## D5.7

**Development and Testing of Traffic Management and Control Measures [DRAFT]**

<table>
<thead>
<tr>
<th>SubProject No.</th>
<th>SP5</th>
<th>SubProject Title</th>
<th>EcoTraffic Management and Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workpackage No.</td>
<td>WP4</td>
<td>Workpackage Title</td>
<td>Implementation</td>
</tr>
<tr>
<td>Task No.</td>
<td>T5.4.3</td>
<td>Task Title</td>
<td>Develop the Simulation Environment for Testing and Evaluation</td>
</tr>
<tr>
<td>Authors</td>
<td>Jonas Lüßmann, Jaap Vreeswijk, Paul Mathias, Matthias Mann</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dissemination level PU/PP/RE/CO</td>
<td>CO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>File Name</td>
<td>120620-DEL-SP5-D5.7-Development and Testing of Traffic Management and Control Measures [DRAFT]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Due date</td>
<td>31/03/2012</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delivery date</td>
<td>15/06/2012</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

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
Abstract

This deliverable will contain: the instructions for setting up a simulation environment in which the systems can be tested and validated, basic data-sets and parameters for setting up the test-bed for the SP5 systems. And present the Traffic Management and Control Measure implemented within the eCoMove extended simulation environment and the test results.
## Control sheet

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Main author</th>
<th>Summary of changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>24.05.2012</td>
<td>Matthias Mann</td>
<td>Initial input</td>
</tr>
<tr>
<td>0.2</td>
<td>08.06.2012</td>
<td>Paul Mathias Robin Blokpoel</td>
<td>Integration of partner contribution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Matthias Mann</td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>13.06.2012</td>
<td>Matthias Mann</td>
<td>Com Manager and Map Feeder</td>
</tr>
<tr>
<td>0.4</td>
<td>14.06.2012</td>
<td>Matthias Mann</td>
<td>Improvement TS Munich</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prepared</td>
<td>Matthias Mann</td>
</tr>
<tr>
<td>Reviewed</td>
<td>n/a (Jaap Vreeswijk)</td>
</tr>
<tr>
<td>Authorized</td>
<td>Jean-Charles Pandazis</td>
</tr>
<tr>
<td>Verified</td>
<td>Manuela Flachi</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Recipient</th>
<th>Date of submission</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project partners</td>
<td>15/06/2012</td>
</tr>
<tr>
<td>European Commission</td>
<td>20/06/2012</td>
</tr>
</tbody>
</table>
# Table of Contents

## 1 INTRODUCTION
- 1.1 SYSTEM OVERVIEW .....................................................8
- 1.2 DOCUMENT OVERVIEW .............................................8
  - 1.2.1 Intended Audience .............................................8
  - 1.2.2 Document Structure .............................................8

## 2 REFERENCED DOCUMENTS
- 2.1 eCoMove DELIVERABLES ........................................8
- 2.2 eCoMove REFERENCE DOCUMENTS ..........................9

## 3 FUNCTIONAL DESCRIPTION OF THE MICRO SIMULATION ADAPTER
- 3.1 ROLE OF SIMULATION IN eCoMove ............................11
- 3.2 FUNCTIONAL REQUIREMENTS ON MS ADAPTER ..........11
- 3.3 OVERVIEW OF THE MS-ADAPTER ..............................12
- 3.4 VISSIM INTERFACES (PTV) .......................................12
  - 3.4.1 VISSIM Com Interface .........................................12
  - 3.4.2 VISSIM Car2X Module .........................................13
  - 3.4.3 VISSIM Driver Model DLL Interface .......................13
  - 3.4.4 External Signal Control DLL Interface .....................13
  - 3.4.5 ANM file interface .............................................14
- 3.5 COMPONENTS OF THE MS-ADAPTER .........................14
  - 3.5.1 VISSIM COM Adapter .........................................14
  - 3.5.2 External FTC (‘extFTC’) ......................................14
  - 3.5.3 ecoNetwork Prediction .........................................14
  - 3.5.4 ecoMap Netstate Adapter .....................................15
  - 3.5.5 TLC Adapter ....................................................15
  - 3.5.6 NetState Converter ............................................15
  - 3.5.7 Virtual COM Manager ..........................................15
  - 3.5.8 Subchapter on Peek TLC (PEEK) .........................16
  - 3.5.9 Subchapter on COMManager .................................16
  - 3.5.10 Subchapter on Map Feeder (NAVTEQ) .................16

## 4 TEST SIDE MODELLING ..................................................17
- 4.1 HELMOND .............................................................17
- 4.2 MUNICH ...............................................................18
- 4.3 FRENCH MOTORWAY ..............................................20

## 5 INSTALLATION INSTRUCTIONS FOR SETTING UP THE SIMULATION ENVIRONMENT .................................................................23
- 5.1 INTRODUCTION ........................................................23
- 5.2 PREREQUISITES .......................................................23
- 5.3 GENERAL .............................................................23
- 5.4 ecoMAP .................................................................23
  - 5.4.1 Installation .........................................................23
  - 5.4.2 Installation .........................................................24
5.4.3 Start................................................................. 24
5.4.4 Remark ............................................................ 24
5.5 KNOPLERFISH ...................................................... 24
  5.5.1 Configuration ................................................. 24
  5.5.2 restart.xargs .................................................. 27
5.6 VISSIM ................................................................. 27
  5.6.1 Remarks .......................................................... 27
  5.6.2 Start ............................................................... 27
  5.6.3 Operation ....................................................... 27

6 VERIFICATION OF ECOMOVE APPLICATION AND
COMPONENTS ....................................................................................... 28
  6.1 VERIFICATION OVERVIEW FOR ecoROUTEADVICE (TUM) ......................... 32
  6.1.1 Verification Test 1: Title ....................................... Error! Bookmark not defined.
  6.2 VERIFICATION OVERVIEW FOR ecoPARKADVICE (TECHNOLUTION) .......... 32
  6.3 VERIFICATION OVERVIEW FOR ecoGREEN WAVE (MAT) .......................... 32
  6.4 VERIFICATION OVERVIEW FOR ecoBALANCED PRIORITY (PEEK) ............ 32
  6.5 VERIFICATION OVERVIEW FOR ecoRAMP METERING (VIALIS) ................. 32
  6.6 VERIFICATION OVERVIEW FOR ecoSPEED AND HEADWAY MANAGEMENT
  (TNO) .................................................................................. 32
  6.7 VERIFICATION OVERVIEW FOR ecoTRUCK PARKING (ASFA) .................... 32
  6.8 VERIFICATION OVERVIEW FOR ecoTOLLING (ASFA) .................................. 32
  6.9 VERIFICATION OVERVIEW FOR ecoGREEN WAVE (MAT) ......................... 33
  6.10 VERIFICATION OVERVIEW FOR ecoAPPROACH ADVICE (TUM) .............. 33
  6.11 VERIFICATION OVERVIEW FOR ecoNETWORK PREDICTION (MAT) ............. 28
     6.11.1 Verification Test 2: Estimation of current and future traffic states ....... 29
     6.11.2 Verification Test 3: Estimation of ideal / desired traffic states .......... 29
  6.12 VERIFICATION OVERVIEW FOR ecoEMISSION ESTIMATION (TNO) ............ 33
  6.13 VERIFICATION OVERVIEW FOR ecoTRAFFIC STRATEGIES (PEEK) ........... 33
  6.14 VERIFICATION OVERVIEW FOR ecoMAPS (NAVTEQ) ............................ 33
  6.15 SUMMARY OF VERIFICATION TESTS .................................................. 33

7 APPENDIX ................................................................................................. 34
FIGURES

Figure 1: Figure Description .......................... Error! Bookmark not defined.

TABLES

Table 1: Finalised eCoMove Deliverables.............. Error! Bookmark not defined.
Table 2: Future eCoMove Deliverables .................. Error! Bookmark not defined.
Table 3: eCoMove reference documents ........................ Error! Bookmark not defined.
Table 4: Data available at the SVN server ............... Error! Bookmark not defined.
Table 5: Requirements on the ecoSituational Model .......................... 11
Table 6: Applications interfering with APPLICATION XYZ Error! Bookmark not defined.
# TERMS AND ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1 Introduction

1.1 System Overview
The eCoMove system is designed to tackle the problem of energy efficiency in road transport by applying the latest vehicle-to-infrastructure and vehicle-to-vehicle communication technologies to create an integrated solution comprising cooperative eco-driving support and eco-traffic management. The project aims to demonstrate that the combination of these new intelligent communication technologies can potentially lead to overall fuel savings and CO₂ emission reductions of up to 20%.

1.2 Document Overview
This document especially focuses on the simulation environment and the outcome of component and application verification.

1.2.1 Intended Audience
The intended audience of this document are...

1.2.2 Document Structure
This document is arranged as follows:
- **Chapter 2** summarizes the referenced documents: deliverables, external documents, and available online development tools.
- **Chapter 3** gives a functional description of the micro simulation adapter and its components as they have been developed.
- **Chapter 4** presents detailed information about the modeling of the simulation test sites Helmond, Munich and the French Motorway.
- **Chapter 5** describes the installation instructions for setting up the simulation environment.
- **Chapter 6** provides an overview and the results of the verification of eCoMove application and components from a simulation perspective.
2 Referenced Documents

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

2.1 Referenced eCoMove Deliverables

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

<table>
<thead>
<tr>
<th>Ref</th>
<th>Doc</th>
<th>Version, Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>[D2.1]</td>
<td>System concept, functionality, requirements and use case description</td>
<td>V0.6, 2010-11</td>
</tr>
<tr>
<td>[D2.2]</td>
<td>High Level Architecture</td>
<td>V1.1.1, 2011-05</td>
</tr>
<tr>
<td>[D5.1]</td>
<td>SP5 Use cases &amp; System Requirements</td>
<td>V3.0, 2011-10</td>
</tr>
<tr>
<td>[D5.2]</td>
<td>SP5 Architecture and System Specifications</td>
<td>V2.0, 2011-03</td>
</tr>
<tr>
<td>[D5.3]</td>
<td>Extension of Simulation Functionalities and Test Site Modelling</td>
<td>V3.6, 2011-07</td>
</tr>
</tbody>
</table>

Table 1: Finalised eCoMove Deliverables

<table>
<thead>
<tr>
<th>Ref</th>
<th>Doc</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>[D2.13]</td>
<td>Final report ecoMessages</td>
<td>2012-06</td>
</tr>
<tr>
<td>[D5.4]</td>
<td>SP5 System Prototype Implementation</td>
<td>2013-01</td>
</tr>
<tr>
<td>[D5.6]</td>
<td>SP5 Prototype Implementation of the applications and components</td>
<td>2012-03</td>
</tr>
</tbody>
</table>

Table 2: Future eCoMove Deliverables

2.2 eCoMove Reference Documents

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

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

### 2.3.1 TRAC

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

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

### 2.3.2 Subversion

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

The eCoMove subversion is also hosted and maintained by ika and is split up into the different subprojects of eCoMove. Table 4 summarizes the SVN servers referenced in this document.

<table>
<thead>
<tr>
<th>Ref</th>
<th>Doc</th>
<th>Version, Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>[SP2_SVN]</td>
<td><a href="https://svn.ika.rwth-aachen.de/trac-ecomove/ecomove">https://svn.ika.rwth-aachen.de/trac-ecomove/ecomove</a></td>
<td></td>
</tr>
<tr>
<td>[SP3_SVN]</td>
<td><a href="https://svn.ika.rwth-aachen.de/trac-ecosmartdriving/ecosmartdriving">https://svn.ika.rwth-aachen.de/trac-ecosmartdriving/ecosmartdriving</a></td>
<td></td>
</tr>
</tbody>
</table>
3  Functional Description of the Micro Simulation Adapter

This chapter summarizes the purpose of the Micro Simulation Adapter and defines the in- and outputs of this component as defined in the system architecture.

3.1  Role of Simulation in eCoMove

As there is a only a limited number of eCoMove equipped vehicles for developing and testing the eCoMove applications in a real environment, traffic simulation plays an important role in the process of the development of functions, in the preparation of field trials and the final impact assessment of the eCoMove System on network level. Therefore, simulation sub-systems will be set up in eCoMove to support the development, the in-loop-testing, the verification of infrastructure functional components and applications and the validation of the whole system.

In order to perform a large scale evaluation of the concept a microscopic simulation environment is being implemented that shall models the reality with installed eCoMove applications as realistic as possible. The main requirements for the simulation environment are (1) to integrate the eCoMove applications without any changes from their test site implementation, (2) to be able to influence the behaviour of the vehicles according to real implementation behaviour, and (3) to systematically assess the impact of eCoMove applications.

3.2  Functional requirements on MS Adapter

Table 5 provides a list with functional requirements on the simulation environment and the micro simulation adapter. References to the requirements can be found in deliverable [D5.3].

<table>
<thead>
<tr>
<th>REQ NR.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECOM-RQ-SP5-0112</td>
<td>The micro simulation models should be capable of modelling traffic movements, traffic-related inefficiencies and the eCoMove applications and components in a realistic way.</td>
</tr>
<tr>
<td>ECOM-RQ-SP5-0114</td>
<td>The simulation environment support developers in testing their applications and components.</td>
</tr>
<tr>
<td>ECOM-RQ-SP5-0115</td>
<td>The simulation environment support roads operators in deteriming the effects of traffic management and control strategies.</td>
</tr>
<tr>
<td>ECOM-RQ-SP5-0121</td>
<td>Simulation models should include control data (junction control, signal plans, VMS and ramp metering locations).</td>
</tr>
<tr>
<td>ECOM-RQ-SP5-0123</td>
<td>Information should be exchanged between vehicles/drivers and infrastructure like this is done in reality.</td>
</tr>
<tr>
<td>ECOM-RQ-SP5-0124</td>
<td>Exchange of information is feasible between simulation environment and applications and components.</td>
</tr>
<tr>
<td>ECOM-RQ-SP5-0126</td>
<td>Link up to an emission model.</td>
</tr>
</tbody>
</table>

Table 5: Requirements on the Micro Simulation Adapter
3.3 **Overview of the MS-Adapter**

Largely based on the architecture of Paul presented during the last meetings. The following figure shows an overview of the micro simulation adapter and how it is integrated within the SP5 architecture.

![Diagram of MS-Adapter](image)

**Figure 1: System overview with ‘NetState’ and ‘NetState Converter’**.

The ‘NetState’ component delivers routing decisions that are forwarded through the ‘VISSIM COM Adapter’ to VISSIM by means of the VISSIM COM-interface. The ‘NetState Converter’ is executed only once in order to generate the ‘NET’ and ‘INP (NetState)’ files.

3.4 **VISSIM Interfaces**

While it offers a wide range of built-in functionality, VISSIM provides in addition different Application Programming Interfaces (API), file interfaces and dedicated modules for the users to automate their workflow and customize their VISSIM models with their own applications. This is of particular relevance in the course of the simulation of new technologies such as cooperative systems, where the interfaces described in the following may be used.

3.4.1 **VISSIM Com Interface**

Occasionally projects will require extensive pre- or post-processing, numerous scenarios to be investigated or the application of customized control algorithms.

For these cases VISSIM can be run from within other applications serving as a toolbox for transportation planning algorithms. Access to model data and simulations is provided through a COM interface, which allows VISSIM to work as an Automation Server and to export the objects, methods and properties.
The VISSIM COM interface supports Microsoft Automation, so you can use any of the RAD (Rapid Application Development) tools ranging from scripting languages like Visual Basic Script or Java Script to programming environments like Visual C++ or Visual J++. Scripts can also be started from within VISSIM (internal scripting).

3.4.2 VISSIM Car2X Module

The Car2X Module allows the implementation and tests of applications that use Vehicle-to-Vehicle and/or Vehicle-to-Infrastructure communication.

The algorithms for the Car2X applications can be written as Python script or C++ program.

It can access data of the vehicles which have the respective equipment flag set in the vehicle type and defines when vehicles send messages to nearby vehicles and how vehicles react on messages received from other vehicles.

The wireless communication is modelled in a separate *.DLL, which is part of the Car2X module.

3.4.3 VISSIM Driver Model DLL Interface

The External Driver Model DLL Interface of VISSIM provides the option to replace the internal driving behavior by a fully user-defined behavior for some or all vehicles in a simulation run.

The user-defined algorithm must be implemented in a DLL written in C/C++ which contains specific functions.

During a simulation run, VISSIM calls the DLL code for each affected vehicle in each simulation time step to determine the behavior of the vehicle. VISSIM passes the current state of the vehicle and its surroundings to the DLL and the DLL computes the acceleration / deceleration of the vehicle and the lateral behavior (mainly for lane changes) and passes the updated state of the vehicle back to VISSIM.

The external driver model can be activated for each vehicle type separately in the dialog box Vehicle Type by checking the checkbox Use external driver model on the tab page External Driver Model and selecting a driver model DLL file and optionally a parameter file to be used.

3.4.4 External Signal Control DLL Interface

Users can also define external signal controllers using the signal controller *.DLL and signal controller *GUI.DLL (in C or C++ programming language).

External signal control can be activated in the VISSIM Signal Control dialog. For the defined controllers, in each controller time step VISSIM contacts all controller *.DLLs at the end of the current simulation time step. First, the current signalization states and detector data of all SCs are passed to the respective *.DLLs. Second, the *.DLLs are asked to calculate new desired signal states which are subsequently passed back to VISSIM. Depending on parameters set by the controller logic, either these signal states are applied immediately, or transition states (e.g. amber when switching
from green to red) are inserted automatically, as defined in the signal group parameters in VISSIM. In the next simulation time step the vehicles in VISSIM will cope with the new signalization.

3.4.5 ANM file interface

ANM stands for Abstract Network Model and is a high level file interface to define VISSIM models. The main use case is to transfer data from other tools (such as the transport planning model VISUM or traffic engineering tool) to VISSIM. ANM is a XML format. The main advantage is that the abstract network model is based on nodes and edges, which is the natural format for many other, more macroscopic tools in transport planning and management. VISSIM can read ANM in two ways:

- Complete import (initial: creating a new VISSIM network)
- Adaptive import (only differences in the new *.ANM file compared with the *.ANM file imported earlier) in order to update only the affected data in the VISSIM network.

3.5 Components of the MS-Adapter

3.5.1 VISSIM COM Adapter

This component uses the car2x VCOM interface from VISSIM in order to deliver vehicle beaconing data to the RSUs (802.11p) and/or to the center (3G). The coupling between ‘VISSIM’ and ‘VISSIM COM Adapter’ can be established through a configuration entry in the file ‘c2x.ini’. The adapter can serve several clients. In the opposite direction it offers an interface to ‘broadcast’ speed advices from RSUs within the simulation environment (VISSIM).

3.5.2 External FTC (‘extFTC’)

An external fixed time TL controller which controls the signal groups of VISSIM directly according to a given predefined program. The ‘extFTC’ component is able to rotate the fixed-time program (can be invoked remotely by using an interface).

It also forwards the actual state of the TLC (including rest times) and current sensor values through a socket interface. This interface is in line with the interface of the eCoMove component ‘TLC / Sensor Adapter’. In VISSIM the ‘ext. FTC’ component must be defined as part of the TLC parameters.

3.5.3 ecoNetwork Prediction

This sub-system ecoNetwork Prediction (also named “NetState”) contains the modelling of the dynamic route choice and traffic assignment for the road network. It completes the available data in a way that is in line with a given objective function. The results are stored into the ecoMap.
3.5.4  **ecoMap Netstate Adapter**

This sub-system ecoMap Netstate Adapter receives dynamic current and predicted network states from ecoNetwork Prediction online and writes the data into the ecoMap, thereby performing a mapping of VISSIM to ecoMap link ids.

3.5.5  **TLC Adapter**

The component TLC Adapter receives dynamic TLC and detector states from the VISSIM COM Adapter online and writes the data into the ecoMap. It is assumed that a mapping of VISSIM to ecoMap link ids has already been done by the VISSIM COM Adapter.

The connection of eCoMove applications to the TLC goes through a TLC adapter, which in the Peek/Helmond case only connects to the CCOL application to give commands to ImFlow (so effectively the CCOL also acts as an inbetween adapter in this case). The only command that will be given is request of priority to ImFlow, while detector and signal group states can be read from the CCOL. However, in this simulation environment there are multiple ways to get detector states and it will be situation dependen whether the detector and signal group states will actually be read this way.

3.5.6  **NetState Converter**

The NetState Converter reads the ‘INP (VISUM)’ and ‘FMA’ files and generates two new files: ‘INP (NetState)’ and ‘NET (NetState)’.

‘INP (NetState)’ does not contain predefined routing decisions (the ones from VISUM are deleted). Instead it is enriched by new vehicle types (one for each destination), inflows per vehicle type (according o-d-data) and local routing decisions per vehicle type (for each intersection). Later on in online operation the vehicle type dependent inflows and local routing decisions are updated / adjusted by the VISSIM Com Adapter (COM interface).

‘NET (NetState)’ contains all static data needed to run ‘NetState’. This mainly comprises the network topology with traffic related attributes (e.g. link capacities, speed restrictions), origins and destinations, and initial net inflows. The whole configuration is fully compliant with the VISSIM configuration.

3.5.7  **Virtual COM Manager**

It substitutes the real communication unit of a test site. Towards the eCoMove JAVA/OSGi system it offers the agreed eCoMove interfaces 1 and 3 (but not the full range of data, but rather the part that is needed here). But it communicates with the VISSIM COM Manager through socket connection.

The component Virtual COM Manager receives vehicle state data from the VISSIM COM Adapter, converts it to a CAM message format and sends the according CAM messages via Ethernet (not wireless LAN) to the COM Manager.
3.5.8 **Subchapter on Peek TLC**

The Peek TLC will run in the simulation environment just like it runs on the street. The controllers in the Netherlands are mostly based on a standard called CCOL and this is also the case for Helmond. On top of CCOL the new traffic light control strategy ImFlow can take over signal phasing decisions, while CCOL keeps track of safety constraints.

Both CCOL and ImFlow can be compiled into stand-alone executables suitable for simulation. The CCOL application can connect to Vissim through a Peek proprietary DDL file. ImFlow and CCOL, connect to each other the same way as on real traffic controllers in the field, but some manual tuning of port numbers may be necessary because you can simulate multiple controllers at once on one pc and then all of those CCOL – ImFlow connections should use different port numbers.

3.5.9 **Subchapter on COMManager**

The COMManager is developed within SP2. It is responsible for the communications between ITS stations. An appropriate API for application and facility development was defined for portable use in an OSGi run-time environment. This API will be used to communicate between the “ITS simulation station” – here describes as simulation environment – and virtual roadside ITS stations respectively virtual center ITS stations.

Detailed information can be found in deliverable [D2.3].

3.5.10 **Subchapter on Map Feeder (NAVTEQ)**

The Map Feeder is developed within SP2. It is responsible for writing vehicle CAM and DENM messages provided by the COMManager into the ecoMap. The Map Feeder implements a net matching functionality in order to reference vehicle information onto ecoMap road elements.
4 Test side Modelling

4.1 Helmond
The city of Helmond is situated in the south of the Netherlands and has been serving many projects for their tests and pilots. The map below shows the locations where roadside units are installed.

![Roadside units in Helmond](image)

Figure 2: roadside units in Helmond

For the main roads in Helmond a microscopic simulation model was generated including detailed modelling of traffic light controlled intersections considering cycle paths as well as pedestrian crossings and the corresponding traffic demand. A traffic demand model is available for the morning and evening peak periods with traffic volumes for every 15 minutes. The following figure shows the modelled network:

![VISSIM model for Helmond](image)

Figure 3: VISSIM model for Helmond
The following figure shows an example of a complex intersection with grey road elements, footpaths in orange and cycle paths in light red; traffic lights are represented using dark red lines.

![VISSIM model of a complex intersection](image)

**Figure 4: VISSIM model of a complex intersection**

### 4.2 Munich

Description of modelled test site including special issues. Which networks are being used and what is particular about them?

Two traffic models are built for the test site Munich:

- a macroscopic traffic model where the focus is on the network and demand modelling
- a microscopic traffic model derived from the macroscopic model with focus on detailed modelling of intersections and vehicle-infrastructure interaction

The following map shows the city of Munich, where two main areas can be identified:

- Area 1: 59.2 km, motorway.
- Area 2: 247.3 km, urban major road.
- Area 3: 116.1 km, urban minor road.
Figure 5: Munich

For this area a macroscopic model including hourly traffic demand for a typical work day and a microscopic simulation model was generated and calibrated using in total six months of historic loop data.

Based on the macroscopic model a microscopic model was derived and improved by detailed modelling of traffic light controlled intersections including real signal plans. The following figure shows an example.

Figure 6: Example of Signal Plan
The following figure shows the modelled network:

Figure 7: VISUM model for Munich

The following figure shows an example of a complex intersection with grey road elements and traffic lights represented by using dark coloured lines.

Figure 8: 3D model of an intersection

4.3 French Motorway
Description of modelled test site including special issues. Which networks are being used and what is particular about them?
The French test site is a motorway only test site and covers a stretch of approximately 30 km of the A10 between Sorigny and Monnaie (from south to north) passing the city of tours (see next figure).
For this area a microscopic traffic model was built using three different demand scenarios for an average working day:

- one demand model covering the morning peak
- one demand model covering the evening peak
- one demand model covering off peak period (noon)

The three demand models were calibrated using aggregated, historical loop data for this part of the network.

![Figure 9: Tours](image)

Besides on and off ramps, the model includes a toll barrier which will be used for the verification and validation of the ecoTolling application. The following figure shows an example for an on and off ramp situation.
Figure 10: VISSIM ramp

As all other models the VISSIM model for the French motorway includes information as lane restrictions, no passing zones and speed limits. The following figure shows the speed limits in front of the toll gate.

Figure 11: VISSIM up stream speed limits and Sorigny toll gate
5 Installation Instructions for setting up the simulation environment

5.1 Introduction
This chapter describes the installation process for the ecoMove Simulation Environment. The purpose of the Simulation Environment is to test and verify ecoMove components inside a traffic simulation.

Here, at first, the prerequisites will be listed. Later on, there are more detailed instructions for each step of the actual installation.

5.2 Prerequisites
1. Install Java Runtime
2. Install Visual Studio C++ (Express) 2008
3. Install VISSIM 5.3.10 (32 Bit) with C2X extensions
4. Register VISSIM COM Server
5. Have ecoMap (ADASRP directory from Navteq) available
6. Have Knopferfish 3.10 available

5.3 General
Create a directory for the subsequent installation steps, e.g.
C:\ecoMove\Simulation

This directory will later on be referred to as the simulation directory.

5.4 ecoMap
5.4.1 Installation
- Copy ADASRP (full version or eCoMove-only version) to eCoMap_NAVTEQ in the simulation directory
- Copy NAVTEQ execution environment's folder to NAVTEQExecutionEnvironment in the simulation directory
- Install the Knopflerfish 3.1.0 OSGi framework
- From the “NAVTEQExecutionEnvironment” folder, copy the following files into the ADASRP binary folder (eCoMap_NAVTEQ\ADASRP.2011\bin\Win32\ADASRP):
  a. all .dll files
  b. props.xargs (just a copy of the Knopflerfish configuration file)
  c. ADASRP.eCoMove.ini
  d. ADASRP.eCoMove.bat
- Copy the folder SP2_bundles to your local directory
5.4.2 Installation

1. Create or copy `init.xargs` and `restart.xargs` (as described in the chapter below) to `eCoMap_NAVTEQ\ADASRP.2011\bin\Win32\ADASRP`
2. Edit the following sections in `ADASRP.eCoMove.ini` (Absolute paths are preferable):
   a. Path to the map database
   b. Path to Knopflerfish `framework.jar` installation
   c. Path to Knopflerfish `jars` folder
   d. Check if there is an environment variable `JAVA_HOME` set on your computer. If not, define it (it should point to the root directory of your Java installation)
      - or -
   e. edit `ADASRP.eCoMove.bat` accordingly (add `SET JAVA_HOME=C:\WhateverYourPathIs` before the `SET PATH` command)

5.4.3 Start

Start ADASRP by just running the `ADASRP.eCoMove.bat`. Both the ADASRP and the Knopflerfish GUI should appear.

5.4.4 Remark

Refer to `NAVTEQ Execution Environment.doc` for more details.

5.5 Knopflerfish

5.5.1 Configuration

5.5.1.1 Method 1 (recommended):

If there are no such files in `eCoMap_NAVTEQ\ADASRP.2011\bin\Win32\ADASRP` folder or it does not contain any settings of your own, you can copy them from the manual\examples folder.

All the paths in two xargs files must be updated according to real locations on local machine!

5.5.1.2 Method 2:

The other option is get a copy of the default knopflerfish's init.xargs file, and update as followed.

Use `-install` and `-start` to install and start the following bundles (in additional to default bundles):

- Knopflerfish bundle
  - Measurement
• Bundles provided by NAVTEQ (can be found in NAVTEQExecutionEnvironment folder)
  o ASASRPFrameworkManager
  o DynamicMapDisplay
  o ADASRPRouteProvider
  o ADASRPPositionClient
  o ecoMapImplNAVTEQ

• Bundles downloaded from repository
  o Dataprodviders_core
  o LDT_admin
  o eu.ecomoveproject.ldt.util
  o ecoMap
  o CamFeeder
  o qfree_poma_api
  o ecomessage_api
  o communication_api
  o ecomessage_qfree
  o ans1RT-unigone

• Virtual COM-Manager

In the init.xargs file, paths to the bundles must be edited according to their real locations on local machine.
The following image shows all the bundles (except for default Knopflerfish's) and their dependencies. Green bundles are provided by NAVTEQ. Blue bundles are downloaded from repository in form of jar files. Red bundles are additional Knopflerfish bundles.
Figure 12: All JAVA bundles (except for default Knopflerfish's) and their dependencies.

In init.xargs, we need to configure ecoMessage:

- Dorg.cvisproject.calm.emulator.broadcastport=6061
- Dcom.qfree.ecomove.ecomessages.loopback=true
- org.cvisproject.calm.emulator.broadcastaddress=127.0.0.1
- Dcom.qfree.ecomove.ecomessages.cam.dstaddress.0=datagram://[2001:11:777:44::55]:5000
- Dcom.qfree.ecomove.ecomessages.verbose=false
- Dcom.qfree.ecomove.ecomessages.cam.port=5000
- Dorg.cvisproject.calm.emulator.broadcastinterval=5000
- Dorg.cvisproject.calm.emulator.enable=true
- Dorg.cvisproject.service.calm.io.serviceIDLookupTable=file:///C:/Mat_USER/Quan/ecomove/DEV/sid-table.txt
- Dorg.osgi.framework.bootdelegation=sun.*,com.sun.*
- Dcom.qfree.ecomove.ecomessages.validate=false

The most important configuration is:

- Dcom.qfree.ecomove.ecomessages.loopback=true

It tells ecomessage_qfree to send messages back to itself. After returning to ecomessage_qfree, the messages will be forward to CamFeeder.

Other java packages should also be imported:

- org.knopflerfish.framework.system.packages.base=javax.net,
- javax.net.ssl,
- sun.net.www.protocol.http,
- sun.net.www.http,
- javax.swing,
- javax.swing.tree,
- javax.swing.table,
- javax.swing.plaf.metal,
- javax.swing.plaf.basic,
- javax.swing.plaf,
- javax.swing.filechooser,
- javax.swing.event,
- javax.swing.border,
- javax.accessibility
5.5.2 restart.xargs

In the restart.xargs file, we simply write down all the configuration mentioned above, and configure the restart folder:

```
-Dorg.osgi.framework.dir=[simulation
directory]\eCoMap_NAVTEQ\ADASRP-
eCoMove\ADASRP.2011\bin\Win32\ADASRP/fwdir
```

5.6 Vissim

5.6.1 Remarks

Always start Vissim 5.3.10 using the parameter --automation. This will allow simultaneous operation of VCom and COM:

```
vissim.exe --automation
```

Alternatively, you can replace the path of your VISSIM installation in all the files called `start_vissim_c2x.bat` and use these to start VISSIM.

5.6.2 Start

Open an ecoMove map in Vissim. Currently there are

- Helmond
- Dortmund
- Munich (not up-to-date)

At least one VehicleType in your Vissim Map must contain EQUIPMENT C2XVEH

5.6.3 Operation

- Verify that the simulation speed is set to 1.0 (i.e. real-time)!

Always close VissimComAdapter (command line window) after each simulation run. Vissim does not do this automatically.
6 Verification of eCoMove Application and Components

This chapter describes the results from the verification tests. As development and verification is on-going at the time of submission this chapter could not be completed. To give an impression of what to expect the verification results of one component have been included. Test results for the other applications and components will follow in the final version of this document in November 2012.

6.1 Verification overview for ecoNetwork Prediction (MAT)

Based on various static network attributes, dynamic capacity related information, road side sensor data and - above all - vehicle generated data (positions, speed, routes), the component ecoNetwork Prediction estimates the current, future and ideal/desired traffic state for the road network in terms of traffic flows, travel times, link emission values and o-d-route distribution schemes.

Within the simulation environment the verification procedure firstly consists in comparing estimated / determined model values with corresponding values from the micro simulation VISSIM that has not been provided to the ecoNetwork Prediction model before. When optimisation is involved (determination of desired traffic states) it will be examined whether the optimised VISSIM scenario gives rise to an overall less fuel consumption / emission for the whole traffic network.

The main functional blocks of the component ecoNetwork Prediction that are to be verified in the simulation environment are the following:

- Estimation of current and future traffic states
  Reference to main requirements: SP5.11.68, SP5.11.69
  Reference to test sites: Simulation environm., Helmond, Munich
  Reference to verification phases: alpha, beta
  Reference to test cases (below): 2

- Estimation of ideal / desired traffic states
  Reference to main requirements: SP5.11.67, SP5.11.70
  Reference to test sites: Simulation environment, Munich
  Reference to verification phases: alpha, beta
  Reference to test cases (below): 3

Note: Verification test 1, defined in chapter 5.10 of deliverable D5.6, is omitted here as it only refers to real test sites.

References of the verification tests for the component ecoNetwork Prediction to other chapters and deliverables:

- The functional description of the component can be found in chapter 3.10 of deliverable D5.6.
- The description of interfaces, parameters and test setups can be found in chapter 4.10 and 5.10 of deliverable D5.6, and chapter 6.10 of this deliverable.
6.1.1 Verification Test 2: Estimation of current and future traffic states

Reference to main requirements: SP5.11.67, SP5.11.70
Reference to verification phases: alpha, beta

6.1.1.1 Evaluation of verification Test Case 2

6.1.2 Verification Test 3: Estimation of ideal / desired traffic states

Reference to main requirements: SP5.11.68, SP5.11.69
Reference to verification phases: alpha, beta

Besides the estimation of the current and future network states, the ecoNetwork Prediction component can optimise the route choice of the traffic in the road network by using fuel consumption or emission related objective function. The routing schemes of the network are expressed in form of local routing decisions (per intersection approach and destination) rather than global route definitions (from origins to destinations). All those local routing decisions are also defined in the micro simulation VISSIM, where they completely define the route choice of the traffic in the simulation.

During the simulation run the adjusted local routing decisions are periodically sent from ecoNetwork Prediction to the component VISSIM COM Adapter that is capable to change the corresponding VISSIM routing decisions via COM interface.

Figure 13: The interplay of VISSIM, VISSIM COM Adapter and ecoNetwork Prediction. State data goes from VISSIM to ecoNetwork Prediction, local routing decisions in the opposite direction.

The verification test consists in running a micro simulation while adjusting / optimising the local routing decisions of the VISSIM scenario. Thereby, all VISSIM vehicle trajectories are logged into a separate file “*.fzp”. When the simulation run is finished, the fzp-file is imported into the Enviver tool to compute the overall emission values of the traffic scenario over simulation duration.
The VISSIM verification test network (see Figure 9 below) is a very small one with 12 links, one source, two destinations and no traffic lights. All links have the same unique capacity.

The o-d-matrix contains the following fixed values:

<table>
<thead>
<tr>
<th>Source 1</th>
<th>Destination 1</th>
<th>Destination 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>900 veh/h</td>
<td>900 veh/h</td>
</tr>
</tbody>
</table>

All VISSIM links have 1 lane. At the beginning of the links 2, 3 and 4 there are speed decisions (40 km/h) that reduce the capacity of all links in the network with the exception of link 1:

- Capacity Link 1 = 1800 veh/h
- Capacity Link $x = 1650$ veh/h, $x \neq 1$

From this it is clear that not the complete network inflow can be routed along the shortest routes to the two destinations.

There are two routing decisions for the two destinations that equally distribute the approaching traffic along the destination relevant exit links (see Figure 9 below):

- $RD_1 = \{ \text{left}=0.5, \text{straight}=0.5 \}$
- $RD_2 = \{ \text{left}=1.0, \text{right}=0.0 \}$
- $RD_3 = \{ \text{right}=0.5, \text{straight}=0.5 \}$
- $RD_4 = \{ \text{left}=0.0, \text{right}=1.0 \}$

Furthermore, there are three detectors defined as can be seen in the figure below.
D5.7 Development and Testing of Traffic Management and Control Measures [DRAFT]

Figure 14: The VISSIM test network with 4 local routing decisions that belong to the two different destinations. The green lines are speed decisions (40 km/h) that reduce the capacity of all links in

6.1.2.1 Evaluation of verification Test Case 3

In Figure 10 below it can be seen how ecoNetwork Prediction optimizes the origin-destination-routes for the test example in order to realize minimal fuel consumption / emission. The shortest connection in the middle (see Figure 10: orange link that serves both destinations) is maximal utilized. The longest route to destination 2 on the other hand is not utilized at all. The traffic flow to destination 1 is split in a way that reflects the restricted capacity of the middle link.

![Figure 15: The ecoNetwork Prediction model shows in its 3D visualization how the destination sub-flows are routed through the network: left=applied o-d-routes to the destination 1, right=applied o-d-routes to the destination 2.](image)

All vehicle trajectories for both scenarios of VISSIM (normal and optimised) have been logged into a VISSIM evaluation fzp-file. The external tool «Enviver» has been applied in order to compute the emission values of these simulation runs.

The two routing decisions RD_1 and RD_3 has been optimised. The new values at the end of the simulation run in VISSIM are:

\[
RD_1 = \{ \text{left} = 0.189641982316971, \text{straight} = 0.810357987880707 \} \\
RD_3 = \{ \text{right} = 0.0, \text{straight} = 1.0 \}
\]

Emission results for a simulation run of 1800 seconds without optimised routing decisions (computed by means of «Enviver.exe» from VISSIM-fzp-file):

<table>
<thead>
<tr>
<th>CO2</th>
<th>NOx</th>
<th>PM10</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.551e+04 g</td>
<td>69.21 g</td>
<td>8.308 g</td>
</tr>
<tr>
<td>7.104e+04 g/h</td>
<td>138.5 g/h</td>
<td>16.62 g/h</td>
</tr>
<tr>
<td>180.3 g/km</td>
<td>0.3515 g/km</td>
<td>0.04219 g/km</td>
</tr>
</tbody>
</table>

Emission results for a simulation run of 1800 seconds with optimised routing decisions (computed by means of «Enviver.exe» from VISSIM-fzp-file):
CO2 | NOx  | PM10  |
---|------|------|
2.676e+04 g | 54.89 g | 6.887 g |
5.354e+04 g/h | 109.8 g/h | 13.78 g/h |
151.6 g/km | 0.311 g/km | 0.03901 g/km |

Reference to main requirements:

- SP5.11.68: The direct comparison of the emission results shows that the optimised routing entails reduced emission values. As can easily be seen for the simple example the optimised routing decisions of the model represent the minimal total fuel consumption of the network.

- SP5.11.69: The traffic flow to destination 1 is split in a way that reflects the restricted capacity of the middle link (see Figure 10: orange link). Thus, the optimised solution takes into account the restricted capacity of this link.

Summary of the verification test:

- The state data (vehicle, detector) is transferred from VISSIM to ecoNetwork Prediction online.
- The local routing decisions determined by ecoNetwork Prediction are used to change the local routing decisions of VISSIM online.
- The referenced requirements are met.

6.2 Verification overview for ecoRouteAdvice (TUM)
See above…

6.3 Verification overview for ecoParkAdvice (TECHNOLUTION)
See above…

6.4 Verification overview for ecoGreen Wave (MAT)
See above…

6.5 Verification overview for ecoBalanced Priority (PEEK)
See above…

6.6 Verification overview for ecoRamp Metering (VIALIS)
See above…

6.7 Verification overview for ecoSpeed and Headway Management (TNO)
See above…

6.8 Verification overview for ecoTruck Parking (ASFA)
See above…

6.9 Verification overview for ecoTolling (ASFA)
See above…
6.10 Verification overview for ecoGreen Wave (MAT)
See above…

6.11 Verification overview for ecoApproach Advice (TUM)
See above…

6.12 Verification overview for ecoEmission Estimation (TNO)
See above…

6.13 Verification overview for ecoTraffic Strategies (PEEK)
See above…

6.14 Verification overview for ecoMaps (NAVTEQ)
See above…

6.15 Summary of Verification tests
Have all requirements been met?
7 Appendix

Place huge figures, source code and tables in this section