the network (i.e. toll plaza). The exit to the E502 is chosen as the “end location of detection zone” at the location 24 km. The road side communication units are also defined as locations.

The sensitivity of the speed and headway management can be set. The shockwave detector triggers on several conditions. One condition is when a string of vehicles drop below a “trigger speed”. If the application should respond quickly to damp smaller perturbation, for example in combination with CACC equipped vehicles, then the trigger speed and string length can be set to small values. The minimum advised speed can also be set in function of the traffic management strategy; this implies that the speed profile will not drop below this value even when stop and go traffic is detected.

A speed profile can be created in several ways for road side and in-vehicle signalling. If variable speed limit signs are used, the fixed locations can be set and a step profile function is generated. For in-vehicle signage, a linear profile is used. For driver assistance application, the rate at which profiles are adapted can be set. The frequency at which shockwaves are monitored and profiled generated can be set accordingly.

Component	Property	Typical value
RIS		
	Communication Unit location [m] (list of locations)	
Map		
	start location of detection zone [m]	20,000
	end location of detection zone [m]	24,000
	maximum speed limit [km/h]	130
Shockwave Detection and Monitoring		
	minimum monitoring frequency (Hz)	2
	start monitoring location [m]	20,000
	end monitoring location [m]	24,000
	trigger speed [km/h]	70
	minimum string length of perturbation (nr vehicles)	5
Profile and message generation		
	minimum broadcast frequency (Hz)	1
	desired acceleration at inflow [m/s2]	-0.5
	desired acceleration at outflow [m/s2]	0.5
	desired time headway	1.2
	minimum advice speed [km/h]	50
Profile and message generation for driver speed advice		
	minimum update rate [sec]	10
	minimum speed step [km/h]	10
	linear or fixed step profile (choice)	
	fixed speed step location (optional list of VSL signs)	

Table 2 – ecoSpeed and Headway Management application properties

2.6.3 *Practical experience*

Preliminary results indicated that the inefficiencies in traffic flow that this application is able to solve would not present in the simulation environment for Tours. Two approaches were thus followed. Firstly the ecoSpeed and Headway Management application was applied at the A270 test site in Helmond. Secondly, the traffic demands are increased in the simulation for Tours such that the anticipated effects do occur. Incrementing traffic demand resulted in strange behaviour of the vehicles near the exits. Cars and HGVs try to take the exit from the second or even third lane, combined with very low speeds, to make the exit just in time. This causes significant delays and jams on the exits. Discussion with the creators of the model revealed that it was not possible to correct this behaviour without significant programming effort; this behaviour is therefore still present in the model.

The OD matrix for Tours has been adapted to overcome delays near exits in such a way that both the demand is increased and the flows on all entries/exits were eliminated, except for the inflow from the two southern points (tolling plaza and first entry) and the entry where the bottleneck was created

2.7 Application: ecoTruck Parking

The simulation process for EcoTruckParking can be summarized as an extrapolation work. The simulation wants to answer the further question: how many energy (and CO2) can be saved if the application is used. To answer this question, this is needed to start by real measurements: what is the consumption of a truck when it stops, looks for, and doesn't find a parking place. With this basic information, this is possible to evaluate by extrapolation the global amount of saved energy.

2.7.1 *Application setup for simulation*

Application details and preparation:

The Application must have been installed on the OBU:

- OBU to be use : SAMSUNG GALAXY S2
- Application file : ecoTruckParking.apk
- 3G network capability : switched on
- GPS localization: switched on
- ASFA servers : in production

The Application can be used while the simulation is done but this is difficult to find an interest to do that. The measurements of truck consumption can be done independently.

Truck details and preparation:

An extrapolation is executed to determinate the average consumption for a place search.
A series of consumption measurement are made for different supposed situation: "normal areas", "in slope areas" and "big areas". So an average and representative consumption must be determinate by considering:

- the number of the each different type of truck areas of the network
- a weight on each area considering the truck frequentation of each area

VOLVO chooses to do the consumption measurement with a truck specially equipped:

- with a multi-parameter probe (including of course a consumption measurement),
- a GPS tracking system,

- a computer connected to the CAN bus to control and record parameters.

The speed profile of such an event has to follow the sequence cruising/decelerating/low speed/accelerating/cruising. At the end of data validation, VOLVO simulated his vehicle model with the reference truck data set on a driver/scenario model. This is done to extend the data to different truck mass.

The truck is a “Renault Truck Premium DXi11 410hp”. The Truck is classified in the “Distribution” type (such as utilitary or Long haul type). The truck mass was 18 tons corresponding to a quasi-empty situation.

The truck is driven by a professional and a technician is present on board to run and ensure the measurements session like planned.

2.7.2 *User Manual and Adaptability*

The scaling up process must be to extrapolate results to a network level (ASFA one) for a year. For that, the different steps will be:

1. Determinate the global consumption for a place search for the global population of trucks on the network for a year considering mainly the results from section 2.7.1 and the global number of trucks on the network,
2. Determinate the number of needed stops for the whole population of trucks considering mainly:
    - the global number of kilometres trav elled for a year by trucks for the network,
    - the average travel on motorways,
    - the maximum driving time period,
3. Determinate a number of unsuccessful stops for the whole population of trucks considering mainly:
    - the number of needed stops for the whole population of trucks
    - the penetration rate for an application such as EcotruckParking (based on an ASFA survey)
4. Determinate the global fuel consumption avoided by using an application broadcasting the truck parking status such as EcoTruckParking by merging steps 1 and 3.

2.7.3 *Practical experience*

When preparing the extrapolation we face different difficulties:

- Sorting the truck area with the categories defined, the central database of ASFA is not so clear on this point for the entire network. A specific and probably detailed work on this point has to be done.

- Several methods can be used to estimate the global number of truck stops avoided: One considering that the starting point is the number (and locations) of rush hours for truck parkings, another want to start by the global number of whole mandatory truck stops on the network. The most probable thing will be to compare and criticize the two results.

Validation results for ecoTruck Parking are described in [D5.9].

2.8 Application: ecoTolling

The ecoTolling Application supports driver to pass in an energy efficient way a toll station. To support this, nonstop tolling lanes are provided to equipped vehicles which also receive lane and speed advice guiding them to an appropriate toll gate.
The research objective is mainly on reducing the fuel consumption considering the acceptance of system and the compliance by the driver to perform in an envisaged way.

2.8.1 *Application setup for simulation*

No additional development had to be done for ecoTolling. All functionality is covered by the used simulation environment VISSIM.
However the eCoTolling needs to be calibrated regarding the following topics:

- Tollgate architecture (lanes types and configuration modes)
- Tollgate traffic volumes (trucks, light vehicles and T-Tag equipped vehicles distribution)
- Speed limits and decreasing speed advices.

To calibrate the model the number of vehicles that daily cross the barrier (annual average) will be available with the following details:

- Total number of light vehicles
- Total number of trucks
- Total number of vehicles that cross with a T-Tag.

Regarding speed decrease advices defined on the Sorigny simulated test site, when the distance to reach the tollgate is:

- 685m: speed limit is 180km/h
- 560m: speed limit is 90km/h
- 420m: speed limit is 70km/h
- 250m: speed limit is 50km/h.
- 160m: speed limit is 30km/h.

The possible parameters to assess the sensitivity of the model can be:

- lane closure for a period (to be précised)
- disabled vehicle

2.8.2 *User Manual and Adaptability*

Simulation runs will be carried out with the system off and then with the system on. Performance output files of both VISSIM and Enviver (CO2 emissions) will be compared :

- Cooperative vehicle(s) approaching a toll barrier will receive a speed and lane advice and adapt their driving accordingly.
- Collect loss time, number of stops and number of vehicles that have passed the observed tolling. Use Enviver to determine CO2 emissions for this part of the network.

The evaluation period ranges from 30 minutes simulation time after startup to 120 minutes after startup. This allows the simulation network to get completely filled with traffic and controllers and applications to adapt to having traffic everywhere.

While the measurements for CO2 are taken for the whole network, the measurements for number of stops and travel time loss are taken for a stretch upstream covering the complete approach towards the toll barrier and ending shortly after the toll gate.

2.8.3 *Practical experience*

The average measured speed for vehicle crossing the tollgate through the nonstop tolling lanes is between 15 and 21 km/h.

2.9 Application: ecoApproach Advice

Three different implementation variants are distinguished in this section. Each variant is designed for a different baseline or different traffic light control algorithm. They have been developed by different partners but share the same basic principles.

2.9.1 *Application setup for simulation*

Variant 1

The ecoApproach Advice has been developed as two dependent OSGi bundles:

1. The QueueLengthEstimator bundle and
2. The VehicleAllocator bundle

including 8 java classes each.

The VehicleAllocator bundle requires the bundle QueueLengthEstimator:

Figure 19 – OSGi Bundle of the ecoApproach Advice

In addition it requires the bundle eCoMap.The dependencies must be configured in the bundle manifest of the bundle VehicleAllocator:

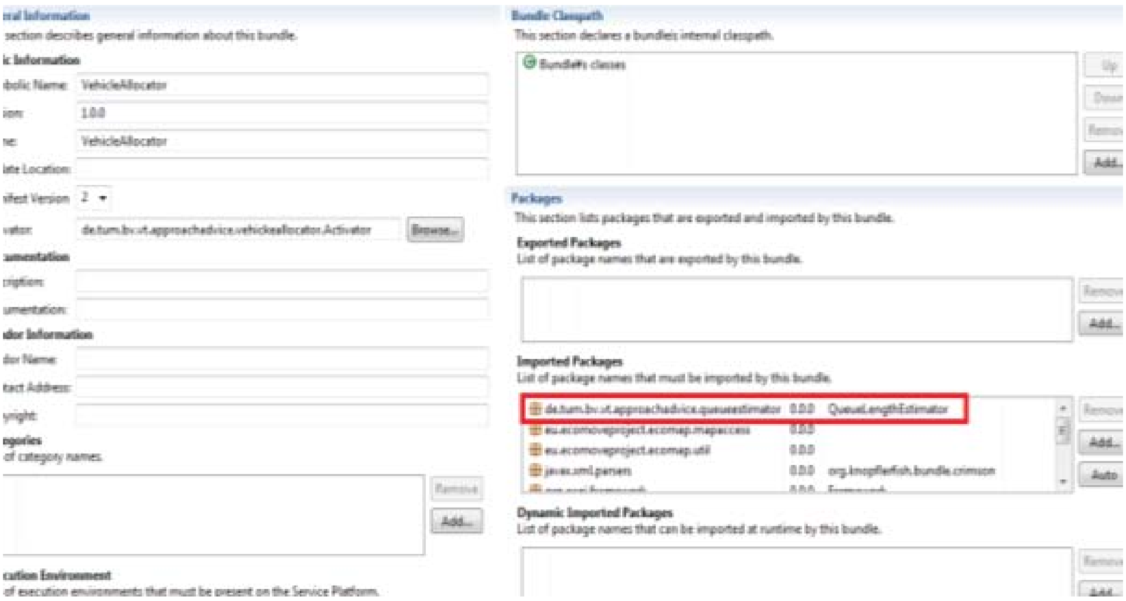

Figure 20 – Dependencies of the Bundle Vehicle Allocator

The integration in the Knopflerfish environment can be seen in the screenshot below:

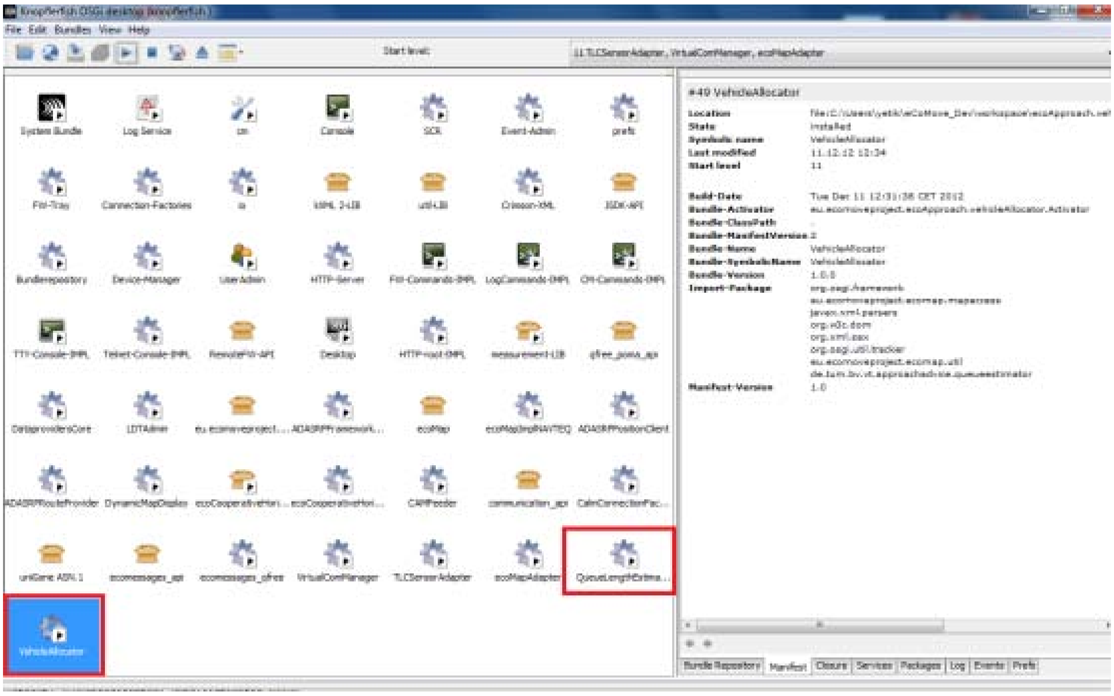

Figure 21 – Integration into the Knopflerfish environment

Both bundles are linked to the VISSIM simulation. They get active when the simulation is started by calling the batch-file. During runtime of the simulation they are feed with dynamic traffic data by the activator classes. The dynamic data include signal state, sensor data from inductive loops and vehicle data (CAM). While the signal state and sensor data is set on change, vehicle data is set each time a C2X vehicle requests a speed and lane advice (SLAM). A C2X vehicle requests an initial SLAM for the next intersection when it is on a link right behind the signal head is located and it requests SLAM updates periodically, when it approaches the next signal head. At SLAM updates, the speed advice may change, but the advised timeslot and the advised lane remains the same, as they are only calculated once for the initial SLAM. The procedure is visualized in the figure below:

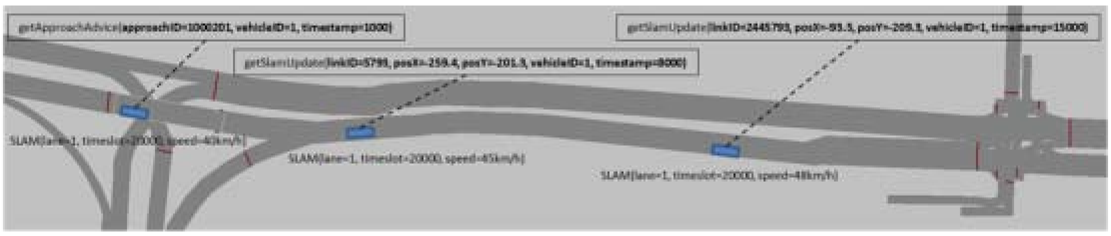

Figure 22 – Visualization of the procedure for a SLAM update.

Variant 2

This application extends upon the ecoBalanced Priority application description. The reader is referred to that section for information with respect to application setup.

Variant 3

For Helmond ecoApproach Advice was based on the predictions of the ImFlow traffic light controller. Due to the dynamic nature of the controller, advice wasn't always possible or reliable. Therefore, when a prediction was more than x (usually 15-30) seconds in the future, no advice was sent to vehicles in order to prevent them from getting wrong advice. Architecture is depicted in the figure below:

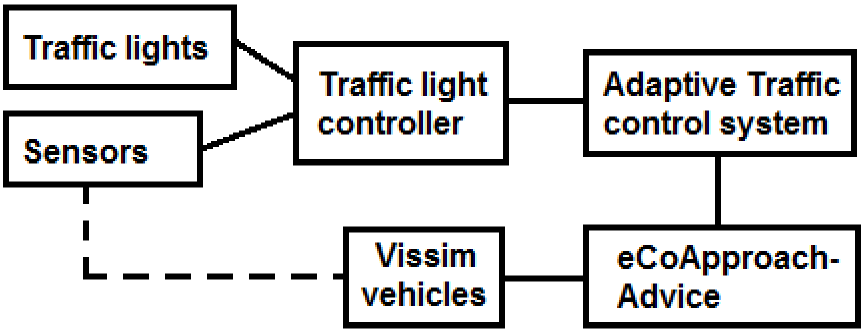

Figure 23 – Helmond ecoApproach Advice Architecture

The traffic lights are signal heads in VISSIM again and together with the sensors connected through the SimInterface.dll to the traffic light controller. The adaptive traffic control system connects to the controller just like in the field. For ecoApproach Advice information about time to green and detection is required. Functionally speaking this information can all come from the Adaptive Traffic Control system, but in practice the connection is actually with the traffic light controller, because it has an easy interface and receives the data from the adaptive system.

With this information about detection and signal planning a queuing model can run in order to make advice for traffic approaching the tail of a queue more accurate. In theory, ImFlow also has a very accurate queue, but in the field also non-ImFlow controllers were used. Therefore, to keep the application generic, a queuing model was included.

Through the VISSIM C2X interface information about the vehicle is transferred to the Java application. This includes current VISSIM lane number, VISSIM (x, y) coordinates, leading vehicle ID and speed. New desired speed is sent back to Vissim by the application. Inside the application the working principles are kept as close to the field implementation as possible. Therefore, VISSIM (x, y) coordinates are converted into (lat, long) coordinates to match the vehicles to approaches defined in the ITM messages which are also sent out on the street. The approach of the ITM can then be linked to the correct SLAM message and matched distance to stop line can be used to find the correct zone in the advice for the vehicle.

Additional for the simulation is the driver behaviour. That is why the current lane is necessary, because the advice is only for traffic going through. A real life driver knows this, but a vehicle in simulation is not aware of these things. Therefore, if the laneID is from a lane that leads to a right- or left- turn, the advice is turned off. This is not perfect, as the direction should actually also be known before entering a lane that leads to a turning signal group, but this is not available on the C2X interface. The current speed and leading vehicle ID are required for modelling anticipating behaviour around the stop line. In the field the advice is simply not shown anymore, but that would make the vehicles in simulation accelerate quickly just to stop again at the tail of the queue or in front of the red light a few instances later. With the leading vehicle ID, of which the application has all information as well, the application can derive the following distance and time. With that information and the current speed a more detailed speed advice can be made that would resemble how a real driver would drive the last meters in front of the stop line. Information about the traffic lights plan is intentionally left out of this equation, because that would give an unfair advantage to the AI driver.

2.9.2 *User Manual and Adaptability*

Variant 1

The ecoApproach Advice is initialized by calling the batch file. The simulation network and the Knopflerfish with the integrated bundles open. The application is executed during runtime of the simulation. It is a precondition, that a certain percentage of equipped vehicles (C2X vehicle) are configured, as these are the only vehicles influenced by the application.
The application requires configuration of the VISSIM network in three files:

- LaneConfigData_Greenwave.xml
- Configuration.java for Vehicle Allocator
- Configuration.java for Queue Estimator

The LaneConfigData_Greenwave.xml contains data about:

- Intersection IDs
- Approaching links (including IDs and Lane numbers)
- Detectors IDs
- Signal group IDs
- Lane types

The Configuration.java for the Vehicle Allocator contains parameters about:

- Cycle time
- Queue balancing factor
- Linking between signal heads and Link IDs
- Saturation flow headways

The Configuration.java for the Queue Estimator contains parameters about:

- Saturation flow headway for platooned and queued vehicles
- Vehicle length
- Deceleration
- Speed limit

- Cycle time
- Correlation settings

During runtime it can be seen that C2X vehicles change their speed. For evaluation VISSIM output files, such as the vehicle protocol and the travel time protocol can be used.
A validation setup foresees the integration with the ecoBalanced Priority. For isolated runtime the signal program is set to fixed time with a cycle time of 90 seconds. For combination of the ecoApproach Advice with the ecoBalanced Priority, the cycle time is no longer fixed, but it must be updated according to the output of the ecoBalanced Priority. The cycle time is used by the VehicleAllocator to determine the next green time of the advised lane.

Variant 2

Two types of ecoApproachAdvice have been implemented. One that is continuously updated as the vehicle approaches the stop line (possible when the vehicle is equipped with an on board unit) and one in which the advice can be sent out only once at a specific location from the stop line (possible when there are variable message signs available at the roadside as is the case in the simulation setup used). In order to come up with an advice the emission of a number of different speed profiles are compared:

1. a speed profile in which a driver *passively* decelerates towards the speed that allows the driver to pass the stop line without further deceleration,
2. a speed profile in which the driver *aggressively* decelerates towards the speed that allows the vehicle to pass the stop line without further deceleration. The speed with which the vehicle can pass the stop line will in this case be higher than when the vehicle decelerates more passively.
3. A speed profile in which the driver decelerates at the last possible moment
4. A speed profile in which the driver starts to decelerate as early as possible.

For cooperative vehicles all of the above speed profiles can be considered. For regular vehicles the first three speed profiles can be considered. Whether a vehicle is cooperative (bidirectional communication possible from vehicle to infrastructure), traceable (unidirectional communication possible from vehicle to infrastructure), or regular (vehicles are detected using regular roadside infrastructure and communication with the driver is done by means of variable messages signs) is determined in VISSIM by choosing the right mix of vehicle types.

In order to be able to choose between the different speed advices `<speed-advice >value</speed-advice>` element must be added to either the `<Junction></Junction>` element of the xml-file used for initialization (if speed advice should be given for the entire intersection), or must be added to the `<Signal></Signal>` element of the xml-file when advice should only be given for certain signals. Allowed values are 0 (= no speed advice), 1 (one shot speed advice), and 2 (continuous speed advice). The element `<distance-`

`advice>value</distance-advice>` can be used to specify the distance at or from which the speed-advice must be given.

Passive deceleration is realized by influencing the speed of a driver in VISSIM indirectly through modification of the desired speed of a driver. Aggressive deceleration is realized by influencing the speed of a driver in VISSIM directly. For all speed profiles minimum and maximum values can be set for both speed and deceleration for the junction as a whole or per signal group by adding `<speed-min></speed-min>`, `<speed-max></speed-max>`, `<acc-min></acc-min>`, and `<acc-max></acc-max>` entries to either the `<Junction></Junction>` element or the `<Signal></Signal>` element of the xml-file

Variant 3

The application is solely made for the Helmond situation and no effort has been made to really productize the application for commercial exploitation. Therefore, most configurations are hard-coded in the application and it cannot directly be implemented in another network.

A detailed manual is not really applicable either then, the code is well documented using javadoc and comments, so that changes of configuration are easy to make. When a queuing model, ITM and SLAM messages have already been configured for the field, it is a relatively easy job of no more than 10 minutes per intersection to adapt the code for a different network. The ITM and queuing model are configured using intuitive XML files, while the SLAM was configured using the .xargs OSGI file in the field (see 4.9.2 for details). However, this simulation application runs without OSGI for improved debugging capabilities, higher stability and faster operation. Therefore, the few SLAM parameters have been configured in the code.

To make this application more easily adaptable to other networks, a configuration file should be made that tells the application which ITM and queue configurations to load. Additionally, relationships between ITMs and SLAMs and queues should also be made explicit. In the field this is easier, because there is only one instance of ITM, SLAM and queuing model per intersection and thus per application. In simulation it runs faster to combine everything in a single application so that more configurations are required.

2.9.3 *Practical experience*

Variant 1

For running the application successfully, it is important that the mapping of the VISSIM link ID to the ADASRP link IDs (=ecomap_path_element_id) is configured in the file LinkMappingInfo.txt. There must be comma as separator sign!

Variant 2

Practical experience has shown that is difficult to give a static advice to vehicles approaching an adaptive intersection as this severely limits the flexibility of the adaptive intersection. Static advices should therefore only be given when the estimated time of departure for the vehicle will change very little as the vehicle approaches the stop line. If there is no possibility to regularly update the speed advice on the approach to the intersection, it is, when given the choice, better, both in terms of delay and CO2 produces, to have the intersection adapt to the approaching vehicles instead of given speed advice.

Variant 3

In practice the ecoApproach Advice application worked quite well in simulation. The driver behaviour issue was something that had to be added after the first evaluations as it was clear that a lot of unrealistic acceleration and deceleration was caused by it. Attempts to use the ecoMap also failed, because for some reason vehicles in VISSIM just didn’t appear in the ecoMap. Therefore, the conversion of VISSIM coordinates to (lat, long) has been added to the application in order to match vehicles to the correct SLAM advice.

2.10 Component: ecoNetwork Prediction

As many different modelling practices exist it is possible to implement a state estimation and prediction system in various ways. This section distinguishes two variants; one implemented by a research institute and another implemented by a traffic systems developer.

2.10.1 *Application setup for simulation*

Variant 1

ecoNetwork Prediction is a component to compute online current, future and optimal / ideal traffic states for a road network. It reads its static configuration data (network topology, road attributes, etc.) from an ASCII configuration file and writes the traffic states into the ecoMap while processing dynamic data (vehicle, detector and signal group states) from the simulation environment.

The component ecoNetwork Prediction is fully integrated in the eCoMove simulation environment. Thus, the simulation environment substitutes a real test site by providing exactly the same dynamic data as the real test site would do. Moreover, the optimized routing schemes, computed by the ecoNetwork Prediction model, can directly be used to change the routing schemes of the simulation environment. This allows examining situations with high penetration rates of equipped vehicles regarding overall traffic CO2 emissions.

In order to integrate the component ecoNetwork Prediction into the simulation environment, the following additional developments have been undertaken:

1. Harmonisation of the static data base → NetState Converter
   The component NetState Converter reads the VISSIM network file (of the eCoMove simulation environment) and creates the ASCII configuration file for the »ecoNetwork Prediction« model. This ensures that the link ids of VISSIM, ecoNetwork Prediction and ecoMap are largely identical, which enables an easy assignment of data. Furthermore, the NetState Converter enriches the VISSIM network file with local routing decisions that are used in operation mode to adjust the routing behaviour of the VISSIM vehicles remotely.

2. Bi-directional interface ecoNetwork Prediction ←→ VISSIMComAdapter
   A socket connection has been implemented in order to exchange data between ecoNetwork Prediction and VISSIMComAdapter. This interface comprises: time synchronisation signals, vehicle / detector / signal group states and routing decision information.

3. Routing decision extensions for VISSIMComAdapter
   The component VISSIMComAdapter has been extended in order to receive routing decisions from ecoNetwork Prediction via socket connection and to adjust local routing decisions of VISSIM periodically via VISSIM COM interface.

4. Various ecoNetwork Prediction extensions for time-synchronisation with VISSIM

The »ecoNetwork Prediction« can work with a virtual time (instead of real time) that is given from the eCoMove simulation environment.

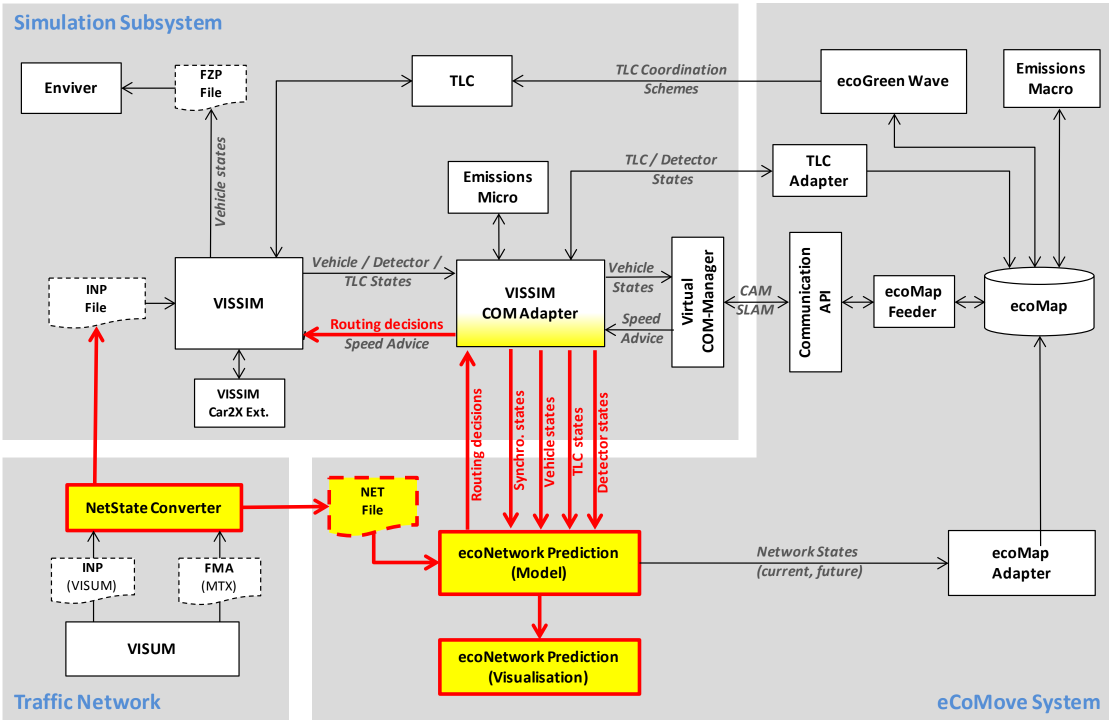

Figure 24 – Integration of component ecoNetwork Prediction in eCoMove simulation environment

Variant 2

This variant is the ecoStrategic Model using the Dynasmart traffic model, as described in D2.10, Traffic Management and Control Simulation Results. It was originally foreseen to couple this component to the simulation environment. However, at the end of the development stage, it was decided not to do the integration with the simulation environment, as it was decided to focus on the implementation in the field in Helmond. There were two reasons for this: (1) the field implementation required more effort than was planned (see section 3.10), and (2) there were problems obtaining the ecoMap, which meant that a connection to the simulation environment could not be made via the ecoMap, and not enough resources were available to make a dedicated connection.

Validation was still executed, so there are no implications for the project due to the mentioned problems.

2.10.2 *User Manual and Adaptability*

Variant 1

In the eCoMove simulation environment there are several batch files in order to start the whole system together with the necessary bundles and applications (ADASRP.eCoMove_Munich.bat, ADASRP.eCoMove_Helmond.bat). Part of this batch job is the launch of ecoNetwork Prediction with a command like »start NetStateMain ..\data\Munich«.

The following items in the configuration ASCII file NetState.conf are relevant to run the ecoNetwork Prediction within the eCoMove simulation environment:

- UseRoutingDecision	*// local routing decisions for VISSIM*
- RoutingDecisionPort	*// local routing decisions for VISSIM*
- RoutingDecisionUpdateInterval	*// local routing decisions for VISSIM*
- SplittingRateEpsilon	*// only update local routing decision if difference // is larger than this value*
- RatioEquippedVehicles	*// The ratio r of equipped vehicles that can send // vehicle generated data (0.0<=r<=1.0)*
- RunWithExtClockSynchro	*// Run the model with external time synchro.*
- InpFile	*// the VISSIM network file*

Variant 2

Not applicable (ecoStrategic Model has not been integrated with simulation environment).

2.10.3 *Practical experience*

Variant 1

The eCoMove simulation environment together with the component “ecoNetwork Prediction” could be realised with all its features and worked as planned. Verification tests have been performed to prove the operation of the system.

It turned out that the online adaptation of VISSIM local routing decisions is not feasible in terms of performance in case of very large VISSIM networks (Munich and Helmond). Therefore, an alternative solution has been realised: While running, the “ecoNetwork Prediction” model writes the routing schemes in the format of VISSIM routing decisions into a separate ASCII file. The content of the ASCII file can be copied into the VISSIM configuration file in order to run the VISSIM scenario a second time with different (optimised) routing decisions.

Variant 2

Not applicable (ecoStrategic Model has not been integrated with simulation environment).

2.11 Component: ecoEmission Estimation and Prediction

2.11.1 *Application setup for simulation*

Micro

The text below refers to the micro version of the ecoEmission Estimation and Prediction component. It is available for use with output from a microscopic traffic simulation model. In the eCoMove project, this would be VISSIM.

The Versit+ code used in Ecomove is a dll. This is a standalone library for software on the Microsoft platform. A similar solution can be developed for use with other software platforms.
The dll can be run in conjunction with an application code. For instance, the ecoNetwork Prediction application uses this dll.

Macro

The text below refers to the macro version of the ecoEmission Estimation and Prediction component. It is available for use with output from a meso- or macroscopic traffic model (specifically Dynasmart and the ecoNetwork Prediction component).

For the eCoMove project, the Matlab routine CO2eCoMove_P.p adds CO2 emissions to output from meso/macro model traffic simulation program DynaSmart or another model that can provide similar output. For CO2eCoMove_P.p the relevant output from DynaSmart is the mean vehicle speed per link and some additional link, node and traffic properties for each time interval. This info is stored in comma-separated-value files (1 row per time interval/link combination), e.g. 'DynaSmartOutputDataFull.csv'.
The routine CO2eCoMove_P.p reads this DynaSmart output and based on TNO's proprietary vehicle emission model VERSIT+ then calculates the corresponding CO2 emissions per time interval/link combination and outputs this in various formats.

2.11.2 *User Manual and Adaptability*

Micro

The dll produces the emissions of a single vehicle velocity trajectory. The interface with the dll requires therefore a time series of the velocity, the vehicle type, and a check code to ensure appropriate access. The interface is low-level and fast: the arrays of data are exchanges by reference. The development of the main code used the associated lib file and a sample code to determine the check code. The dll was compiled under Linux, using the MinGW compiler in combination with the dlltool. The interface or calling sequence, of the dll is the following:

```
FUNCTION:  em(n,tim,vel,acc,inc,veh,emi,check)

  INPUT:
    int n: number of samples
    double tim[0..n-1]: time series [sec]
```

double vel[0..n-1]: velocity series [km/h]
double acc[0..n-1]: acceleration series [m/s2]
double inc[0..n-1]: inclination series [% height:distance]
int veh: passenger car: 1, medium truck: 2, heavy truck: 3, bus: 4
int check: check code (reproduce supplied code on application side)
OUTPUT:
double emi[0..3]: emissions in grams (CO2: 0, NOx: 1, PM10: 2)

Macro

How to run CO2eCoMove_P.m

A complete set of routines and files to run CO2eCoMove_P.p is found in the archive 'eCoMove p-files 4 DynaSmart.zip', which is delivered with the Matlab Routine (see Figure 23 for the list of files). The recipe to install and run the routines is simple and straightforward.

- Unzip the archive in a local folder.
- Start Matlab and go to the previously mentioned local folder or include it in your Matlab path.
- Start the emission addition routine CO2eCoMove_P.p by issuing the command d = co2ecomove_p;
- Doing this without any input arguments will execute the routine with all defaults.
- In the archive a sample DynaSmart output file for testing 'DynaSmartOutputDataFull.csv' is included and this is then used.
- Depending on the speed of your PC, CO2eCoMove_P will take between 3 and 6 minutes (or longer if you have a very slow PC) to process the csv-file and besides the output struct d may yield 3 output files, i.e.
    - DynasmartOutputDataFull_PlusEmissions.mat
    - DynasmartOutputDataFull_PlusEmissions.xlsx
    - DynasmartOutputDataFull_PlusEmissions.csv
- In principle these contain the same information albeit in different formats.
- For eCoMove the most important information is the total emission of CO2 per link per time period, which in the xlsx-file can be found in column O.
- Next, that information can be extracted as a 2D matrix, with time and link number along the vertical resp. horizontal axes, very easily with the routine get2dpar.p by issuing the command d2 = get2dpar(d);
- The routine can optionally store the 2D matrix contained in struct d2 as a mat-file, e.g. 'DynasmartOutputDataFull_CO2_E_Link_Total_(g)_2D.mat',
- Also it can make a 2D colour plot of the CO2 emissions per link per time period as a function of time.
- For more details, see session clipping in section 3. Session Example.

Name	Type
CO2eCoMove_P Quick Reference.pdf	Adobe Acrobat Document
co2ecomove_p.m	M File
co2ecomove_p.p	P File
co2ertable_p1	Microsoft Access Table Shortcut
co2ertbl_p.p	P File
dynasmart2eco.p	P File
DynaSmartOutputDataFull.csv	Microsoft Excel Comma Separated Values File
DynasmartOutputDataFull_CO2_E_Link_Total_(g)_2D	Microsoft Access Table Shortcut
DynasmartOutputDataFull_PlusEmissions	Microsoft Access Table Shortcut
DynasmartOutputDataFull_PlusEmissions.csv	Microsoft Excel Comma Separated Values File
DynasmartOutputDataFull_PlusEmissions.xlsx	Microsoft Excel Worksheet
figure2.p	P File
findcell.p	P File
findfiles.p	P File
get2dpar.m	M File
get2dpar.p	P File
roundnd.p	P File
signatur.p	P File
sigtno.p	P File
sigtnologo.p	P File
sort_nat.p	P File
subdir.p	P File
tno_new_logo.png	PNG image
utc2datenum.p	P File

Figure 25 – List of files used in ecoEmission Estimation and Prediction macro

Session Example

>> d = co2ecomove_p('',1,1,1);

Running CO2eCoMove_P, which calculates CO2 emissions for DynaSmart macro traffic data per link
- csvfilein = DynasmartOutputDataFull.csv
- matfileout = DynasmartOutputDataFull_PlusEmissions.mat
- xlsfileout = DynasmartOutputDataFull_PlusEmissions.xlsx
- csvfileout = DynasmartOutputDataFull_PlusEmissions.csv

CO2eCoMove_P: importing DynaSmart output file DynasmartOutputDataFull.csv
CO2eCoMove_P: import successful

CO2eCoMove_P: starting with calculation and addition of CO2 emission per row

CO2eCoMove_P: first iteration of main loop for 'from' nodes

CO2ERTbl_P: loading co2ertable_p1.mat
- processed row 5000 of 45360

```
…
- processed row 45000 of 45360

CO2eCoMove_P: second iteration of main loop for 'to' nodes

 - processed row 5000 of 45360
…
- processed row 45000 of 45360
CO2eCoMove_P: Saving output struct d to DynasmartOutputDataFull_PlusEmissions.mat

CO2eCoMove_P: Saving matrix data (d.data_hdrs & d.data) to DynasmartOutputDataFull_PlusEmissions.xlsx

CO2eCoMove_P: Saving Dynasmart traffic data together with VERSIT+ emission data to DynasmartOutputDataFull_PlusEmissions.csv

CO2eCoMove_P: All processing finished, bye now
 - Total processing time for this file is 161 s
 - Number of rows (link/time_period combinations) processed is 45360
 - Number of unique time periods and of unique links is 120 resp. 378
 - Time period length info is [min max mean std] = [1  1  1  0] minutes
 - Average processing time used per unique time period is 1.3404 s
>> d
d =
         csvfilein: 'DynasmartOutputDataFull.csv'
        matfileout: 'DynasmartOutputDataFull_PlusEmissions.mat'
        xlsfileout: 'DynasmartOutputDataFull_PlusEmissions.xlsx'
        csvfileout: 'DynasmartOutputDataFull_PlusEmissions.csv'
     calculated_on: '28-Sep-2012 09:38:50'
   co2ecomove_info: 'co2ecomove_p.p, 28-sep-2012 09:35:11'
   co2ertable_info: 'co2ertable_p1.mat, 19-jul-2012 10:33:55, with 8 emission rate tables'
         data_hdrs: {1x23 cell}
              data: [45360x23 double]
>>
>> d2 = get2dpar(d,[],[],[],1);
>> d2
d2 =
     matfilein: 'DynasmartOutputDataFull_PlusEmissions.mat'
    matfileout: 'DynasmartOutputDataFull_CO2_E_Link_Total_(g)_2D.mat'
          xpar: 'LINKNR'
          ypar: 'UTCTIME(s)'
          zpar: 'CO2 E Link Total (g)'
             x: [378x1 double]
             y: [120x1 double]
            zz: [120x378 double]
```

>>

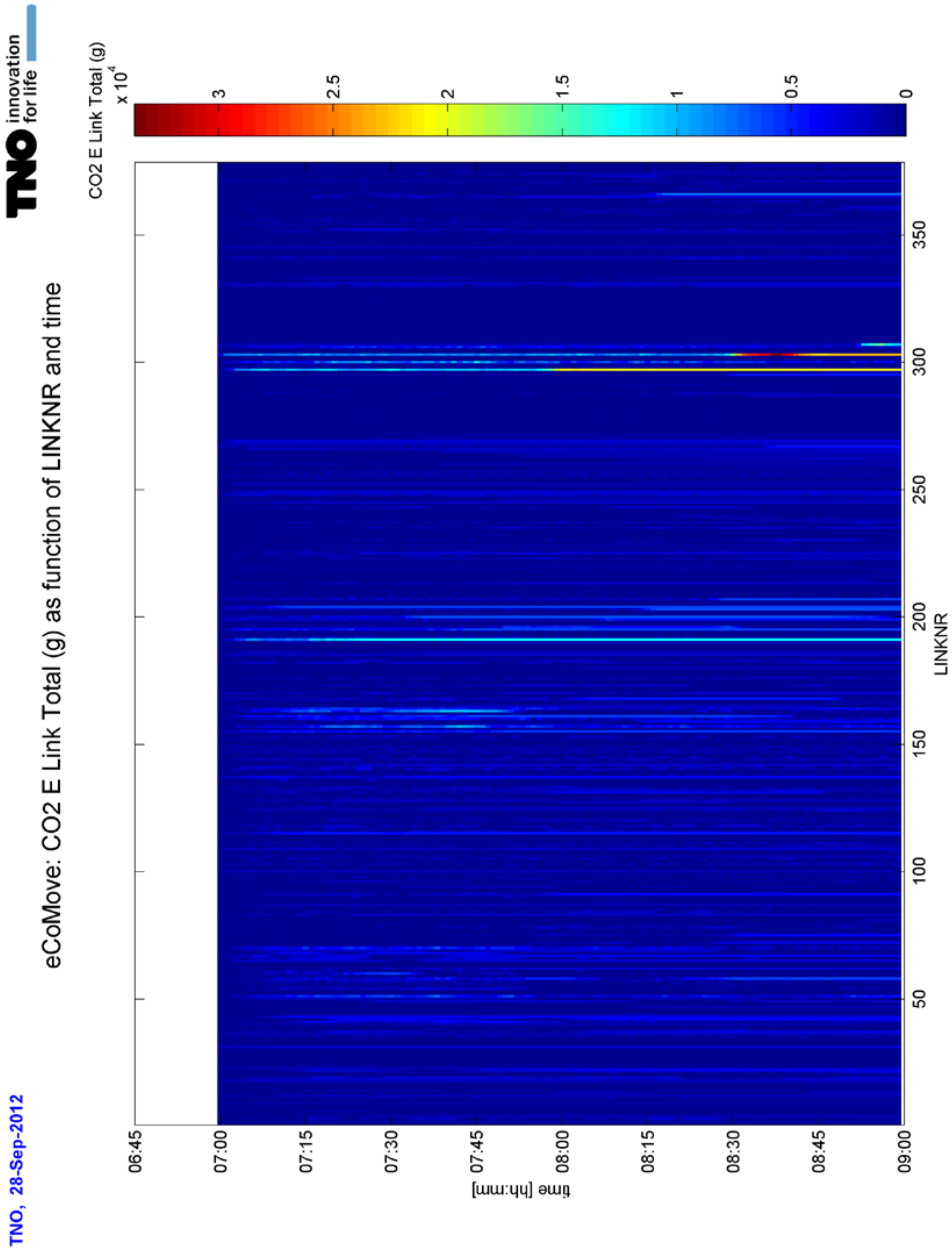

Figure 26 – An example of the data calculated with CO2eCoMove_P.p and extracted and displayed with Get2DPar.p: the total CO2 emission per link (X-axis) per time period as a function of time (Y-axis).

2.11.3 *Practical experience*

Micro

The adaptation of the existing emission model on which the micro version of the ecoEmission Estimation and Prediction component is based was relatively straightforward and no problems were encountered during the testing and validation of it. It was distributed among partners and no problems have been reported.

Macro

The macro version of the ecoEmission Estimation and Prediction component took much more effort to develop and test. This was because extensive microscopic traffic simulation was needed to derive emission function suitable for use at the macroscopic level. Because of the specific requirements, it was not possible to use existing data sets. The finished version has been distributed among partners and although there were a few bugs, these were easy to deal with and the component works well.

2.12 Component: ecoTraffic Strategies

2.12.1 *Practical experience*

Due to technical, organization and practical limitations ecoTraffic Strategies was not implemented. To develop and operate such an application more knowledge was needed about the performance of applications controlled by ecoTraffic Strategies, their response to control targets, the behavior of applications as well as synergies between the applications. Although a design for ecoTraffic Strategies was made, the structure and planning of the eCoMove project did not allow building an actual prototype. Evaluation of the potential of ecoTraffic Strategies will be done using a traffic simulation, but in an indirect manner through the validation of other applications. Emphasis will be on joint operation of applications ecoRoute Advice, ecoGreen Wave, ecoBalanced Priority and ecoApproach Advice. Main aim is to gain knowledge about the performance of applications in response to control targets, their behavior in specific traffic conditions as well as synergies between applications. The role of ecoTraffic Strategies is further discussed on [D5.4] while the [D5.10] contains results from simulations of the single applications as well as combinations of them.

3 Integration at test sites

This chapter describes the test site activities. How were the applications setup for the test sites, what preparations a configurations were made, and what are the most notable practical experiences? It gives an overview of the work performed in WP4. See Table 3.

	Test Site	
Application	**Verification**	**Validation**
Format	*Network*	*Network*
ecoRouting – Network routing	-	-
ecoRouting – Local level routing	Helmond	-
ecoParking Advice	Helmond	-
ecoGreenwave – German variant	-	-
ecoGreenwave – Dutch variant	-	-
Balanced Priority – Variant 1	Helmond	Helmond
Balanced Priority – Variant 2	-	-
ecoRamp-Metering – Green frequeny and speed advice	-	-
ecoRamp – Metering – Virtual Stop lines	-	-
ecoRamp Metering -Truck Priority	-	-
ecoSpeed and Headway Management	-	-
ecoTruckparking	Motorways A7, A46, France	Motorways A7, A46, France
ecoTolling	-	-
ecoApproach Advice – Variant 1	-	-
ecoApproach Advice – Variant 2	-	-
ecoApproach Advice – Variant 3	Helmond	Helmond
ecoNetwork State and Prediction - Variant 1	VISSIM - Helmond	VISSIM - Helmond
ecoNetwork State and Prediction Variant 2	-	-

Table 3. An overview of the applications related to the verification / validation at the test sites

3.1 Note on Test Site Munich

Despite many attempts it was not possible to get the necessary support from local authorities and find a suitable location to setup a test site in Munich. In particular the installation of roadside units was problematic. Hence, it was not possible to verify and validate ecoApproach Advice, ecoBalanced Priority and ecoGreen Wave at a real-world site in Munich as planned. Besides, as it was not possible to include a fleet operator in test site Munich a low penetration rate of vehicles hindered validation of ecoRoute Advice. Therefore only verification will take place in the real test site environment.

3.2 Application: ecoRoute Advice

3.2.1 *Application setup for test site*

Local level routing

When travelling to the business park from the West the route passes a railroad crossing. Due to the frequent railroad traffic the crossing closes often. When the crossing is closed the traffic can reach the destination via a longer route which crosses the railroad via a tunnel. Both routes are shown in Figure 27. The red route is the shorter route which passes the railroad crossing. The alternative route is drawn in blue.

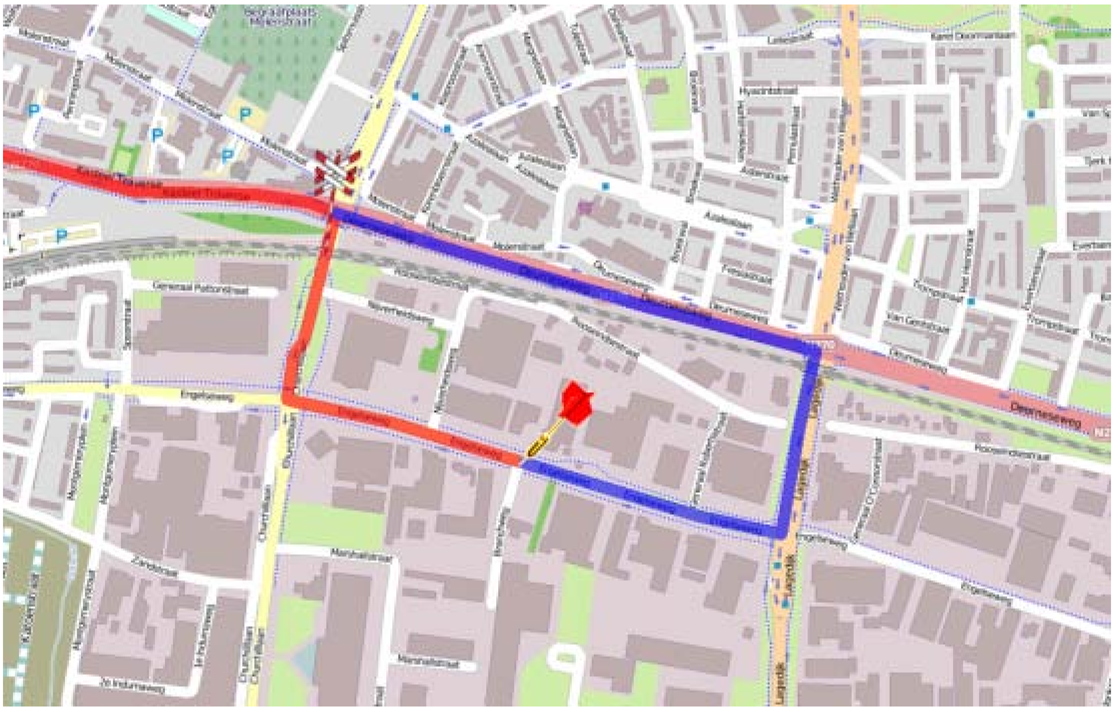

Figure 27 – Alternative routes in Helmond

The traffic controller close to the railroad crossing receives a signal from the railroad crossing when a train is approaching. Therefore it can signal the oncoming traffic with an advice to take the longer route.

3.2.2 *User Manual and Adaptability*

Local level routing

The detour information to bypass the railroad crossing (see Figure 27) is encoded as a TPEG-TEC message. The message contains the alternative route encoded with OpenLR. The XML specification of the message is shown below.


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns1:TpegMLDocument xmlns:ns1="http://www.tisa.org/TPEG/TpegMLDocument_3_0"
xmlns:ns2="http://www.tisa.org/TPEG/AbstractDataTypes_3_0" xmlns:ns3="http://www.tisa.org/TPEG/MessageManagementContainer_3_0"
   xmlns:ns4="http://www.tisa.org/TPEG/TrafficEventCompact_3_0" xmlns:ns5="http://www.tisa.org/TPEG/LocationReferencingContainer_3_0"
xmlns:ns6="http://www.tisa.org/TPEG/TrafficFlowAndPrediction_3_0"
   xmlns:ns7="http://www.tisa.org/TPEG/ServiceAndNetworkInformation_3_0">
   <ns1:TransportFrame xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="ns1:ServiceFrameML">
```

```
      <ns1:EncryptionIndicator>0</ns1:EncryptionIndicator>
      <ns1:ServiceComponent xsi:type="ns1:ServiceComponentML">
        <ns1:ApplicationRootMessage>
          <ns1:ApplicationRootMessageML xsi:type="ns4:TECMessage">
            <ns4:mmt>
              <ns3:messageID>1333</ns3:messageID>
              <ns3:versionID>0</ns3:versionID>
              <ns3:messageExpiryTime>2012-07-04T12:00:00</ns3:messageExpiryTime>
              <ns3:cancelFlag>false</ns3:cancelFlag>
              <ns3:priority ns3:table="mmc001_Priority" ns3:code="1"/>
            </ns4:mmt>
            <ns4:event>
              <!-- Churchillaan closed due to railway crossing; detour recommendation for vehicles driving west on N270 via
Lagedijk/Engelseweg -->
              <ns4:effectCode ns4:table="tec001_EffectCode" ns4:code="7"/>
              <ns4:startTime>2012-07-04T12:00:00</ns4:startTime>
              <ns4:stopTime>2012-07-04T12:00:00</ns4:stopTime>
              <ns4:lengthAffected>50</ns4:lengthAffected>
              <ns4:averageSpeedAbsolute>30</ns4:averageSpeedAbsolute>
              <ns4:delay>300</ns4:delay>
              <ns4:cause xsi:type="ns4:DirectCause">
                <ns4:mainCause ns4:table="tec002_CauseCode" ns4:code="2"/>
                <ns4:warningLevel ns4:table="tec003_WarningLevel" ns4:code="1"/>
                <ns4:unverifiedInformation>false</ns4:unverifiedInformation>
                <ns4:laneRestrictionType ns4:table="tec004_LaneRestriction" ns4:code="1"/>
                <ns4:freeText>
                  <ns2:languageCode ns2:table="typ001_LanguageCode" ns2:code="66"/>
                  <ns2:value>eCoMove detour recommendation</ns2:value>
                </ns4:freeText>
              </ns4:cause>
              <ns4:advice>
                <ns4:adviceCode ns4:table="tec005_AdviceCode" ns4:code="5"/>
              </ns4:advice>
            </ns4:event>
            <ns4:loc>
              <ns5:method xsi:type="ns5:OpenLRLocationReference">
                 <ns5:Version>1</ns5:Version>
                 <!-- Churchillaan -->
                 <ns5:LocationReference>CwQIKCSa/yKRAf/a/6siAQ==</ns5:LocationReference>
              </ns5:method>
            </ns4:loc>
          </ns1:ApplicationRootMessageML>
        </ns1:ApplicationRootMessage>
      </ns1:ServiceComponent>
    </ns1:TransportFrame>
</ns1:TpegMLDocument>
```


The XML message is added to a TPEG message and is broadcast at the relevant intersections.

3.2.3 *Practical experience*

Local level routing

When the diversion XML content was provided to be transmitted a problem arose with the message size: the XML data as provided is 3361 bytes long, which does not fit into a TPEG datagram when transmitted via GeoNetworking. To overcome this problem the TPEG sender was modified to decode and encode the XML contents, allowing the output to be filtered. The encoder contains a list of relevant elements; only these, and their enclosing elements, will be encoded into the TPEG message.

The elements relevant for the receivers in the vehicles are:

```
"ApplicationRootMessageML",
"effectCode",
"messageID",
"versionID",
```

```
"messageExpiryTime",
"cancelFlag",
"LocationReference",
"freeFlowTravelTime",
"delay"
```

Besides this, the message messageExpiryTime element must be filled with a correct date and time. This is done during the encoding process. The resulting TPEG message is shown below; it has an XML payload size of 725 bytes.


```
TPEGM - TPEG Message:
Version: 0
MessageType: TPEGM
GenerationTime: 1362658168308 2013-03-07 12:09:28.308
Management:
  ActionId: SequenceNo: 7867  StationId: 103
  DataVersion: 0
  Reliability: 100
  IsNegation: false
Location:
  EventPosition: EventPositionCurrentDefinition:
    Latitude: 51.477190  Longitude: 5.662690
    Elev: 20.00 m
  LocationRef: RefTrace:
    TraceId: 0
    Waypoints: 0 points
  RelevanceArea:
    GeoAreaCenter: Latitude: 51.477190  Longitude: 5.662690
    Shape: Circle: 300.00 m
    DisseminationAreaSize: SAME
XML TPEG data:
  Length: 725 bytes
<ns1:TpegMLDocument>
<ns1:TransportFrame>
<ns1:ServiceComponent>
<ns1:ApplicationRootMessage>
<ns1:ApplicationRootMessageML xsi:type="ns4:TECMessage">
<ns4:mmt>
<ns3:messageID>1333</ns3:messageID>
<ns3:versionID>0</ns3:versionID>
<ns3:messageExpiryTime>2013-03-07T12:14:28</ns3:messageExpiryTime>
<ns3:cancelFlag>false</ns3:cancelFlag>
</ns4:mmt>
<ns4:event>
<ns4:effectCode ns4:table="tec001_EffectCode" ns4:code="7"></ns4:effectCode>
<ns4:delay>300</ns4:delay>
</ns4:event>
<ns4:loc>
<ns5:method>
<ns5:LocationReference>CwQIKCSa/yKRAf/a/6siAQ==</ns5:LocationReference>
</ns5:method>
</ns4:loc>
</ns1:ApplicationRootMessageML>
</ns1:ApplicationRootMessage>
</ns1:ServiceComponent>
</ns1:TransportFrame>
</ns1:TpegMLDocument>

TPEGM message size = 770
```


Originally it was planned to send out the TPEG-TEC message only when the railway crossing was (going to be) closed. In practice it turned out that this was not possible because the traffic controller close to the railroad crossing (XP104) was being replaced by a new controller during the test period. Consequently the railroad information was not available. To be able to perform tests with vehicles it was decided to transmit the TPEG-TEC message constantly from a number of intersections.

3.3 Application: ecoParking Advice

3.3.1 Application setup for test site

The ecoParking Advice application is provides a freely addressable webservice. This webservice uses the xRoute server from PTV which also provides a webservice.

To hook ecoParking Advice to the test site a configuration for the test site is necessary. The delivered configuration currently holds parkings of the city of Helmond.

The configuration of the ecoParking advice consists of a series of geometries for the parking area's using the WGS84 coordinate system.

The ecoParking advice application runs as a webservice within the eCoMove framework. The webservice URL is configurable in the configuration file of the webcontainer. This configuration is described in the manual of the webcontainer.

From the Traveler Support Manager in the simulation environment calls can be made to the webservice that is provided by the ecoParking advice service. The interface description can be found in the design, implementation and verification documentation.

3.3.2 User Manual and Adaptability

See Section 2.2.2

3.3.3 Practical experience

During testing the following usage and development issue was noted:

- Because of missing parking saturation information the application can still route towards parking areas that are full. In the Netherlands work is on-going to get this information in a public database. In Munich this data was omitted from the ecoMap. As a result ecoParking Advice has no possibility to access this data.

The parking saturation data is now available on some locations in the Netherlands, specifically the cities where the MobiMaestro system is running. The ecoParkAdvice application is being introduced to those cities. In cooperation with the National data Warehouse several eCoMove partners are involved in defining a generic and public service for parking data for all parking facilities in the Netherlands. When this is available ecoParkAdvice can be realized as a generic service useable in the complete country.

3.4 Application: ecoGreen Wave

3.4.1 ***Application setup for test site***

German Variant

No test was executed. Also see section 3.1. The city of Munich did not allow to change any signal times.

Dutch Variant

No test was executed as this was not feasible with the legacy systems presents.

3.5 *Application: ecoBalanced Priority*

As ecoBalanced Priority builds upon existing traffic light control algorithms there are various ways to implement this application. Two variants are distinguished in this section: one developed by a traffic systems vendor and one developed by a research institute. Only the first variant was implemented in the field.

3.5.1 *Application setup for test site*

Variant 1

ecoBalanced Priority covered 14 signalized intersections in Helmond. The equipment of the FREILOT project was re-used and the software upgraded to the latest standards and expanded with eCoMove applications.

Figure 28 – Overview Helmond ecoBalanced Priority intersections

3.5.2 *User Manual and Adaptability*

Variant 1

To provide ecoBalanced Priority at the test site in Helmond the eCoMove software had to be combined with other operational systems and pilots. During the eCoMove test period the infrastructure had to be shared between the following projects:

- FREILOT
- CONTRAST
- eCoMove

The following OSGi bundles have been used for all projects:

- nl.peek.audit.logger; generic logging
- nl.peek.spits.ldm; the LDM (by Peek)
- org.safespot.trail; reference tracks with map matching (by Peek)
- nl.peek.eurocontroller; the TLC proxy
- nl.peek.eurocontroller.api; API of the TLC proxy
- JsonRpc2; JSON used by the TLC proxy
- nl.peek.uts.rpdb; proprietary protocol to connect to the TLC

- org.cvisproject.gpsd; GPS receiver (connects to gpsd)

The following OSGi bundles have been used for FREILOT:

- nl.peek.cvis.rsubeacon; FAST CAM messages
- nl.peek.spits.calmapi; CALM FAST router API
- nl.peek.priority.messages; radio messages for priority service
- nl.peek.priority.roadside; FAST priority service
- nl.peek.keystore; security key material
- nl.peek.vehicle.secret; decoding vehicle priority keys
- nl.peek.logging.messages; radio messages for logging
- nl.peek.logging.roadside; FAST logging service
- org.bn; ASN.1 encoder/decoder

The following OSGi bundles have been used for CONTRAST:

- MessageManager; radio messages with encoder/decoder
- nl.peek.contrast; data collection and message transmission

The following OSGi bundles have been used for eCoMove:

- communication_api; communication API
- connectionfactories_peek; connection to Peek Geonet router
- ecomessages_api; eco messages API
- ecomessages_qfree; eco messages (by Qfree)
- nl.peek.ecomove.messageconverter; data collection and message transmission
- uniGone ASN.1; ASN.1 encoder/decoder

As can be seen from the list above the implementation has been simplified, especially in the area around the LDM. This is possible because the TLC is static; therefore the map only needs to contain the local reference tracks across the intersection.

3.5.3 *Practical experience*

Variant 1

To provide ecoBalanced Priority at the test site in Helmond the practical problem arose how to combine this with other operational systems and pilots. During the eCoMove test period the radio infrastructure had to be shared between the following projects:

- FREILOT using CALM/FAST at 5.89 GHz
- CONTRAST using GeoNetworking at 5.9 GHz
- eCoMove using GeoNetworking at 5.9 GHz

As a first step the radio frequencies had to be the same to be able to share the roadside equipment. This was accomplished by changing the radio frequency of all FREILOT equipment (roadside and on-board) to 5.9 GHz.

Next, the GeoNetworking protocol stack was added to all roadside systems. The CALM FAST and GeoNetworking protocol stacks have been designed such that they can operate simultaneous, using one radio unit.

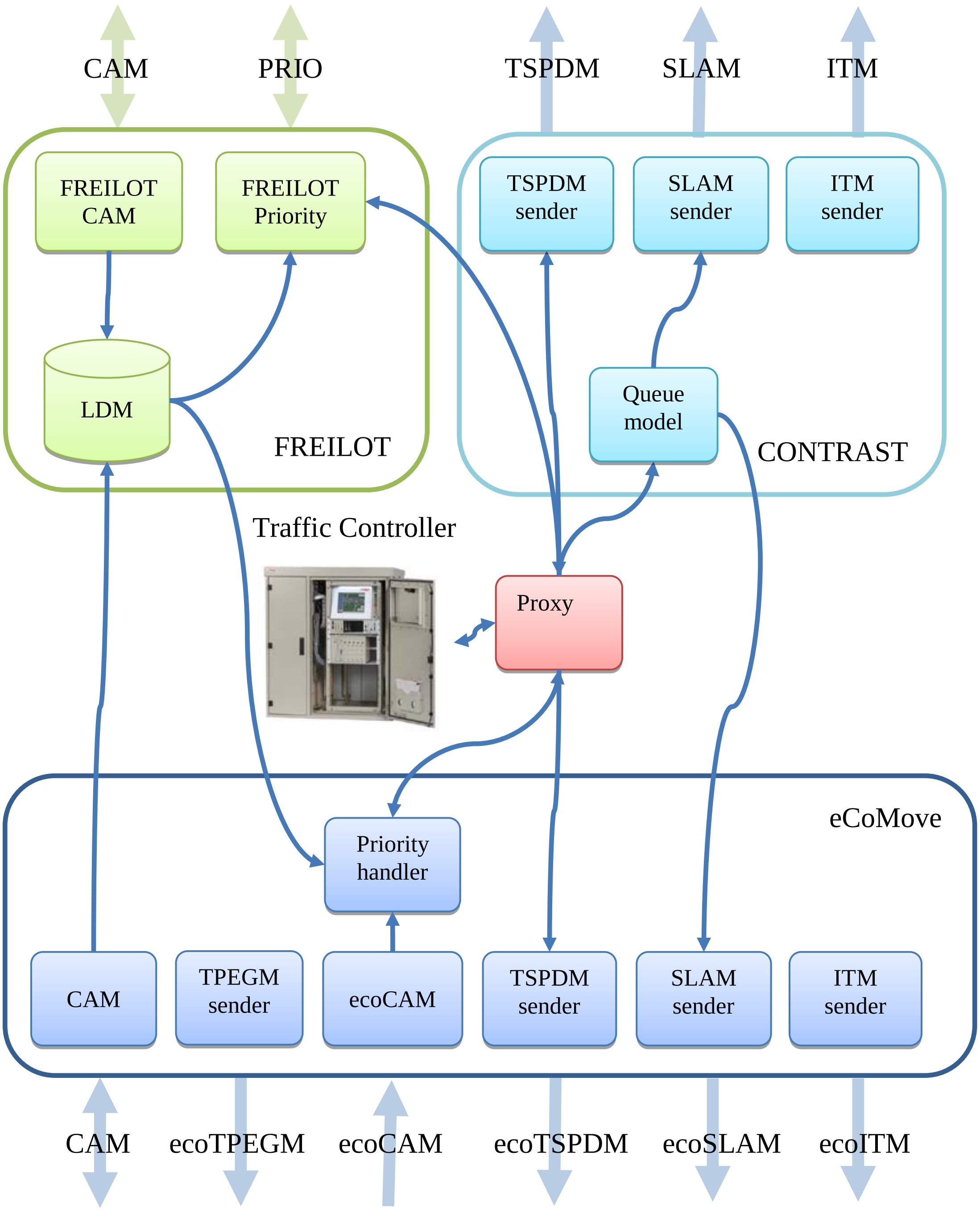

Figure 29 – A roadside unit supporting three different projects

Adding two additional projects to the existing FREILOT infrastructure would increase the number of connections to the traffic controller significantly. As this number is limited a Proxy has been introduced which connects once to the traffic controller, and offers multiple connections (via a JSON interface) to the project specific functions.

Both the CONTRAST and the eCoMove projects send out similar messages (TSPDM, SLAM and ITM) with very minor differences. To allow both projects to use their specific variants the messages are bound to different port numbers.

The final software architecture is shown in Figure 29. Here it can be seen that the Proxy decouples the traffic controller from the projects. The FREILOT LDM is used to store and track the locations of FREILOT and eCoMove road side units and on-board units. The traffic queue model developed for CONTRAST is used to generate the ecoSLAM messages of eCoMove.

When an ecoCAM message is received from an eCoMove vehicle the eCoMove priority handler will verify the location in the LDM and request priority for the vehicle from the Proxy, which in turn hands over the priority request to the traffic controller. Next the traffic controller will propose the priority request to the network control system (ImFlow).

3.6 Application: ecoRamp Metering

Practical experience

Green frequency and speed advice

Testing was planned for the summer of 2013. The implementation of the ecoRamp Metering application for the field test was delayed from August 2012 to June 2013 due to the temporary allocation of the involved staff members to other projects. In July 2013 two eCoMove-partners were asked for support regarding the communication platform and the availability of a vehicle for the field test in August/September 2013. No support was possible. The finalization of the field test has been cancelled. Rijkswaterstaat had not given an answer on the request to perform a field test. Lesson learnt is added to chapter 6 of [D5.4].

Virtual stop lines

Since the Virtual Stop Line Strategy was experimental and with low penetration rates no meaningful validation results could be obtained from real-world tests the application was not implemented at one of the test sites. Also see section 2.5. It had no impact on the project.

Truck priority

Since the Truck Priority Strategy was experimental and with low penetration rates no meaningful validation results could be obtained from real-world tests the application was not implemented at one of the test sites. Also see section 2.5. It had no impact on the project.

3.7 Application: ecoSpeed and Headway management

No test site activities. Application only exists in simulation environment.

3.8 Application: ecoTruck Parking

The test activity is mainly focused on consumption measurement, starting point of the simulation (extrapolation activity)

3.8.1 *Application setup for test site*

The measurement session must be done in normal traffic situation, avoiding days and hours with heavy traffic. The data collected must reflect average search condition. Data representing a long stop due to a blocked traffic can't be representative.

ASFA check in this own traffic database a day corresponding to these characteristics on the motorways A7 and A46 (the date finally chosen was the morning of 9th of May 2012).

The test site is a real site with real conditions, so nothing on the ground have been change or adapted for the measurement session.

ASFA and VOLVO define a specific ride on each truck area (yellow arrows on the following maps). The topography, the ride and the slope situation of an area are supposed to influence the truck consumption. So we have classified each area to the following categories:

- Normal Area
- Big area
- Negative slope Area
- Positive slope Area

This classification will help the extrapolation of the measurements to all kind of truck parking area of ASFA network.

If relevant and probably because of results too different from the "Normal Area" category, it could be possible to create a new category, "Small Area" after analysing the data collection.
Here after an overview of the test sites (A7 and A46 Motorways where truck parking areas are surrounded):

Figure 30 - A map with an overview of the truck parking areas.

Here after, details on each truck parking areas the route planned to be followed (yellow arrows) are presented in the Figure 31 - Figure 36.

Figure 31 - Communay South (A46 motorway)

Communay South (A46 motorway) is classified as a “positive slope truck parking area”

Figure 32 - Saint Rambert d'Albon east and west (A7 motorway)

Saint Rambert d'Albon east and west (A7 motorway) is classified as two "normal truck parking areas".

Figure 33 - Roussillon (Truck Only area, A7 motorway)

Roussillon (Truck Only area, A7 motorway) is classified as a “big truck parking area”. On this area, the truck has to visit all the lines of parking place.

Figure 34 - Pond d'Isère, (A7 Motorway)

Pond d'Isère, (A7 Motorway) is classified as a "normal truck parking area".

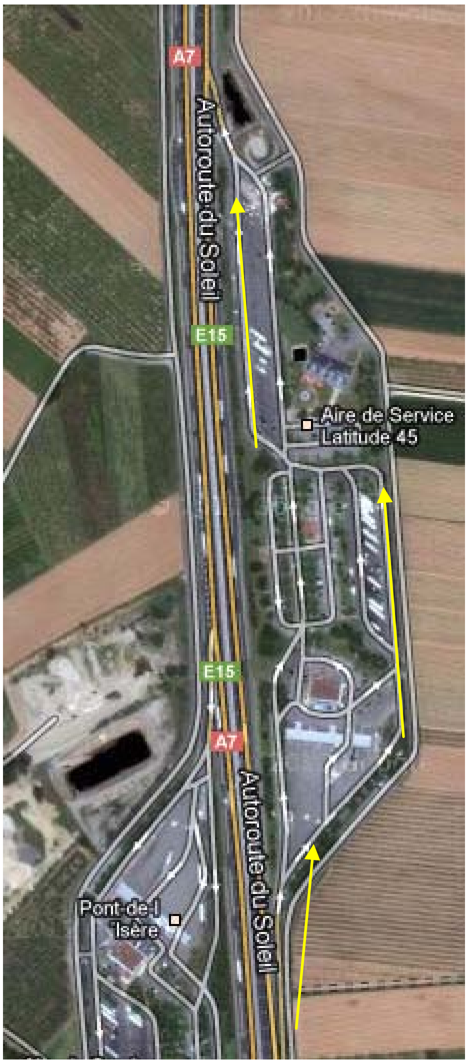

Figure 35 - Latitude 45 (A7 Motorway)

Latitude 45 (A7 Motorway) is classified as a “normal truck parking area”.

Figure 36 - Vienne East and West (A7 Motorway)

The Vienne East and West (A7 Motorway) is Classified as two “normal truck parking areas “, envisaged as “small truck parking area”

3.8.2 *User Manual and Adaptability*

For test site:
The instructions for the measurement were the following:

- Start the measure when the truck exits the cruise velocity at the entrance of the area
- Stop the measure when the truck reaches the cruise velocity at the exit of the area

For the application:
The ecoTruckParking application is design to run on a test site composed by the ASFA/ ASF motorways A46, A7, A9 (approx 2 x 580 km). The On Board Unit user must physically be on these motorways to make a request. The map matching process will reject all requests coming from outside these motorways (excluding of course all the area linked (services and rest area …)).

3.8.3 *Practical experience*

For the test sites:
Difficulties: the availability of an equipped truck accompanied by a skilled team
Facilities: the test site doesn't require any configuration or ground modification.

For the application:
Perhaps the most sensitive part of the EcoTruckParking Application is the map matching module. The map matching process is based on the last GPS three locations returned by the OBU. From this three points the system have to deduce where is the OBU and what is its direction. Two main points play a great role in the accuracy of a map matching process:

- The GPS system location accuracy,
- The road network modeling wholeness.

We choose to base the mapmatching process used for the eCoTruckParking application on a motorway network defined only by the mainline roads themselves. The rest areas are not modelized. A distance tolerance to the roads is used; This tolerance permits to answer two points: the side rest areas are de-facto included in the scope, the poor accuracy delivered by the GPS system is skirted.

These statements give a simple and reliable map matching process able to respond to all the main situations. The limits of such a map matching process are reached for requests made:

- On a roundabout;
- On a rest area where the truck is drived in the opposite direction compared by the main line;.
- In roadwork sections where the two directions of drive have to be temporary modified and doesn't still correspond to network modelling;
- On some areas mixing the traffic flow from the motorway two directions, this is impossible to know where the truck will go.

In all these case, map matching process can't be able to deduce the direction of the truck, and thus no truck park status will be displayed.

3.9 Application: ecoTolling

No test site activities. Application only exists in simulation environment.

3.10 Application: ecoApproach Advice

Three different implementation variants are distinguished in this section. Each variant is designed for a different baseline or different traffic light control algorithm. They have been developed by different partners but share the same basic principles.

3.10.1 *Application setup for test site*

Variant 1

No test site activities. Application only exists in simulation environment

Variant 2

No test site activities. Application only exists in simulation environment

Variant 3

ecoApproach Advice covered 14 signalized intersections in Helmond. The equipment of the FREILOT project was re-used and the software upgraded to the latest standards and expanded with eCoMove applications.

Figure 37 – Overview Helmond ecoBalanced Priority intersections

The ecoApproach Advice application works the same as in simulation (see section 2.9.1) except for the fact the traffic light controller is actually attached to real sensors and signal heads. No special arrangements made in simulation with regard to driver behaviour had to be made in the field.

3.10.2 *User Manual and Adaptability*

Variant 3

To configure the ecoApproach Advice application several parameters have to be set. The ITM and queuing model are configured using intuitive XML files, while the SLAM and the locations of the XML files were configured using the .xargs OSGI file in the field.

The XML file for the queuing model contains data about the names of detectors and signal heads in the traffic light controller and their locations with respect to the queues. For the ITM the configuration contains information about the geographic location and connections between different approaches of (possibly) different ITMs. An example of an ITM configuration is given below:


```
<intersection>
	<stationID>101</stationID>
	<name>Helmond 101, Kasteeltraverse - Zuid Koninginnewal</name>
	<id>101</id> <!--Intersection ID-->
	<!-- reference point-->
	<latitude>51.477668</latitude>
	<longitude>5.657423</longitude>
	<elevation>21.0</elevation>

	<!-- approaches are ordered by intersection arms (usually an intersection has 4 arms, a T-intersection has 3) -->
	<approaches>
		<approach>
			<seqId>1</seqId>
			<approachId>1002</approachId>
			<approachName>SG2</approachName>
			<lane>
				<laneNumber>10020</laneNumber>
				<laneWidth>3.5</laneWidth>
				<laneLength>138.7</laneLength>
				<speedLimit>50.0</speedLimit>
				<wayPoint>
					<ptLat>51.477522</ptLat>
					<ptLon>5.659740</ptLon>
					<ptAlt>20.0</ptAlt>
				</wayPoint>
				<wayPoint>
					<ptLat>51.477705</ptLat>
					<ptLon>5.657756</ptLon>
					<ptAlt>21.0</ptAlt>
				</wayPoint>
				<connectsTo>
					<lane>40000</lane>
					<intersection>101</intersection>
					<maneuver>
						<uTurn>0</uTurn>
						<sharpRight>0</sharpRight>
						<right>1</right>
						<slightRight>0</slightRight>
						<straight>0</straight>
						<slightLeft>0</slightLeft>
						<left>0</left>
						<sharpLeft>0</sharpLeft>
					</maneuver>
				</connectsTo>
```


The numbering and naming of the approaches is important for automatically matching the correct approach to a certain SLAM. Then some additional OSGI parameters are required to let the applications run properly:

```
# parameters required to connect to the 101 RSU
```

```
-
Fnl.peek.ecomove.ITMessage.itmConfigLocation=/home/knopflerfish_osgi_3.0.0/osgi/jars/CONTRAST
/Helmond101_topology.xml
-Fnl.peek.ecomove.ITMessage.itmTransmitPeriodMillis=5000
-Fnl.peek.ecomove.TSPDMessage.tspdmTransmitPeriodMillis=1000
-Fnl.peek.ecomove.queueingModel.queueConfigLocation=/home/knopflerfish_osgi_3.0.0/osgi/jars/
ECOMOVE/Queue101.xml
-Fnl.peek.ecomove.SLAMessage.slamQueues=101008,101002
-Fnl.peek.ecomove.SLAMessage.slamTransmitPeriodMillis=1000
-Fnl.peek.ecomove.geonet.daemonAddress=router
-Fnl.peek.ecomove.geonet.daemonPort=1235
-Fcom.qfree.ecomove.ecomessages.tspdm.port=5102
-Fcom.qfree.ecomove.ecomessages.tspdm.dstaddress.0=geonetworking://shb:5102
-Fcom.qfree.ecomove.ecomessages.slam.port=5104
-Fcom.qfree.ecomove.ecomessages.slam.dstaddress.0=geonetworking://shb:5104
-Fcom.qfree.ecomove.ecomessages.itm.port=5105
-Fcom.qfree.ecomove.ecomessages.itm.dstaddress.0=geonetworking://shb:5105
```

Note that the SlamQueue parameter is the same number as defined for the queue in the queue configuration XML. The ITM number % 1000 should also give the same number as the slamqueue parameter %1000, otherwise the software cannot match the right SLAM to the right ITM approach object.

3.10.3 *Practical experience*

Variant 3

The software was working well in the field. For a FOT project with similar goals it has been stable for many months. Practical experience pointed out that the GPS accuracy and stability of the vehicles was very important, because when the GPS is off, the vehicle cannot be matched to an ITM approach and therefore cannot see an advice. In practice, interpretation of the SLAM message at the vehicle side was challenging but successful after several tests. The only change we requested was to get a more accurate time to stop line, since 1 second accuracy was too few. During verification tests, interference of ecoBalanced Priority was experienced in the sense that green priority changed the time-to-green which made the speed advice unreliable. The three points mentioned are subject for future research.

3.11 Component: ecoNetwork Prediction

As many different modelling practices exist it is possible to implement a state estimation and prediction system in various ways. This section distinguishes two variants; one implemented by a research institute and another implemented by a traffic systems developer.

3.11.1 *Application setup for test site*

Variant 1

ecoNetwork Prediction is a component to compute online current, future and optimal / ideal traffic states for a road network. It reads its static configuration data (network topology, road attributes, etc.) from an ASCII configuration file and writes the traffic states into the ecoMap while processing dynamic data (vehicle, detector and signal group states) from the real test site.

The component ecoNetwork Prediction is fully integrated in the eCoMove system of the real test site. It receives periodically dynamic data from various data sources that are connected to the system via specific adapters (e.g. MunichDynDataAdapter).

In order to integrate the component ecoNetwork Prediction into the real test site, the following additional developments have been undertaken:

1. Reading and writing of dynamic data → ecoMap Adapter
   This component reads/writes dynamic data from/into the ecoMap data base. It is connected with »ecoNetwork Prediction« via socket and uses the ecoMap API to communicate with the ecoMap data server. The following data is read: detector states, signal group states, vehicle states. The following data is written: current and future traffic states.

2. Receiving and converting dynamic detector data from the city of Munich and the Munich motorways → MunichDynData Adapter and ecoLoop Adapter
   This component receives dynamic detector states from the test site Helmond and sends the data directly via socket to the ecoNetwork Prediction. In order to assign the data, specific ASCII mapping files are used.

3. Receiving and converting dynamic detector data from the Helmond test site → »NetState Helmond Adapter«
   This component receives dynamic detector and signal group states from the test site Helmond and sends the data directly via socket to the ecoNetwork Prediction. In order to assign the data, specific ASCII mapping files are used.

4. Local or remote visualisation of »ecoNetwork Prediction« results → NetState Visualisation
   This component receives model results periodically from ecoNetwork Prediction via socket and visualizes the data on a network topology that is read from a separate ASCII file (NET-file). The following types of values can be visualized: traffic flows, travel times, emissions, and speed values. Moreover, the total of emissions over the whole network will be shown.

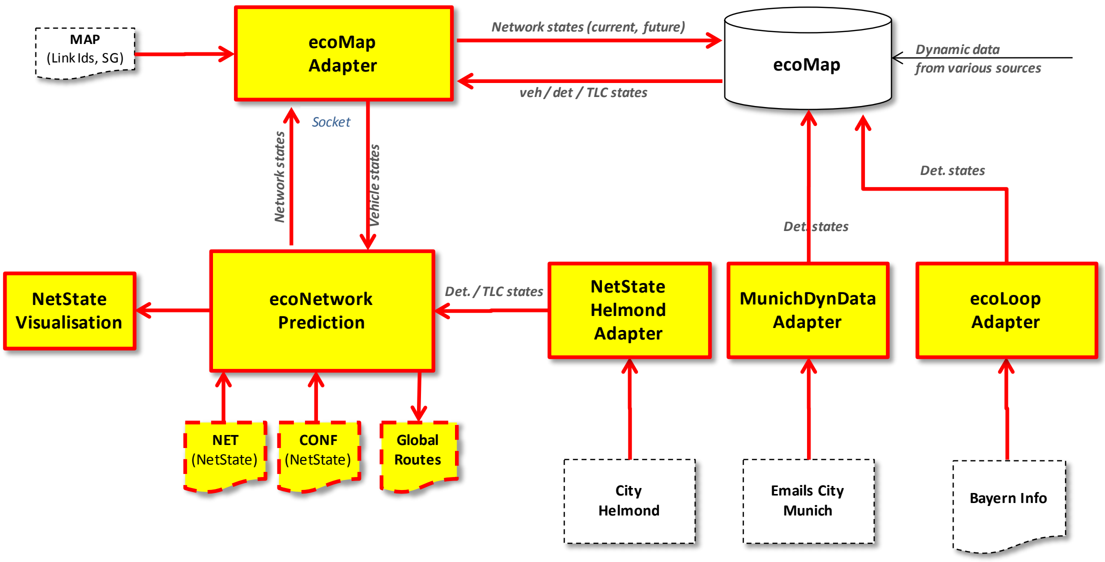

Figure 38 – Integration of component ecoNetwork Prediction in the real test site (Munich, Helmond)

Variant 2

This variant is the ecoStrategic Model developed by TNO, using the Dynasmart traffic model, as described in D2.10, Traffic Management and Control Simulation Results. It was implemented in the field for the Helmond test site. A live feed was set up by Peek Traffic to get real-time detector data from approximately 40 intersections in Helmond to the TNO server. The detector data feeds into the ecoStrategic Model. The output is currently provided as .csv files; if a link with the ecoMap can be established, the output can be stored in the ecoMap so that other applications can use it. The remainder of this section describes the setup of eStraM for simulation. First, an overview of the eStraM modules will be given. Then, we will describe the software dependencies, the inputs and the folder structure of eStraM. In the end we will explain how to visualize the Helmond network and its hotspots.

<u>eStraM overview</u>

Figure 39 depicts a modular overview of eStraM, which consists of the Data Collection and Preparation module, OD Estimation Module, Traffic Prediction Module, Simulation module, Multi objective criteria simulation module, the output module and the hotspot identification module. These modules and their relationships are explained in D2.10, Traffic Management and Control Simulation Results. eStraM is linked to the ecoEmission estimation Module Macro and could in the future be linked to the eCoMap as well. eStraM integrates all these modules into one integrated application, called the ‘shell’. Each separate module as well as the shell itself is programmed in Java as a separate project, but some have dependencies to applications which are programmed and compiled in other programming languages.

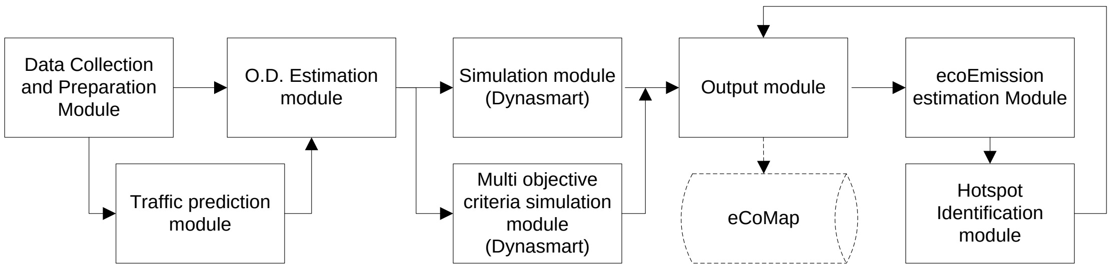

Figure 39 – eStraM modular overview

<u>Software dependencies</u>

The following list shows the dependencies from each separate module to other software:

Module (java project)	Dependency	Description
Data Collection and Preparation module	No dependency	-
O.D. Estimation Module	REMODE_Dyna_OD.exe	OD Estimation software
Traffic Prediction Module	EcoMove_Prediction_Aug2012.exe	Prediction model for count data
Simulation Module	REMODE_Dyna_Sim.exe	Dynasmart Simulation package runnable from command line
Multi objective criteria simulation module	Dynasmart, co2ecomove2x.exe	Module uses Dynasmart but with an adapted cost function, and using emission data provided by the ecoEmission estimation and prediction component
Output Module	No dependency	-
ecoEmission estimation and prediction component	co2ecomove2x.exe	Emission estimation component Macro
Hotspot Identification Module	Hotspot_cars.exe	Hotspot identification program
eStraM shell	All of the above	The Java project combines all separate modules

Table 4 – Software dependency overview eStraM

All eStraM modules and its software dependencies should be available in order to prepare the eStraM application setup for simulation.

Inputs
eStraM needs a number of inputs to set it up for simulation in Helmond. Table 3 shows these inputs.

Input	Used by	Description
lookupTableVxx.csv	Data Collection and Preparation module	Mapping TLC count id's to Dynasmart Network link id's
TLC Count files historical	Data Collection and Preparation module	Count values Helmond from period available in historical database in csv files
TLC Count files live	Data Collection and Preparation module	Count values Helmond per 5 minute per junction in csv file
Dynasmart Network	Simulation Module	Complete Dynasmart simulation network of Helmond

Table 5 – inputs for eStraM

Peek provides, via a live feed, TLC count data to the TNO server, which is refreshed (roughly) every five minutes, providing aggregated data (traffic counts) over the last five minutes.

Folder structure and setup
In order to run eStraM, all modules, software dependencies and inputs need to be placed in the correct folder. Figure 40 shows the directory structure. N.B. Emission Macro Module refers to the macro version of the ecoEmission Estimation and Prediction component.

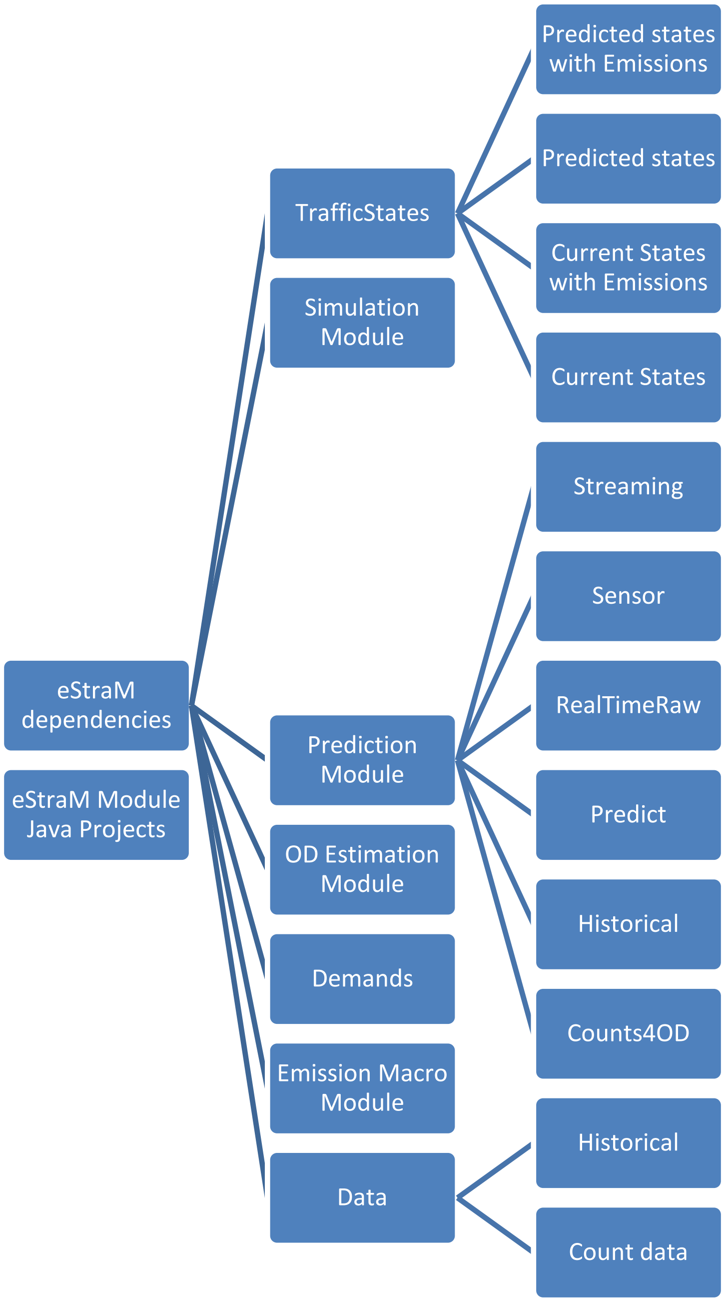

Figure 40 – Directory Structure

The basic entry point for the application is the Application.java class in the ecoStrategicModel project which is located in the eStraM Module Java Projects folder. eStraM runs standard in the offline mode, in which all input data will be processed into traffic state which will be saved in the Traffic States folder.

Visualize network

Dynasmart can be used to visualize the network in order to interpret the results. Figure 41 shows a screenshot of the Dynasmart Simulation software with the Helmond network. For more information on how to work with Dynasmart we would like to refer you to the Dynasmart manual [ref Dynasmart Manual].

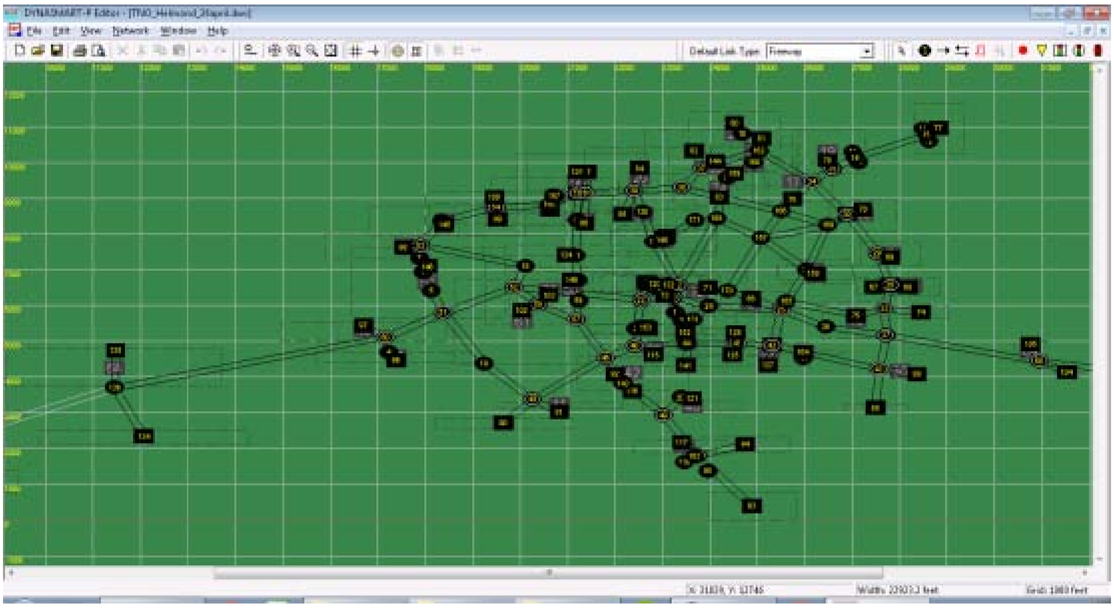

Figure 41 – Screenshot of Helmond network in Dynasmart

The emission hotspots are visualized with the hotspot identification module. Figure 42 shows a screenshot of the hotspots in Helmond from a simulation.

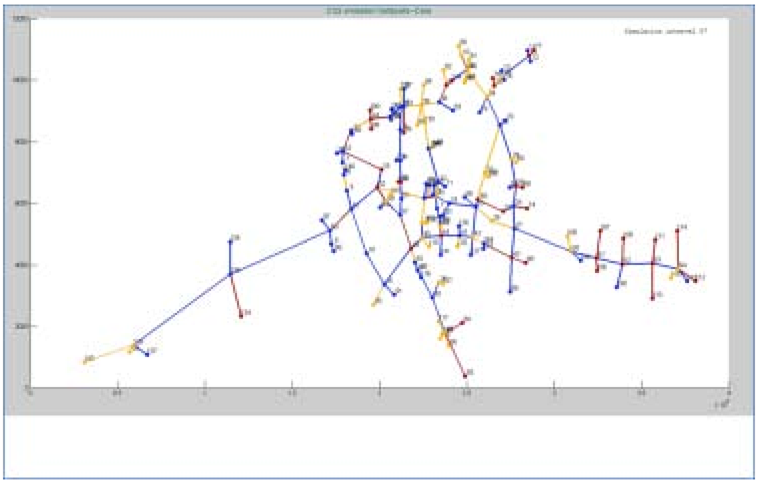

Figure 42 – Visualisation of hotspot identification in Helmond network

3.11.2 *User Manual and Adaptability*

Variant 1

In the eCoMove simulation environment there is a batch files (Start_NetState.bat) in order to start the »ecoNetwork Prediction« component. The OSGi bundles (ecoMap, adapters) must be started separately.

The following items in the configuration ASCII file »NetState.conf« are relevant to run the »ecoNetwork Prediction« within the eCoMove test site environment:

- NetFile *// the static ecoNetwork Prediction configuration*
- NetStateVisuAddress *// data transmission to NetStateVisu*
- NetStateVisuPort *// data transmission to NetStateVisu*
- RunWithExtClockSynchro *// Run the model with external time synchro.*
- SignalGroupPort *// port to receive dynamic data*
- DetectorPort *// port to receive dynamic data*
- WriteRoutingSchemes *// write routing schemes to an external ASCII file*
- TypeOfRouteLinks *// type of link references for routing schemes*
- RunWithPrediction *// also estimate a prediction*
- PredictionHorizon *// the prediction horizon in minutes*
- ConsiderDetectors *// consider dynamic detector values*
- SimulationThreads *// parallel computation scheme*
- TrafficStateUpdateInterval *// time interval for traffic state update in sec.*
- LinkMappingFile *// which mapping file to use*

Variant 2

The preparation for test site operation of the ecoStrategic Model includes the following main steps:

- Prepare (live) data feed (TLC data, or similar data from loops or other sources of data);
- Install the shell and all necessary elements (see Figure 40), including the Dynasmart model and ecoEmission Estimation and Prediction component, macro version;
- Adapt the settings file in java (paths, parameters).

For any new network, a calibrated model has to be made available and mapping tables have to be made to link input data to the network model elements.

Ideally, the output module feeds into the ecoMap, but this connection has not been made for the ecoStrategic Model in Helmond.

3.11.3 *Practical experience*

Variant 1

The eCoMove component “ecoNetwork Prediction” could be realised with all its features and worked as planned. Verification tests have been performed to prove the operation of the system.

Variant 2

At the moment of writing, the following observations about the practical experience can be made:

- There were many difficulties translating the network from VISSIM to Dynasmart. These included mapping issues, but also the fact that it took a long time for the VISSIM model to be finalized. It involved a much manual work.
- There were difficulties in getting the detector data real-time and in a suitable format. Here, we also encountered many mapping issues.
- For an on-line version of eStraM, the problem of timing of detector data needs to be sorted out. Currently, the live data provided does not come in at a consistent 5 minute interval (sometimes there is a random gap of random duration, so extra efforts are needed to interpret the data).
- Attempts were made to get floating vehicle data, were not possible or required too much effort.
- It took very long to finalise the NDA for the ecoMap issues. This involved much administrative effort, resulted in long delay which resulted in no use of the ecoMap.
- There was a poor match between the timing of work and deliverables: this is an R&D project, but with much focus on requirements and architecture documentation, at moments when a full design was not yet available due to uncertainty about available data. Also, the test site networks needed were not available in design phase.
- It was decided to focus on getting the eStraM to work for validation. A full on-line version may be a bridge too far within this project (given time and budget restraints), although the work that needs to be done to achieve this is not complicated, technically.
- Much time was spent on getting the network and the input data available and in the right format, and also in processing it in the right way.

On a positive note:

- eStraM was implemented and all of the modules work.
- eStraM works fast enough on a normal pc (the sort of pc that is easy to set up).
- There was successful cooperation with partners (though with substantial efforts needed from both sides).

Despite the number of problems, validation was done, results are in D5.9 (field implementations) and D5.10 (for the simulation implementation). Datasets for field implementations were small but results are consistent. Lessons learnt (main one: using live traffic data requires substantial efforts) will be useful in future projects.

3.12 Component: ecoEmission Estimation and Prediction

3.12.1 ***Practical experience***

See section 2.11. It works the same way wherever it is implemented.

3.13 Component: ecoTraffic Strategies

3.13.1 ***Practical experience***

Due to technical, organization and practical limitations ecoTraffic Strategies was not implemented. Also see section 2.12.

4 Final Conclusions

The integration of 7 applications and 3 components in simulation is described in this document. The practical experiences show valuable results.
For test, development or demonstration systems a free and open routing engine is more suitable than a commercial product as this gives more flexibility in test site usage. However there is no guarantee that the interface will remain the same. For production systems this is unacceptable so ‘a’ product should be chosen to provide the green routing algorithm as this is no longer part of the eCoMove toolkit.
Practical experience has shown that is difficult to give a static advice to vehicles approaching an adaptive intersection as this severely limits the flexibility of the adaptive intersection. Hence, there exists a trade-off between flexibility and predictability traffic lights.

The integration of 5 applications and one component at test sites is described in this document. The practical experiences show valuable results.
Lessons learned are to check at the start of the project the feasibility of the demonstration and to get commitment from all actors. Also the availability of live traffic data and the accuracy of GPS are important.
It shows the way how to adapt and to use one test site for several European projects.