Grant Agreement Number: 723016

Project acronym: **INFRAMIX**

Project full title: INFRAMIX - Road INFRAstructure ready for MIXed vehicle traffic flows

D.2.2

ARCHITECTURE AND INTERFACE SPECIFICATION OF THE CO-SIMULATION ENVIRONMENT

Due delivery date: 28/02/2018
Actual delivery date: 21/03/2018
Organization name of lead participant for this deliverable: **FOK**

Project co-funded by the European Commission within Horizon 2020		
Dissemination level		
PU	Public	X
PP	Restricted to other programme participants	
RE	Restricted to a group specified by the consortium	
CO	Confidential, only for members of the consortium	

Project funded by the European Union's Horizon 2020 Research and Innovation Programme (2014 – 2020)

Document Control Sheet

Deliverable number:	2.2
Deliverable responsible:	FOK
Work package:	2
Editor:	Robert Protzmann

Author(s) – in alphabetical order		
Name	Organisation	E-mail
Karl Hübner	FOK	karl.huebner@fokus.fraunhofer.de
Maximilian Neubauer	FOK	maximilian.neubauer@fokus.fraunhofer.de
Robert Protzmann	FOK	robert.protzmann@fokus.fraunhofer.de
Severin Sander	FOK	severin.sander@fokus.fraunhofer.de
Evdokia Papanikolaou	ICCS	evdokia.papanikolaou@iccs.gr
Ioannis Papamichail	TUC	ipapa@dssl.tuc.gr
Andreas Kerschbaumer	VIF	Andreas.Kerschbaumer@v2c2.at
Marlies Mischinger	VIF	Marlies.Mischinger@v2c2.at
Stefan Thonhofer	VIF	Stefan.Thonhofer@v2c2.at

Document Revision History			
Version	Date	Modifications Introduced	
0.1	05.12.2017	Initial draft and table of contents	FOK
0.2	11.01.2018	Updated chapters 2.2 and 3.2	FOK, TUC
0.3	12.01.2018	Provided structure for chapter 2	FOK
0.4	17.01.2018	Included contents of VIF for chapter 3.2	VIF, FOK
0.5	22.01.2018	Prepared chapter 5, clarified contents of chapter 3.1	FOK
0.6	24.01.2018	Prepared chapters 2,3, and 4 for review	FOK, VIF
0.7	25.01.2018	Delivered contents for chapter 5	FOK
0.8	30.01.2018	Delivered contents for chapter 5	VIF
1.0	07.02.2018	D2.2 ready for review by SIE	FOK, VIF, ICCS
1.1	28.02.2018	Updates according to Reviews by BMW, SIE, and TUC	FOK, VIF
1.2	05.03.2018	Minor updates addressing issues by BMW and ATE	FOK
2.0	15.03.2018	Final Document approved by AAE, ASF, ATE, BMW, ENIDE, ICCS, and VIF	FOK
3.0	19.03.2018	Final Document approved by all partners	FOK

Legal Disclaimer

Abbreviations and Acronyms

Acronym	Definition
5G	5th Generation Mobile Networks
ACC	Adaptive Cruise Control
AV	Automated Vehicle
AWC	Adverse Weather Conditions
BTN	Bottleneck
CAM	Cooperative Awareness Message
CCV	Connected Conventional Vehicle
C-ITS	Cooperative Intelligent Transport Systems
CV	Conventional Vehicle
DEMN	Decentralized Environmental Notification Message
DLA	Dynamic Lane Assignment
DLL	Dynamic-link library
DPR	Dynamic Penetration Rate of automated vehicles
EC	European Commission
ECU	Electronic Control Unit
ETSI	European Telecommunications Standards Institute
EU	European Union
GA	Grant Agreement
GNSS	Global Navigation Satellite System
HLA	High Level Architecture
HIL	Hardware in the Loop
ICOS	Independent Co-Simulation
IMC	INFRAMIX Management Centre
IP	IP Connectivity
ITS	Intelligent Transport Systems
ITS G5	WLAN based ad hoc communication standard
IVIM	Infrastructure to Vehicle Information Message
KPI	Key Performance Indicator
LCA	Lane change advice
LCAFC	Lane Change Advice Flow Control
LTE	Long Term Evolution
MAPEM	MAP (topology) Extended Message
MTFC	Mainstream Traffic Flow Control
NLD	New Lane Design
OBU	OnBoard Unit
ODR	OpenDrive
OEM	Original Equipment Manufacturer
PO	Project Officer
RSU	RoadSide Unit
RWZ	Roadworks Zone
SAE	Society of Automotive Engineers
SLC	Single Lane Closure
SUMO	Simulation of Urban Mobility (Traffic simulation tool)
TMC	Traffic Management Centre
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-X (X represents any entity capable of receiving C-ITS communications)

VMS	Variable Message Signs
VSimRTI	Vehicle-2-X Simulation Runtime Infrastructure
WGS84	World Geodetic System 1984
WLAN	Wireless Local Area Network
WP	Work Package
XFCD	Extended Floating Car Data
XML	Extensible Markup Language

Table of Contents

List of Figures

List of Tables

Executive Summary

INFRAMIX EU project aims to prepare the road infrastructure to support the coexistence of conventional and automated vehicles, targeting to the transition period when the number of automated vehicles will gradually increase. A “hybrid” road infrastructure will be the project outcome after defining the necessary upgrades and adaptations of the current road infrastructure as well as designing and testing novel physical and digital elements. In order to ensure an uninterrupted, predictable, safe and efficient traffic, novel technologies are designed, diverse components are incorporated and different technologies are combined, making the definition of the requirements for such a system, a complicated rather crucial step to the concept design and to the project workflow. An indispensable part of this process is the status quo analysis that set the baseline of the technological level.

This deliverable is created in the starting phase for INFRAMIX and presents the INFRAMIX co-simulation environment to be used validating the proposed use cases before entering the field test. The main goal is to simulate realistic traffic on the test sites, the detailed behaviour of either automated vehicles or vehicles with human drivers, and the communication among vehicles, infrastructure, and traffic management services. Furthermore, the presented simulation framework is prepared to work in hybrid testing scenarios, where real vehicles and real road situations interact with the co-simulation and vice versa.

In chapter 1, the scope of the co-simulation environment and the purpose this document is shown. Chapter 2 describes the simulation methodology and the simulation frameworks used to achieve this task. In chapter 3 an overview of the requirements on the simulation environment is given, which mainly derive from deliverable D2.1 [1], which served as an input document for this task. Chapter 4 gives a detailed list of all simulation models to cover all required aspects. Additionally, all input and output parameters, that each simulation model needs or produces, are given. Finally, in chapter 5 the architecture of the INFRAMIX co-simulation environment is presented, as well as challenges to be solved during this task.

1. Introduction

1.1 Aim of the project

The INFRAMIX project aims to prepare the road infrastructure to support the transition period and the coexistence of conventional and automated vehicles. Its main objective is to design, upgrade, adapt, and test both physical and digital elements of the road infrastructure. The key outcome will be a “hybrid” road infrastructure able to handle the transition period and become the basis for future automated transport systems. Towards this objective different technologies are deployed; mature simulation tools adapted to the peculiarities of automated vehicles, new methods for traffic flow modelling, traffic estimation and traffic control algorithms. The project outcomes will be assessed via simulation and in real stretches of advanced highways.

INFRAMIX builds on three traffic scenarios: *dynamic lane assignment, roadwork zones* and *bottlenecks*. Highways are mainly addressed, as they are expected to be the initial hosts of mixed traffic. However, the key results can also be transferred to urban roads. In order to assess the mixture of traffic and the proposed technologies by both virtual and hybrid testing, a co-simulation environment is required, which is presented in this document.

1.2 Purpose of the document

The purpose of this document is to provide an overview about the simulation tools and simulation models used for the INFRAMIX co-simulation environment. Based on the outcomes of task T2.1 and the requirements catalogue of deliverable D2.1 [1], the requirements on the simulation environment are collected and presented in this document. Furthermore, this document aims to find the input and output parameters of all simulation models in order to link them properly to one co-simulation environment, which is capable to cover all aspects of the INFRAMIX use cases. Eventually, the co-simulation environment is constructed of the two co-simulation frameworks VSimRTI and ICOS. While VSimRTI covers microscopic aspects of the simulation, such as vehicle traffic, communication, road infrastructure, and services, the simulation framework ICOS models the sub-microscopic behaviour of automated vehicles and vehicles with human drivers. Therefore, this document also presents how those simulation frameworks are coupled with each other to accomplish this task.

2. Simulation Methodology

The goal of the INFRAMIX simulation environment is to assess the proposed scenarios by simulating each different use case on different level of detail. The level of detail, that is sufficient to properly validate each of the use cases, will be investigated during the project. However, the proposed solution of a co-simulation should be able to cover all of these.

The proposed INFRAMIX simulation environment consists mainly of two parts. Firstly, the simulation framework VSimRTI covers the simulation of microscopic traffic, communication, applications, and road infrastructure. Secondly, the simulation framework ICOS models the behaviour of both human drivers and autonomous vehicles in detail. Both simulation frameworks are coupled with each other. Therefore, the INFRAMIX simulation environment is able to cover microscopic, as well as sub-microscopic aspects of the simulation. Thanks to this setup, all proposed use cases can be simulated. Furthermore, the simulation framework is prepared to work for hybrid testing, that is, integrating sensor information from real road infrastructure in real time, and, feed real vehicles with information from the simulation. However, precise requirements on the simulation setup for hybrid testing will be defined in the frame of Task 2.3 aiming to prepare the simulation environment for the hybrid testing. The following chapters describe the simulation environment in detail, mainly focusing on both simulation frameworks, VSimRTI, and ICOS.

Figure 1 gives an overview of the INFRAMIX simulation environment. The INFRAMIX Management Centre (IMC), which is depicted in Figure 1, refers to an advanced version of a traffic management centre (TMC) performing improved functionalities especially for the proposed use cases. However, the IMC itself is not modelled in the simulation framework presented in this document.

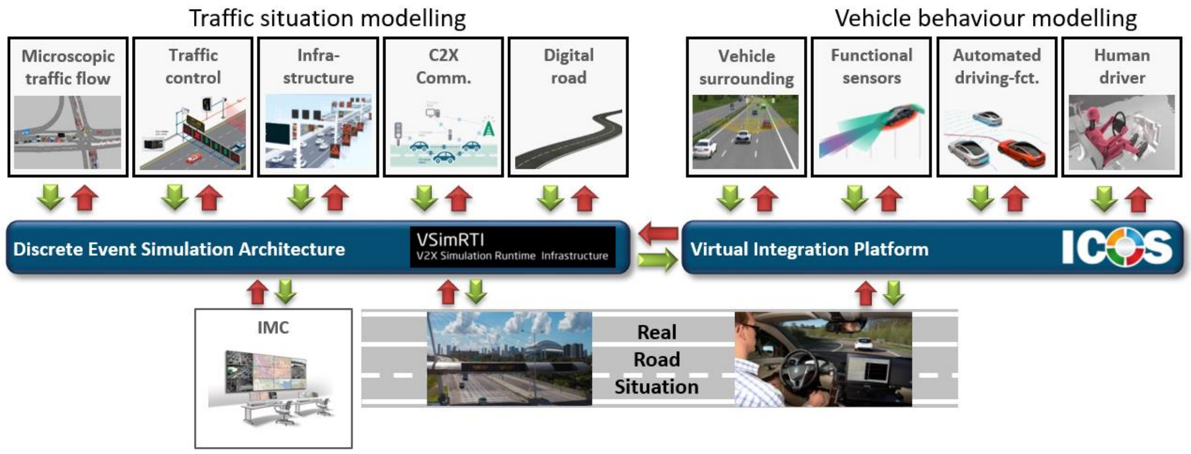

Figure 1 – INFRAMIX - Simulation framework

2.1 VSimRTI

The V2X Simulation Runtime Infrastructure (VSimRTI) is a comprehensive framework for the assessment of new solutions for Cooperative Intelligent Transportation Systems. Vehicle movements and sophisticated communication technologies like Vehicle-2-X communication and cellular networks can be modelled in detail. VSimRTI couples different simulators to allow the simulation of the various aspects of future Intelligent Transportation Systems [2].

In the following sections, the basic concepts of the simulation infrastructure of VSimRTI as well as the supported simulation domains are described.

2.2.1 Ambassador Concept

In contrast to existing fixed simulator couplings, the VSimRTI simulation infrastructure allows the easy integration and exchange of simulators. Thus, the high flexibility of VSimRTI enables the coupling of the most appropriate simulators for a realistic presentation of vehicular traffic, emissions, wireless communication (cellular and ad-hoc), user behaviour, and the modelling of mobility applications. Depending on the specific requirements of a simulation scenario, the most relevant simulators can be used.

VSimRTI uses an ambassador concept inspired by some fundamental concepts of the High Level Architecture (HLA). Thus, it is possible to couple arbitrary simulation systems with a remote control interface. Furthermore, the runtime infrastructure manages everything that is related to timing, exchange of messages between the simulators instances, and the deployment of each simulator (see Figure 2).

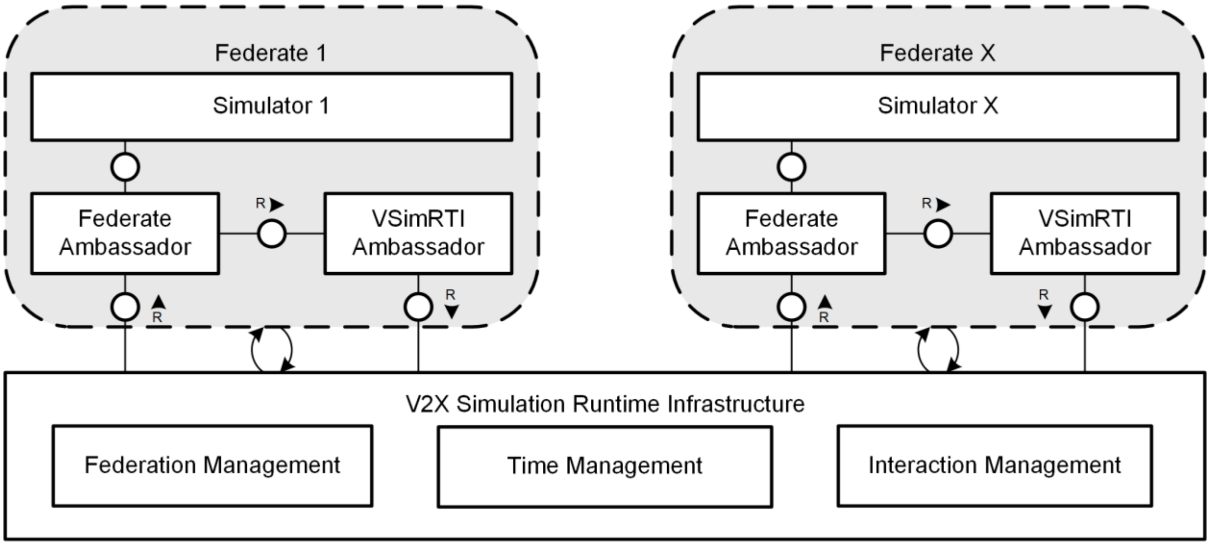

Figure 2 – Basic concept of simulator coupling with VSimRTI

- **Federation Management:** to control the lifecycle of all connected simulators and ICOS. More precisely, to start, join, and stop the individual simulators in the federation during the configured time frame of the simulation.
- **Time Management:** to synchronize the time between the independent processes of the simulation federates. In a joint procedure with the federates, VSimRTI performs the time advance grants for local events that could either coming from the simulator itself or produced by another regulated simulator in the federation.
- **Interaction Management:** to coordinate the information exchange between all simulators. All simulators communicate messages only with VSimRTI, which then distributes the messages to all other federates that have subscribed to the according information.

2.2.2 Simulation Domains

In this section the simulation domains, which are relevant for the INFRAMIX project, are described briefly. Existing simulator coupling covers some of these domains, and only adaptions need to be made for the scenarios of INFRAMIX. For others, new simulators have to be implemented.

Vehicle Applications (ADAS functions): With vehicle functions, the behaviour of single vehicles can be modelled. This can happen on various levels. On microscopic level vehicles would, for example, react on sensors and V2X messages, and initiate lane or speed changes. On sub-microscopic level, an interaction of the application could include the calculation of detailed trajectories. With VSimRTI_App, an application simulator is given which provides functionalities on both levels, however, the specific vehicle functions need to be developed during the INFRAMIX project.

Microscopic Traffic: Provides realistic mobility pattern for a multitude of vehicles and generates the overall traffic flow in the investigated scenarios. Each vehicle in the simulation is modelled as an individual agent using simplified car-following models. All simulated vehicles in the Traffic Simulator, such as SUMO [3], have a corresponding agent in the application simulator and could be equipped with models to provide the dedicated functionality of e.g. behaviour of conventional vehicles or communication capabilities of connected vehicles. Furthermore, this approach allows the modelling of different vehicle types such as trucks and passenger cars.

Traffic Control Strategies: Are, in real world, implemented in a solution from a Traffic Service Provider, Vehicle Services of dedicated OEMs, or the INFRAMIX Management Centre (IMC) integrated at an Infrastructure Provider / Road Operator as mainly investigated in INFRAMIX. In the context of VSimRTI, those strategies have to be modelled as well in the VSimRTI_App. For example, traffic control algorithms and applications of the real IMC can be found in the simulation as well (in the so-called TMC model). Therefore, interfaces have to be implemented to include the solutions provided by the INFRAMIX partners.

Ad-hoc Communication: V2X technologies are usually based on WLAN based ad hoc communication (ETSI ITS G5), which can be simulated with advanced communication simulators, such as OMNeT++ or ns-3.

Cellular Communication: Next to ad hoc communication techniques, cellular communication is used to exchange message among vehicles and other participants such as road infrastructure or even web services. For the simulation of such communication (LTE or even 5G) the VSimRTI_Cell simulator can be used [2].

Infrastructure (Sensors, VMS): Includes two different functionalities. Firstly, it senses information on the traffic pattern from the Traffic Simulator, using sensor models such as counters or induction loops. Secondly, it informs vehicles via Variable Message Signs. This domain requires the development of a new simulator, which accomplishes these tasks.

2.2 ICOS

In today's development processes, the simulation of dynamic systems allows predictions and concept decisions related to the final product to be made at an early stage. This not only involves the modelling, simulation and testing of individual structural components or modules, but also requires the interplay of a large number of functions (with simulation models and also hardware components from various domains) to build up the full system (right up to the full vehicle).

The term "co-simulation" refers to a simulation approach where the components of modern mechatronic systems are interconnected in a suitable way. Therefore the complex interactions between these sub-systems from different development areas are taken into account. The coupling of existing (specific) simulation programs (and the models implemented therein) from different areas of expertise, represents a promising approach for the simulation of the overall system.

ICOS takes into account the complex interactions in a suitable and correct way. The platform enables the precise co-working of different simulation tools (right up to the real-time-capable "Hardware-in-the-Loop" systems) via a so-called "co-simulation framework". Only the verified interaction of numerous models (and therefore also simulated components) enables a realistic virtual concept design and validation of the overall system consisting of vehicle, driver and environment.

2.1.1 Coupling methods

One important challenge in the development of co-simulation is the coupling of models with strong differing dynamic behaviour (time constants). In this case, extremely small time steps must be set for the data exchange between the models. Otherwise, the additional quantization caused by the co-simulation can lead to errors, which can force the co-simulation to diverge. This means that incorrect results are generated, which might even go unnoticed.

ICOS is based on new and highly innovative coupling algorithms to successfully overcome this problem. Methods such as "Waveform Relaxation", multi-rate approaches, adaptive time step control or so-called energy-preserving coupling methods guarantee the correct interaction of models, even with large step sizes (see Figure 3). With ICOS, it is possible to achieve a very high accuracy, a good convergence behaviour and a short calculation time compared to conventional methods.

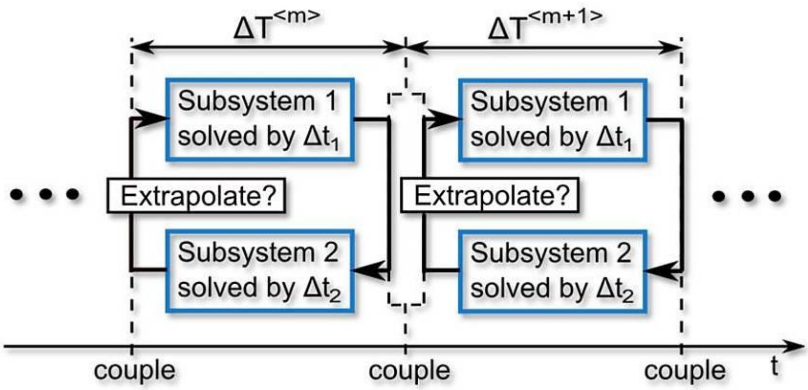

Figure 3 – Coupling method of the co-simulation ICOS

A co-simulation framework has to interconnect the various simulation tools. Additionally, numerous tasks have to be fulfilled:

- communication setup;
- control of the co-simulation procedure;
- synchronization data exchange; and
- step size control and extrapolation strategies.

A further focus of research is the development of a decentralized, fully self-organizing network (framework) for the interconnection of the simulation tools. This framework handles the mentioned tasks in an autonomous and correct way.

2.2.2 Real-time-co-simulation

Nowadays, co-simulation methods typically are limited to offline, non-real-time applications which is a strong limitation regarding re-use of existing simulation models. In the case of so-called "hard-real-time" simulation, all participating simulation models have to deliver their specific calculation results within pre-defined time slots. If this rule is violated, an error occurs and the real time system has to be stopped. The "Hardware-in-the-loop systems"

(HiL) which are used in the modern development process belong to this group e.g. component or ECU test-rigs (see Figure 4).

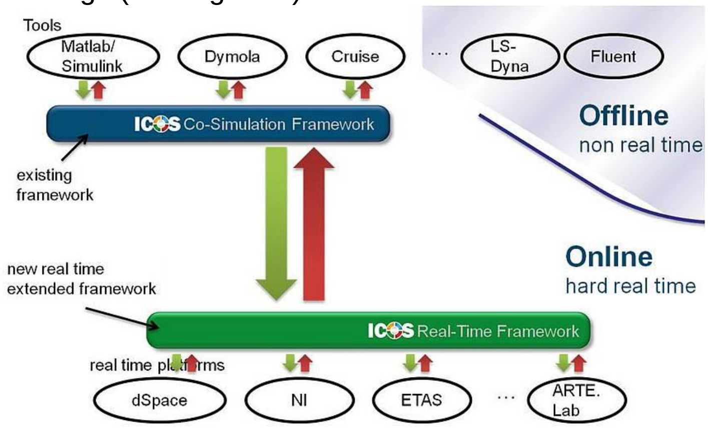

Figure 4 – Principle of the real-time-co-simulation

2.2.3 Fields of application for co-simulation

Traditional co-simulation platforms are usually limited to one domain, such as the calculation of the thermal management system of the simulation or the communication architecture with a heterogeneous set of tools. Along with this confinement, such frameworks can usually only treat problems with a limited dynamic range; i.e. the temporal behaviour of the various models is similar, in fact often very slow. To overcome this problem ICOS includes methodologies to handle different time constants within a system, so that different engineering domains can be easily integrated.

Using this platform a variety of mechatronic problems, for e.g. out of integrated safety, alternative drive trains and chassis control as well as active vibration compensation and safety-critical communication can be efficiently developed, analysed and validated. Moreover, not only the vehicle alone stands in the focal point; even vehicle environment as well as the driver's behaviour can also be taken into account (see Figure 5).

Figure 5 – Interdisciplinary modelling of the vehicle, including driving behaviour and environment

A key feature of ICOS is the support of a vast amount of simulation tools. In INFRAMIX tools are coupled which are needed for a sub-microscopic view on the vehicles behaviour in its environment. These models are described in chapter 3.2.

3. Simulation Requirements

The Co-simulation environment needs to address various requirements, which come up with the proposed use cases for the three INFRAMIX scenarios Dynamic Lane Assignment, Roadworks zones, and Bottlenecks. This section aims to collect all requirements the use cases raise regarding each simulation domain. The following sections are divided into microscopic and sub-microscopic simulation requirements, since VSimRTI and ICOS cover those different levels of the co-simulation environment separately.

3.1 Microscopic Requirements

This section gives an overview of the requirements the microscopic simulation models have. These requirements are derived from deliverable D2.1.

3.1.1 Map

Various components of the co-simulation require a digital map of the test site in different levels of detail. From the microscopic point of view, the concept envisions an initially static map, which must fulfil the following requirements:

- road layout (reference nodes, connections, restrictions)
- locations of infrastructure elements
    - road sensors
    - traffic signs (including VMS)
- locations of RSUs

In the use cases, which include wireless distribution of information about road work zones, a dynamic map update is required. Such update would contain the updated road layout, information about closed and open lanes, and information about lane marking.
The positions of road elements need to be provided in WGS84 coordinates, or must be able to be projected into WGS84 coordinate system.

3.1.2 Vehicle traffic

In general, three types of vehicles have to be considered, according to their role in the proposed use cases (see deliverable D2.1):

- **Connected Conventional Vehicle (CCV):** Vehicle with SAE level of automation equal to 0, 1 or 2, that communicates through wireless messages in real world with the IMC and in the simulation with the TMC-model (through cellular or ITS-G5 communication).
- **Automated Vehicle (AV):** Vehicle with SAE level of automation equal to 3, 4, or 5 that communicates through wireless messages in real world with the IMC and in the simulation with the TMC-model (via cellular or ITS-G5 communication).
- **Conventional Vehicle (CV):** Vehicle with SAE level of automation equal to 0, 1, or 2 which does not communicate with the IMC or TMC-model. Furthermore, automated vehicles, which do not communicate with the IMC or TMC-model regularly, are considered conventional, as their level of automation is not recognised.

Furthermore, both autonomous vehicles and vehicles with human drivers usually behave differently in real traffic. Different levels of automation and different OEMs mean that autonomous vehicles show different behaviour. Also, human drivers vary in their behaviour (e.g. aggressive vs. passive drivers). This needs to be considered in the traffic simulation as well by providing a possibility to parametrize the individual vehicles accordingly (e.g. the duration of lane changes, time headway, or the minimum gap towards the front vehicle). However, the specific parametrization of the vehicles will be elaborated in tasks 2.3 and 2.4 of this working package.

Additionally, for the calibration of the traffic simulation the following requirements are given:

- Traffic flow in number of vehicles per time base, per lane at various locations across the test site.
- Set of parameters describing the vehicle's properties in the simulation, such as width, mass, mean acceleration, maximum speed, and others.

3.1.3 Infrastructure

The simulation models two purposes of the road infrastructure and accordingly two directions of interaction to the TMC model:

1) Road sensors collect traffic information and provide them to the TMC model
2) VMS inform all vehicles with control data from the TMC model (for speed or lane advisory etc.)

RSUs, which could be counted to road infrastructure, however with the purpose of communicating messages to vehicles, are regarded by the Communication domain of simulation.

3.1.4 Communication

Simulation regards for the two different communication paths ITS G5 and cellular communication with detailed models that include all needed aspects such as packet errors or transmission delays. Communication paths over the wired backend (e.g. internal TMC communication or communication to vehicle services) are not modelled in this detail with abstract models. The main communicating entities will be:

- Vehicles
- TMC model (in connection with RSUs)
- Traffic Server, which hosts vehicle services

The different communication paths which need to modelled by the simulation can be found in Table 1.

Table 1 – Communication paths to be modelled in the simulation

	SENDER	RECEIVER	CONTENT
CELL-ULAR	Traffic Server	Vehicle	Traffic information Individualized traffic control
	Vehicle	Traffic Server	XFCD
ITS-G5	TMC model (via RSUs)	Vehicle	DENM IVIM MAPEM
	Vehicle	TMC model (via RSUs)	CAM (RSUs aggregate CAM information)

Concluding, the following message types need to be modelled accordingly:

CAM: Cooperative Awareness Message

- Transmitted periodically with controlled frequency
- Figure 6 shows a schematic of the structure of CAM according to ETSI [4].

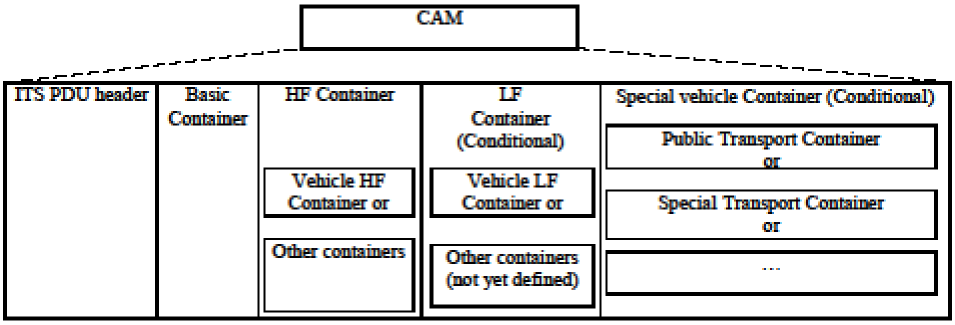

Figure 6 – General structure of CAM (ETSI EN 302 637-2 V1.3.2, [4])

DENM: Decentralized Environmental Notification Message

- Transmission is triggered by certain events in order to notify other vehicles
- Figure 7 gives the schematic of the structure of DENM according to ETSI [5].

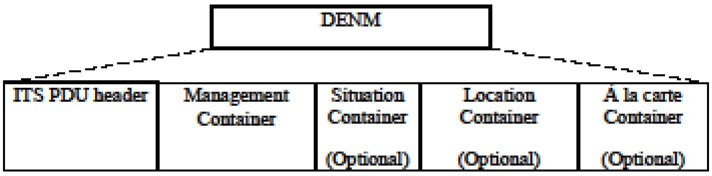

Figure 7 – General structure of DENM (ETSI EN 302 637-3 V1.2.1, [5])

IVI: Infrastructure to Vehicle Information Message

- Is sent event-based by the TMC model
- Informs about static or variable road signs, virtual signs or road works
- Figure 8 shows the schematic of the structure of IVIM according to ETSI [6].

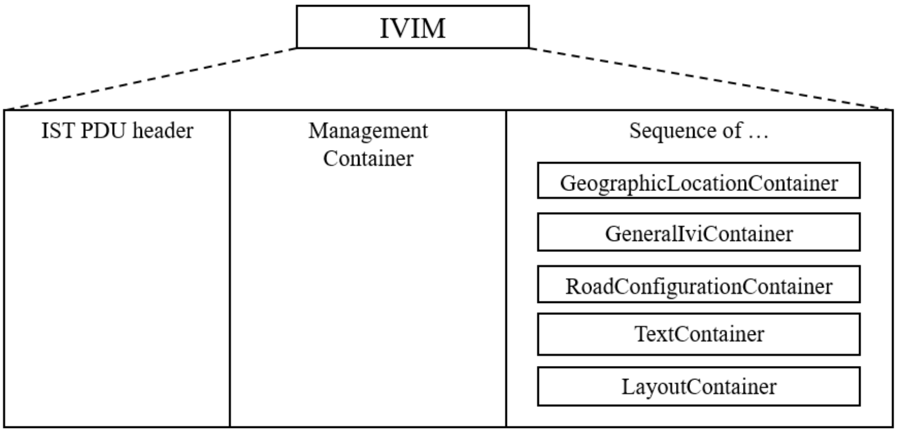

Figure 8 – General structure of IVIM (ETSI TS 103 301 V1.1.1, [6])

MAPEM: MAP (topology) Extended Message

- Consists of road topological details, e.g. changes in lane geometry in road work areas
- Figure 9 gives an overview about the structure of MAPEM according to ETSI [6].

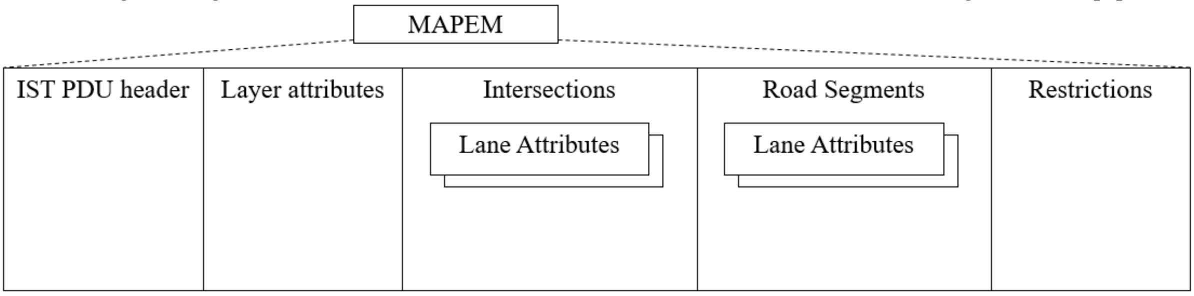

Figure 9 – General (simplified) Structure of MAPEM (ETSI TS 103 301 V1.1.1, [6])

XFCD: Extended Floating Car Data

- Exchange of sensor data, position, road and traffic information for Traffic Servers, connected via cellular communication links

3.1.5 Services/Applications

Traffic services and vehicle applications implement traffic control and traffic estimation strategies. The following requirements come up to the co-simulation environment, in order to integrate each of the service components into the simulation.

Vehicles

- Collect data (e.g. position, velocity, etc.) from vehicles that are equipped with communication services
- Provide data to other entities
  - IMC model (via the RSUs), using C-ITS G5
  - Traffic Server (vehicle services), using cellular communication
- Receive advices from control algorithms regarding, (e.g. lane changes and speeds either from RSUs and TMC model or from vehicle services)

TMC model

- Measure traffic occurrence with road sensor model during the simulation (e.g. traffic flow per lane)
- Controls Variable Message Signs to inform all vehicles
- Model bidirectional data exchange with RSUs, which in turn exchange data (according to the ETSI message definitions) with the connected vehicles
- Model the data exchange with vehicle services

Traffic Server

- Receive xFCD from cellular connected vehicles
- Provides aggregated traffic information gathered from xFCD floating car data and traffic estimations from the TMC model

3.2 Sub-microscopic Requirements

The sub-microscopic simulation consists of different models, which are developed in different simulation environments, whereby each model is responsible for a certain task. For the coupling of these models the co-simulation platform ICOS is used. ICOS enables a simultaneous simulation of all models and manages the synchronisation of the inputs and outputs between the models. Figure 10 shows a diagram, which presents the interaction, the signal flow and the data exchange between all models within the sub-microscopic simulation. The red box depicts the interface between ICOS and VSimRTI. It receives relevant data from VSimRTI and transfers the information to the dynamic data and the sensor model. The *Driving Function* box depicts the driver's category. It´s possible to select between the Human Driver and the Automated Driver.
This simulation allows a detailed investigation of one certain vehicle in the overall simulation (microscopic- & sub-microscopic simulation together). It means that we can assess the vehicle dynamics behaviour in different cases of an SAE 4 level driving function or a human driver. Of particular importance are dynamic investigations during a lane change at bottlenecks or roadwork zones. A detailed description of all models is given in the following sections.

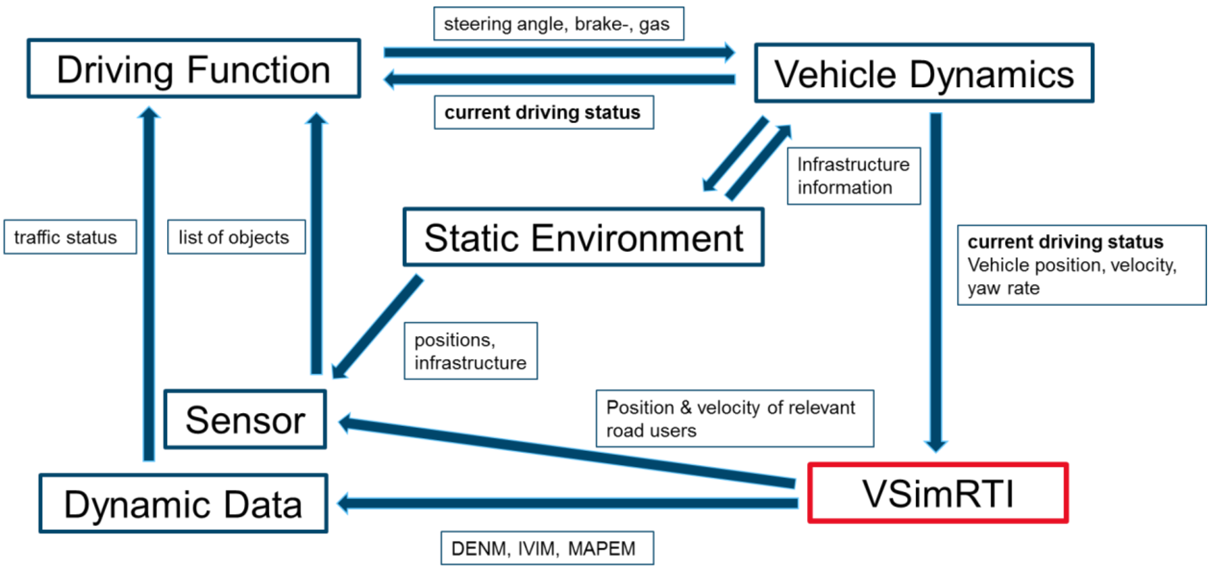

Figure 10 – Sub-microscopic Models and Parameter exchange

3.2.1 Static Environment

The main challenge for the static environment model is to provide relevant information to the ego-vehicle regarding the road infrastructure. This means a highly accurate course of all lanes on the current road section and positions of the traffic signs. All this necessary information is included in the HD-Map, which builds the basis of that model.
Furthermore, the HD-Map should include a detailed description of lanes by attributes. Examples are the type of the lane marking (solid, broken), the colour of the lane marking and the lane type (driving, hard shoulder). Even traffic signs have to be described in detail by attributes. Examples are the type of the sign (signs giving orders, warning signs) or its alignment. The static environment has to filter all relevant information from the HD-Map and provides it to the vehicle sensor.

3.2.2 Human Driver

The human driver model is used for detailed investigations of conventional vehicles and connected conventional vehicles. To use this model in the sub-microscopic simulation the driving function block in Figure 10 can be replaced by the human driver model.
The psychologically based driver model represents driving styles of human drivers. The model captures how a real human steers, accelerates, or brakes and how different environmental and task characteristics influence the performance of steering, braking, driving distraction, fatigue, lane changes. Individual driver style characteristics such as aggressive or safe driving are represented.

3.2.3 Automated Driver

The automated driving functions represent the highway pilot (SAE Level 4). Depending on the detected environment, the driving strategies and the information coming from digital infrastructure the automated driver is planning its path and sends a request of acceleration, deceleration and steering to the vehicle (vehicle dynamics model).

3.2.4 Vehicle Sensors

Vehicle sensor models are used for ongoing monitoring of the vehicles environment. The sensor provides continuously updated information regarding the traffic situation to the automated driving function so that the manoeuvre planning can be adapted to the current traffic situation. Configuration parameters of the vehicle sensor are the range and azimuth angle.

3.2.5 Vehicle Dynamics

Modelling a representative vehicle dynamics allows a simulation of the realistic vehicle movement (cars and trucks). Depending on the vehicle control the model performs human initiated or automated driving manoeuvres. The behaviour of acceleration and deceleration, but also the behaviour when driving along curves can be performed within real physical limits of the vehicle (vehicle performance, grip, driver needs). The key task of the vehicle dynamics model is a detailed description of long- and lateral control of the vehicle.

3.2.6 Dynamic Data

This model receives messages from the microscopic simulation which are relevant for the sub-microscopic simulation. Relevant messages are DENM, MAPEM and IVIM. This model splits up the messages and assesses the relevance of the message content. Depending on that assessment the content of the messages is remitted to the driving function.

4. Model Description

The simulation models are separated into two main areas/categories: The *Microscopic traffic Simulation* enables a total view on scenarios including realistic vehicle behaviour, the V*ehicle behaviour simulation* allows a detailed view on the vehicles interaction with its environment (including traffic and infrastructure). Relevant simulation models have to be defined.

4.1 Microscopic Traffic Simulation

Simulation model		MICROSCOPIC TRAFFIC FLOW		
Tool / Version		SUMO 0.32.0		
Model usage / Function		Depicts dynamic information of traffic flow. Uses simple car following models e.g. Krauss		
Parametrization		Digital map in SUMO specific XML format. Route and trips for each vehicle via either XML or socket interface.		
Data exchange (Format / Unit)	**Input**	*Vehicle parameter definition*		
		- Acceleration	m/s^2	float
		- Deceleration	m/s^2	float
		- Vehicle length	m	float
		- Maximal speed	m/s	float
		- Minimal gap to vehicle in front	m	float
		- Driver imperfection	%	float
		- Driver's desired (minimum) time headway	s	float
		- Maximal lateral speed during lane changes	m/s	float
		Vehicle traffic definition (for each vehicle)		
		- Route	List of edges	String[]
		- Vehicle type reference		String
		- Time of departure	s	double
	Output	*Traffic flow data of single vehicles*		
		- Position (x,y Coordinates)	[m,m]	int[2]
		- Current edge identification		String
		- Lane number (the rightmost is 0)	{0, 1, 2, …}	int
		- Direction/Heading	degrees (from north clockwise)	float
		- Speed	m/s	float
		- Slope	degrees	float
		- Driven distance	m	float
		- Stopped		bool
		- Route identification		String
		- Acceleration	m/s^2	float
		Traffic flow (edge or lane-based		
		- The beginning of the measured interval	s	float
		- The ending of the measured interval	s	float
		- Edge identification	*EDGE-ID*	String
		- Collected vehicle seconds (sum)	s	float
		- Mean travel time	s	float
		- Mean density	#veh/km	float
		- Mean occupancy	% [0, 100]	float
		- Total waiting time	s	float
		- Mean speed	m/s	float
		- Number of departed vehicles	#veh	int

		- Number of arrived vehicles - Number of entered vehicles Number of left vehicles	#veh #veh #veh	int int int

Simulation model		TRAFFIC STATE ESTIMATION PER LANE		
Tool / Version		C/C++		
Model usage / Function		Estimation of traffic flow, speed and density per section and lane of the network. A Kalman filter estimator is used in order to estimate the traffic state per lane in case there is no measurement that can be used directly		
Parametrization		An ASCII file can be used to specify the geometrical characteristics and other parameter values used by the Kalman filter estimator.		
Data exchange (Format / Unit)	**Input**	- Flow measurements from detectors located at each lane at the entrance of the network and at each lane between pairs of origins (on-/off-ramps) - Speeds reported by connected vehicles per section-lane of the network - Lane changes (lateral flow) reported by connected vehicles per section	- veh/h - km/h - veh/h	- double[] - double[] - double[]
	Output	- Flow - Speed - Density	- veh/h - km/h - veh/km/lane	- double[] - double[] - double[]

Simulation model		TRAFFIC STATE ESTIMATION		
Tool / Version		C/C++		
Model usage / Function		Estimation of traffic flow, speed and density per section of the network. A Kalman filter estimator is used in order to estimate the traffic state in case there is no measurement that can be used directly.		
Parametrization		An ASCII file can be used to specify the geometrical characteristics and other parameter values used by the Kalman filter estimator.		
Data exchange (Format / Unit)	**Input**	- Flow measurements from detectors located at the entrance of the network and at each lane between pairs of origins (on-/off-ramps) - Speeds reported by connected vehicles per section of the network	- veh/h - km/h	- double[] - double[]
	Output	- Flow - Speed - Density	- veh/h - km/h - veh/km	- double[] - double[] - double[]

Simulation model	TRAFFIC MANAGEMENT & CONTROL (DBAS)
Tool / Version	C/C++
Model usage / Function	This model defines the Driving Behaviour Adaptation in Real Time at Sags (DBAS). The control strategy receives real-time measurements (or estimates) of the current traffic conditions and

		suggests to the AV drivers (or imposes directly) an appropriate value for the time-gap parameter and possibly also for the vehicle acceleration.		
Parametrization		An ASCII file can be used to specify the geometrical characteristics and other parameter values used by the controller.		
Data exchange (Format / Unit)	**Input**	- Flow per section of the network - Speed per section of the network	- veh/h - km/h	- double[] - double[]
	Output	- time gap to be applied by AVs per section if it is smaller than the one applied already - acceleration to be applied by AVs per section if it is bigger than the one applied already	- sec - km/h^2	- double[] - double[]

Simulation model		TRAFFIC MANAGEMENT & CONTROL (LCA)		
Tool / Version		C/C++		
Model usage / Function		Lane-Change Advice to connected vehicles at bottlenecks The control strategy is fed with real-time lane-specific information about the prevailing traffic conditions and decides on the necessary lane-changing activities to achieve a pre-specified (possibly traffic-dependent) lane distribution of vehicles while approaching a bottleneck, aiming at increasing the bottleneck capacity.		
Parametrization		An ASCII file can be used to specify the geometrical characteristics and other parameter values used by the controller.		
Data exchange (Format / Unit)	**Input**	- Flow per section-lane at the entrance of the controlled area of the network - Occupancy (or density) per section-lane of the network - Ids for all AVs/CCVs - Position for all AVs/CCVs (section id, lane id and distance from the beginning of the section) for all AVs/CCVs	- veh/h - veh/km	- double[] - double[] - string[] - double[][]
	Output	- lane change advise to be applied by AVs/CCVs per section-lane	0: no advice +1:- change to the right lane -1: change to the left	- int[]

Simulation model		TRAFFIC MANAGEMENT & CONTROL (FC)		
Tool / Version		C/C++		
Model usage		This model applies a controller to calculate a VSL that will lead to the necessary flow control at bottlenecks. The control strategy is fed with real-time occupancy (or density) at the bottleneck and decides on the VSL to be applied on each gantry (or AV) aiming at increasing throughput at the bottleneck		
Parametrization		An ASCII file can be used to specify the geometrical characteristics and other parameter values used by the controller.		
Data exchange (Format / Unit)	**Input**	- Occupancy (or density) at the bottleneck section - Speed per section of the network	- veh/km - km/h	- double - double[]
	Output	- speed limit (VSL) to be displayed at the gantries or to be applied by the AVs per section	- veh/h	- double[]

Simulation model		INFRASTRUCTURE SENSORS		
Tool / Version		VSimRTI/TBD		
Model usage/Function		This model measures the traffic flow and occupancy on given road segments in the traffic simulation.		
Parametrization		List of infrastructure sensors in XML format containing the position and type of each infrastructure sensor		
Data exchange (Format / Unit)	Input	*For each sensor:* - Identifier - Coordinates (WGS84) - Lane number (the rightmost is 0) - Direction - Type (induction loop, camera, etc.) - Length (for induction loops) - Update frequency	 degrees unitless unitless unitless m Hz	 string lat,lon integer enum enum double double
	Output	- Traffic flow - Mean speed - Occupancy - Number of vehicles per measurement	veh/h km/h veh/km #vehicles	double double double integer

Simulation model		INFRASTRUCTURE VMS		
Tool / Version		VSimRTI/TBD		
Model usage / Function		Modelling of variable message signs to change speed limits and lane assignments dynamically during the simulation.		
Parametrization		List of variable message signs in XML format containing the position, lane allocation, and display capabilities for each sign.		
Data exchange (Format / Unit)	Input	*For each VMS:* - Identifier - Coordinates (WGS84) - Lanes (the rightmost is 0) - Direction - Type (gantry, road-side) *During simulation:* - Identifier - Advisory type (speed, lane-assigment) - New Speed limit - Lane - Vehicle type (e.g. CV, AV, trucks) *For each vehicle:* - Position	 degrees unitless unitless unitless m/s lat/lon	 string lat,lon int[] enum enum string enum double int enum double[]
	Output	*For each vehicle:* - Allowed lanes - Restricted lanes - Speed limit to apply	 m/s	 int[] int[] double

Simulation model		CELLULAR V2X-COMMUNICATION		
Tool / Version		VSimRTI_Cell2 / 18.0		
Model usage /Function		Models the message exchange of vehicles via cellular based communication networks.		
Parametrization		Parametrization via config file (cell regions, capacity, delay, packet loss, …)		
Data exchange (Format /	Input	- Sending timestamp - Message length - Message payload	ns #bytes	long long byte[]

Unit)		- Sender's address - Receiver's address - Addressing mode	IP address IP address {Broadcast, Geocast,..}	int[] int[] complex
	Output	- Receiving timestamp - Message length - Message payload - Sender's address - Receiver's address	ns #bytes IP address IP address	long long byte[] int[] int[]

Simulation model	AD-HOC V2X-COMMUNICATION			
Tool / Version	OMNeT++/ns-3			
Model usage /Function	Models the message exchange of vehicles via ad hoc communication.			
Parametrization	Parametrization via config file (baseband frequency, transmission power, receiver sensitivity, …)			
Data exchange (Format / Unit)	**Input**	- Sending timestamp - Message length - Message payload - Sender's address - Receiver's address - Addressing mode	ns #bytes IP address IP address {Broadcast, Geocast,..}	long long byte[] int[] int[] complex
	Output	- Receiving timestamp - Message length - Message payload - Sender's address - Receiver's address	ns #bytes IP address IP address	long long byte[] int[] int[]

Simulation model	V2X COMMUNICATION SERVICE			
Tool / Version	VSimRTI_App / 18.0			
Model usage /Function	Generates ETSI standardized and custom V2X messages triggered by time or specific events.			
Parametrization				
Data exchange (Format / Unit)	**Input**	Simulation timestep CAM: - Position - Vehicle length - Vehicle width - Vehicle class - Vehicle speed - Vehicle heading - Longitudinal acceleration - SAE Level of automation DENM: - Event position - Road id - Event type	ns Latitude/Longitude m m m/s degrees 0.1*m/s^2 Latitude/Longitude {fog, ice, snow, rain, speed, position, direction, curve, obstacle, parkinglot}	long double[] double double String int int int int double[] string

		- Event strength		double
		- Caused speed by event	m/s	double
		MAPEM:		
		- layerType		enum
		- roadSegments		RoadSegment[]
		RoadSegment:		
		- id		int
		- refPoint		double[]
		- laneSet	Latitude/Longitude	GenericLane[]
		GenericLane:		
		- laneID		int
		- laneWidth		int
		- speedLimit		double
		- nodeList	List of points	double[]
		IVI:		
		- validFrom	ns	long
		- validTo	ns	long
		- iviStatus	{new, update,..}	enum
		- iviContainer		*IviContainer[]*
		IviContainer:		
		- iviType	{warning, info,..}	enum
		- minAwarenessTime	sec	int
		- applicableLanes		*Lane[]*
		- direction	{same, opposite, both}	enum
		- iviPurpose	{safety,..}	enum
		- roadSignCodes		*RoadSign[]*
		Lane:		
		- status	{open, closed,..}	enum
		- index	(0=rightmost,…)	int
		- type	{traffic,dedicated,..}	enum
		RoadSign:	Vienna Convention	
		- class		int
		- code		int
		- option		int
		- value		int
		- unit		enum
	Output	*V2X Message*		
		- Sending timestamp	ns	long
		- Message length	#bytes	long
		- Message payload		byte[]
		- Sender's address	IP address	int[]
		- Receiver's address	IP address	int[]
		- Addressing mode	{Broadcast, Geocast,..}	complex

Simulation model	MICROSCOPIC MAP
Tool / Version	VSimRTI scenario-database / 18.0
Model usage / Function	Provides information about the road topology for other simulation models, such as microscopic traffic simulation (SUMO) or map representation in application models.

Parametrization		OpenDrive XML File (ODR)		
Data exchange (Format / Unit)	Output	*Nodes:* - Id - Position - Intersection	 Latitude/Longitude in WGS84	vector string double[] boolean
		Connections: - Id - From - To - Geometry - Lanes - Length - Max speed - Road type	 Node Id Node Id Node Ids #lanes M m/s {motorway, primary, secondary,…}	vector string string string string[] int double double string
		Restrictions - Type - Source - Target - Via	 {Only, Not} Connection Id Connection Id Node Id	vector enum string string string
		Traffic Signs: - Type - Position - Value	 Latitude/Longitude in WGS84	vector enum double[] double
		Sensors: - Type - Position - Connection - Lanes	 {Induction Loop, Camera,…} Latitude/Longitude in WGS84 Connection Id	vector enum double[] string int[]

4.2 Vehicle Behaviour Simulation

Simulation model		STATIC ENVIRONMENT		
Tool / Version		C++		
Model usage / Function		Depicts static information of the environment		
Parametrization		OpenDrive XML File (ODR)		
Data exchange (Format / Unit)	Input	**Ego-vehicle-position** • Cartesian coordinates are in relation to the lanes which are given in the HD map - X-coordinate - Y- Coordinates - Heading (left is positive)	 - m - m - rad	 - float - float - float
	Output	**Lanes** • 8 lanes left to the reference line • 8 lanes right to the reference line • 1 reference line • sum: 17 lanes		

		• Cartesian coordinates are in relation to the lanes which are given in the HD map		
		<u>Cartesian coordinates</u> - X-coordinate (17x) - Y-coordinate (17x)	- m - m	- float [] - float []
		<u>Polar coordinate</u> - Radius (sensor system) (17x) - Angle (sensor system) (17x) - Lane identification number - Road identification number - Road mark - Road mark quality	- m - rad	- float [] - float [] - int [] - int [] - int [] - int []
		Signals		
		<u>Cartesian coordinates</u> - X-coordinate - Y-coordinate	- m - m	- float [] - float []
		- <u>Polar coordinate</u> - Radius (sensor system) - Angle (sensor system)	- m - rad	- float [] - float []
		- Signal identification number - Signal name - Signal dynamic - Signal orientation - Signal type - Signal subtype - Signal value - Signal quality		- float [] - float [] - float [] - float [] - float [] - float [] - float [] - float []

Simulation model		HUMAN DRIVER		
Tool / Version		MATLAB R2013 and Python		
Model usage / Function		- Depicts the driving behaviour of humans.		
Parametrization		- Tbd. (different driver types)		
Data exchange (Format / Unit)	**Input**	**Lanes** • 1 lane left to the vehicle position • 1 lane right to the vehicle position <u>Cartesian coordinates</u> - X-coordinate (2x) - Y-coordinate (2x)	- m - m	- float [] - float []
	Output	- Steer angle request - Driving/Braking torque request - Steering angle - Break (pedal) - Acceleration (gas pedal)	- rad - Nm - Rad - % - %	- float - float - float - double(0...1) - double(0...1)

Simulation model	AUTOMATED DRIVER
Tool / Version	MATLAB R2013

Model usage/ Function		- Adaptive Cruise Control - Lane Keeping Assist - Lane Change Assist and Lane Change Pilot - Connected, Cooperative Driving - Emergency Brake Assist - Lane merging Pilot		
Parametrization		- Driving strategies (tbd)		
Data exchange (Format / Unit)	**Input**	- Position of vehicles and object - Left/right lanes	- m - m	- float [] - float []
	Output	- Steer angle request - Driving/Braking torque request	- rad - Nm	- float - float

Simulation model		VEHICLE SENSOR		
Tool / Version		C++		
Model usage / Function		- Description of the sensor´s field of view - Detection of vehicles and objects in the vehicles environment - Detection of traffic signs and road markings		
Parametrization		- Object list (what the sensors can detect) - Sensors azimuth and range (depending on sensor type) - Visibility of Traffic signs, object and vehicles		
Data exchange (Format / Unit)	**Input**	**Lanes** • 8 lanes left to the reference line • 8 lanes right to the reference line • 1 reference line • sum: 17 lanes Polar coordinate - Radius (sensor system) - Angle (sensor system) - Lane identification number - Road identification number - Road mark - Road mark quality **Signals** Polar coordinate - Radius (sensor system) - Angle (sensor system) - Signal identification number - Signal name - Signal dynamic - Signal orientation - Signal type - Signal subtype - Signal value - Signal quality **Vehicles around the ego vehicle** - Vehicles radius (vehicle ego-system) - Vehicle Angle (vehicle ego-system) - Vehicle Velocity	-m -rad -m -rad -m -rad - m/s	- float [] - float [] - int [] - int [] - int [] - int [] - float [] - float [] - float [] - float [] - float [] - float [] - float [] - float [] - float [] - float [] - float [] - float [] - float []

	Output	**Lanes** • 8 lanes left to the reference line • 8 lanes right to the reference line • 1 reference line • sum: 17 lanes <u>Polar coordinate</u> - Radius (sensor system) - Angle (sensor system) - Lane identification number - Road identification number - Road mark - Road mark quality **Signals** <u>Polar coordinate</u> - Radius (sensor system) - Angle (sensor system) - Signal identification number - Signal name - Signal dynamic - Signal orientation - Signal type - Signal subtype - Signal value - Signal quality	-m -rad -m -rad	- float [] - float [] - int [] - int [] - int [] - int [] - float [] - float [] - float [] - float [] - float [] - float [] - float [] - float [] - float [] - float []

Simulation model		VEHICLE DYNAMICS		
Tool / Version		MATLAB R2013 and/or CarMaker v5.0		
Model usage / Function		- Execution of physically possible longitudinal and lateral vehicle movements - Vehicle control based on driving tasks from human driver - Vehicle control based on automated manoeuvring		
Parametrization		- Vehicle dimensions (length, width, mass, …) - Vehicle performance (acceleration and deceleration capabilities) - Vehicle Handling (lane change capabilities)		
Data exchange (Format / Unit)	**Input**	- Steering angle - Requested drive torque - Requested brake torque - Road grip	- rad - Nm - Nm	- float - float - float - float[]
	Output	- X-Position on road - Y-Position referring to reference lane - Vehicle speed	- m - m - m/s	- float - float - float

Simulation model	DYNAMIC DATA
Tool / Version	MATLAB R2013 or C++
Model usage / Function	All dynamic information is gathered in this model. - Receive all dynamic data - Assessment the data´s relevance

		- Distribution of relevance data within the framework		
Data exchange (Format / Unit)	**Input**	Cartesian coordinates are in relation to the lanes which are given in the HD map		
		DENM Data		
		- Event position	- (x/y) Cartesian coordinates	- double[]
		- Road id		- string
		- Event strength		- double
		- Caused speed by event	- m/s	- double
		- Event type	- {fog, ice, snow, rain, speed, position, direction, curve, obstacle, parkinglot}	
		MAPEM Data		
		- layerType		- enum
		- roadSegments		- RoadSegment[]
		RoadSegment:		
		- id		- int
		- refPoint	- (x/y) Cartesian coordinates	- double[]
		- laneSet		- GenericLane[]
		GenericLane:		
		- laneID		- int
		- laneWidth		- int
		- speedLimit		- double
		- nodeList	- List of points	- double[]
		IVIM Data		
		- validFrom	- ns	- long
		- validTo	- ns	- long
		- iviStatus	- {new, update,..}	- enum
		- iviContainer		- IviContainer[]
		IviContainer:		
		- iviType	- {warning, info,..}	- enum
		- minAwarenessTime	- sec	- int
		- applicableLanes		- Lane[]
		- direction	- {same, opposite, both}	- enum
		- iviPurpose	- {safety,..}	- enum
		- roadSignCodes		- RoadSign[]
		Lane:		
		- status	- {open, closed,..}	- enum
		- index	- (0=rightmost,…)	- int
		- type	- {traffic,dedicated,..}	- enum
		RoadSign:	Vienna Convention	
		- class		- int
		- code		- int
		- option		- int
		- value		- int
		- unit		- enum

	Output	Cartesian coordinates are in relation to the lanes which are given in the HD map **DENM Data** - Event position - Road id - Event strength - Caused speed by event - Event type **MAPEM Data** - layerType - roadSegments *RoadSegment:* - id - refPoint - laneSet *GenericLane:* - laneID - laneWidth - speedLimit - nodeList **IVIM Data** - validFrom - validTo - iviStatus - iviContainer *IviContainer:* - iviType - minAwarenessTime - applicableLanes - direction - iviPurpose - roadSignCodes *Lane:* - status - index - type *RoadSign:* - class - code - option - value - unit	- (x/y) Cartesian coordinates - m/s - {fog, ice, snow, rain, speed, position, direction, curve, obstacle, parkinglot} - Lat/long WGS84 - (x/y) Cartesian coordinates - List of points - ns - ns - {new, update,..} - {warning, info,..} - sec - {same, opposite, both} - {safety,..} - {open, closed,..} - (0=rightmost,…) - {traffic, dedicated,..} Vienna Convention	- double[] - string - double - double - enum - RoadSegment[] - int - double[] - GenericLane[] - int - int - double - double[] - long - long - enum - IviContainer[] - enum - int - Lane[] - enum - enum - RoadSign[] - enum - int - enum - int - int - int - int - enum

5. Simulation Coupling

The co-simulation environment to be used in the INFRAMIX project consists of both simulation frameworks VSimRTI and ICOS. In order to simulate all proposed scenarios and use-cases, both tools are coupled with each other. Thereby, the simulation models introduced in chapter 4 can be used simultaneously. In this context, VSimRTI takes over the role of the microscopic simulation of traffic, the simulation of communication amongst vehicles and road side units, and the application and service modelling. On the other side, ICOS provides the detailed simulation of driving behaviour of both, autonomous vehicles and conventional vehicles with human drivers.

In order to simulate vehicle traffic on highways, VSimRTI uses the microscopic traffic simulator SUMO. Digital maps of the test sites serve as a basis for the simulation. In order to simulate communication between vehicles and the infrastructure, communication simulators are utilized which are already coupled with VSimRTI. Furthermore, traffic control strategies, which would affect the driver's behaviour, will be implemented in the application simulator of VSimRTI. The sub-microscopic simulation of the driving behaviour of one single vehicle in the simulation is provided by ICOS. Thereby it is possible to simulate a realistic reaction on certain events, for example lane-change advices or map updates due to road works. In addition, the vehicle's behaviour would have influences on the overall traffic flow. Therefore, a coupling of both simulation frameworks is necessary, enabling the physical interpreted sub-microscopic vehicle model to react depending on the surrounding vehicles coming from the microscopic traffic simulation. The dynamic behaviour of the sub-microscopic vehicle model is used to parameterize the microscopic traffic flow model. This is established by integrating ICOS into VSimRTI by following its ambassador concept. The following sections describe the architecture of the coupling, the interface specification between ICOS and VSimRTI, and various challenges that have to be solved.

5.1 Architecture

The two co-simulation frameworks are linked together by integrating the sub-microscopic simulation framework ICOS into VSimRTI. To be more specific, ICOS is integrated following the ambassador concept of VSimRTI as described in chapter 2.1. In context of the simulations to be made, ICOS takes care of modelling everything related to the vehicle behaviour, while VSimRTI and its simulation models cover the microscopic aspects, such as traffic, infrastructure, and communication.

The IcosAmbassador provides the link between ICOS and VSimRTI. This ambassador implementation is directly integrated into the VSimRTI co-simulation while executing one instance of ICOS and handling the data exchange between ICOS and VSimRTI. Since VSimRTI is written in Java, the integration of ICOS is implemented using the Java Native Interface API. This way we can include the DLL of ICOS directly. Figure 11 shows the integration of ICOS into the VSimRTI co-simulation.

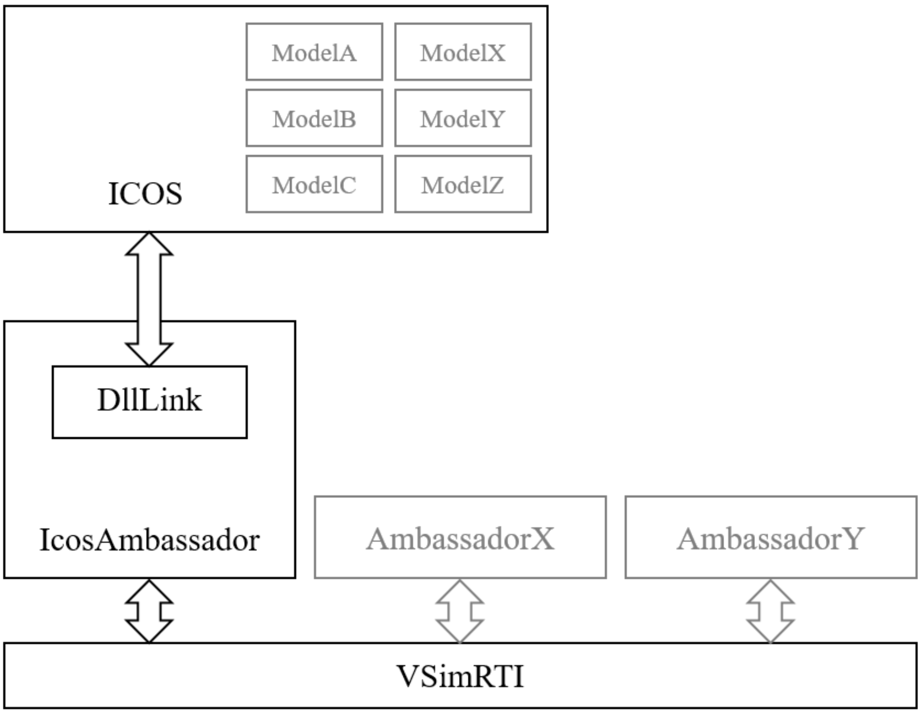

Figure 11 – Integration of ICOS into VSimRTI

The data exchange between ICOS and other simulation tools coupled with VSimRTI is carried out by internal messages being interchanged among the individual ambassadors. For example, the positions and velocities of the individual vehicles in both simulation tools SUMO (microscopic traffic) and ICOS (sub-microscopic vehicle behaviour) must be exchanged between each other. This task is performed via the “VehicleMovements” message, which is produced and consumed by both simulation tools. The VehicleMovements created by ICOS would contain positions only of the vehicle, which is simulated by ICOS (see blue vehicle in Figure 12), whereas the VehicleMovements produced by SUMO would contain all other vehicles from the traffic simulation (see green vehicles in Figure 12).

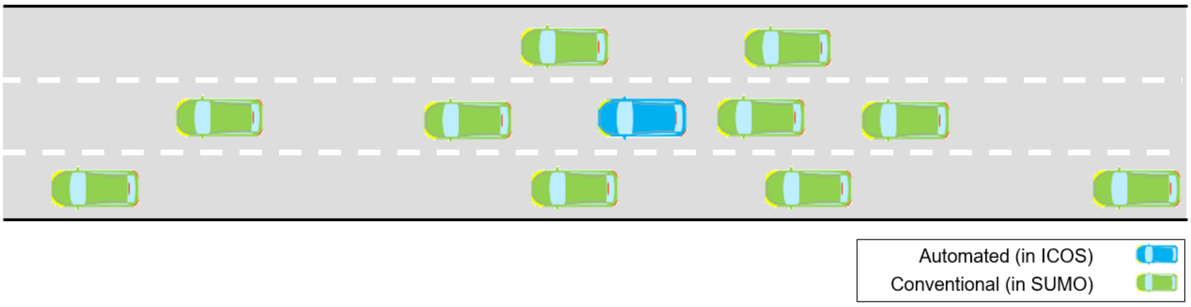

Figure 12 – Simulation of vehicles by different simulation models (SUMO and ICOS)

On the other side, SUMO would consume the VehicleMovements message of ICOS and update the position of the respective vehicle in its traffic simulation, and vice versa. Figure 13 shows the message exchange accordingly.

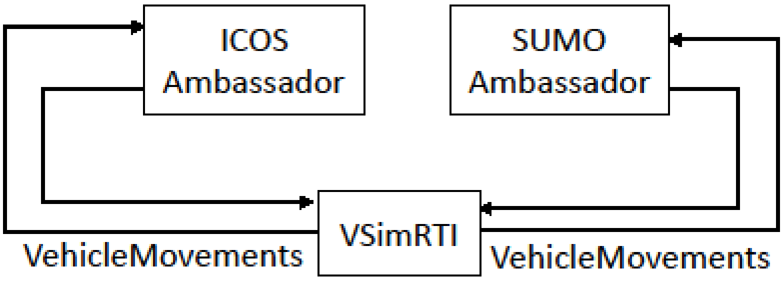

Figure 13 – Exchange of vehicle movements

In the simulation of the proposed use cases of INFRAMIX, vehicles are going to be simulated either microscopically by SUMO, or very detailed by ICOS. In this context, VSimRTI takes care about the assignment of the responsible simulator to each vehicle in the simulation. For this purpose, single vehicles or groups of vehicles can be assigned to the individual simulators. In this particular case, ICOS would handle one specific vehicle, while SUMO would handle all other vehicles. Furthermore, each of the microscopic simulation models proposed in chapter 4.1 are coupled with VSimRTI as well, enabling the full-fledged co-simulation framework to be used in the INFRAMIX project. Figure 14 shows the coupling of the all those simulation models in an exemplary manner.

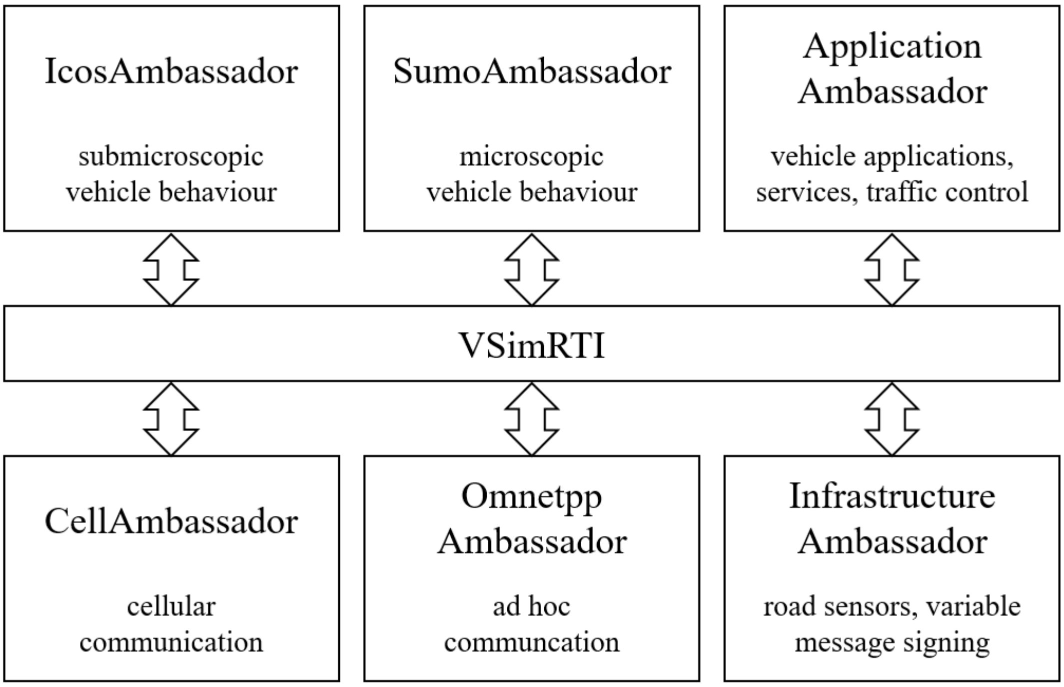

Figure 14 – Coupling of all simulation models to VSimRTI

5.2 Interface Specification

The following tables define the parameters to be exchanged between ICOS and VSimRTI.

VSimRTI to ICOS

Vehicle data for each vehicle in the simulation			
Description	Sends information about the surrounding vehicles in the simulation		
Parameters (Identifier, Unit)	Position	x/y coordinates	double[]
	Heading	rad (left is positive)	double
	Velocity	m/s	double
	Acceleration	m/s²	double
Frequency of data exchange	1-10Hz	For each microscopic simulation step	

Vehicle Action			
Description	Performs certain actions on the vehicle in ICOS		
Parameters (Identifier, Unit)	Allowed speed	m/s	double
	Lane Change Guidance	-1 (change lane to the left) 0 (stay on current lane) +1 (change lane to the right)	int
	Distance gap	m	double
Frequency of data exchange	-	Event based	

V2X message exchange data (DENM)			
Description	Forward a DEN message to the vehicle in ICOS		
Parameters (Identifier, Unit)	Event position	x/y coordinates	double[]
	Road id	ns	string
	Event strength		double
	Caused speed by event	m/s	double
	Event type	fog, ice, snow, rain, speed, position, direction, curve, obstacle, parkinglot	
Frequency of data exchange	-	On occurrence of DEN-message	

V2X message exchange data (IVIM)			
Description	Forward a IVI message to the vehicle in ICOS		
Parameters (Identifier, Unit)	validFrom	ns	long
	validTo	ns	long
	iviStatus	{new, update,..}	enum
	iviContainer		IviContainer[]
	IviContainer		
	iviType	{warning, info,..}	enum
	minAwarenessTime	sec	int
	applicableLanes		Lane[]
	direction	{same, opposite, both}	enum
	iviPurpose	{safety,..}	enum
	roadSignCodes		RoadSign[]

	Lane		
	status	{open, closed,..}	enum
	index	(0=rightmost,…)	int
	type	{traffic,dedicated,..}	enum
	speedLimit	m/s	double
	Road Sign Vienna Convention		
	class		int
	code		int
	option		int
	value		int
	unit		enum
Frequency of data exchange	-	On occurrence of IVI-message	

V2X message exchange data (MAPEM)			
Description	Forward a MAPE message to the vehicle in ICOS		
Parameters (Identifier, Unit)	layerType		enum
	roadSegments		RoadSegment[]
	RoadSegment		
	id		int
	refPoint		double[]
	laneSet		GenericLane[]
	GenericLane		
	laneID		int
	laneWidth		int
	speedLimit		double
	nodeList	List of points	double[]
Frequency of data exchange	-	On occurrence of MAPE-message	

ICOS to VSimRTI

Vehicle data of Ego-Vehicle in ICOS			
Description	Send back information about the vehicle in ICOS to VSimRTI		
Parameters (Identifier, Unit)	Position	x/y coordinates	double[]
	Heading	rad (left is positive)	double
	Velocity	m/s	double
	Acceleration	m/s²	double
	Lane index		int
Frequency of data exchange	100 Hz	For each sub-microscopic simulation step in ICOS	

5.3 Coupling Challenges

This chapter gives an insight into the challenges to be accomplished by the coupling of VSimRTI and ICOS. Those challenges emerge mainly due to the different levels of detail the individual simulation tools work with.

5.3.1 Different Time Resolutions

The coupling of the traffic simulation in SUMO with the vehicle behaviour simulation in ICOS comes with the problem of different timing. More precisely, the traffic simulation in SUMO usually works with simulation steps between 1 second and 100ms (1-10 Hz), whereas the models in the vehicle simulation in ICOS work on a higher resolution (e.g. 100Hz). Due to the different time step sizes, extrapolation techniques are required for the correct data exchange between different simulations. The co-simulation platform ICOS allows the user to select among several extrapolation methods

Figure 15 – Definition of micro and macro step size

Each simulation tool connected to ICOS may use an arbitrary time step size for its internal numerical routines. This solver step size is called micro step size. Only at certain communication points, data is exchanged between ICOS and the simulation tool. The resulting communication step size is called macro step size and can be configured within ICOS. The lowest possible value of the macro step size is the micro step size of the simulation tool. When selecting a suitable macro step size, usually a trade-off between runtime performance (corresponding to a large macro step size) and numerical precision (corresponding to a small macro step size) has to be found. See Figure 15 for an illustration of micro and macro step sizes.

Another important parameter, which has to be configured for the co-simulation, is the scheduling mode of the used simulation tools. ICOS allows either parallel or sequential execution of each simulation step, while in case of sequential the order of the simulation tools can be configured.

The most important use case of co-simulation are the so-called closed-loop problems. In case of sequential execution, extrapolation has to be done for at least one exchanged signal. In case of parallel execution, both signals have to be extrapolated. Depending on the dynamics of the signal, a suitable extrapolation method has to be selected. ICOS supports:

- Zero order hold (ZOH),
- First order hold (FOH), and
- Second order hold (SOH) extrapolation.

Figure 16 shows an illustration of the extrapolation methods. The extrapolation quality strongly depends on the dynamics of the exchanged signal. For instance, the signal in Figure 16 is indeed best extrapolated using ZOH, while FOH and SOH are better suited for strongly oscillating signals.

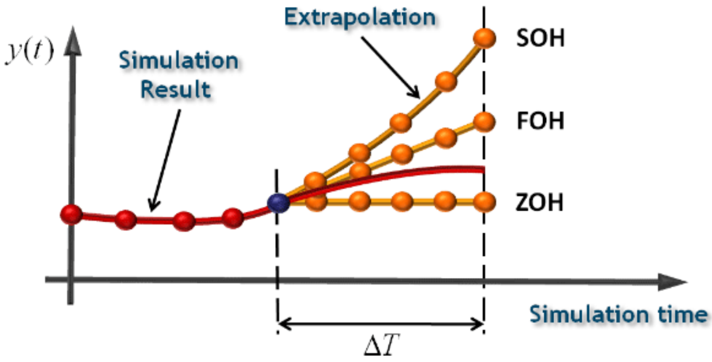

Figure 16 – Illustration of different extrapolation methods.

A macro time step of one second seems to be sufficient for both involved components (VSimRTI including SUMO, and the vehicle model). This can be seen from the first test, as described in section 5.4. However, the macro time step can easily be changed if required in future simulation runs.

In the INFRAMIX simulation framework, we do not expect strong dynamics of any exchanged signals. The simulated vehicle data should change smoothly over the macro time steps (1 s). Hence ZOH extrapolation seems to be sufficient for most of the data, while FOH might be suitable for position data (because the vehicles are always moving). However, some use cases might require other extrapolation techniques. We cannot draw any conclusions about extrapolation quality from the first communication test (see section 5.4) because due to its open-loop topology, no extrapolation is used.

5.3.2 Digital Maps

Digital maps of the test sites must be provided for most of the simulation models in the INFRAMIX simulation environment. For example, the sub-microscopic simulation needs a detailed representation of the road infrastructure. For safe vehicle guidance along a trajectory the availability of an accurate lane description as well the position of traffic signs is indispensable. To assist the driver with lane keeping and lane changing, the driving function needs the course of all lanes in front of the ego-vehicle to calculate reliable trajectories. For longitudinal control the information of speed limits and end of speed limits is required.

On the other side, the microscopic simulations do not require this high level of detail in order to model traffic flows on the test site. Here, basic descriptions of the road geometry including the number of lanes for each road segments is sufficient. However, since the individual simulation tools are coupled with each other and need to exchange precise positions, they should work on the same basis of map. Therefore, the individual simulation tools must be provided with mechanism in order to import the given HD map. In the context of the INFRAMIX simulation environment, we decided to work with the OpenDrive [7] map format. With OpenDrive all given requirements are covered. Additionally, most of the proposed simulation tools already provide import mechanisms for this format (e.g. SUMO and ICOS). Consequently, any given HD map data must be converted into the OpenDrive format at first.

5.3.3 Dynamic Maps

Changes in the course of lanes due to roadworks will be managed by the so called MAPEM message. This message includes the topological definition of all lanes inside the roadwork zone, links between the segments, type of lanes as well restricted lanes. In the simulation those messages will be artificially created, e.g. by the TMC model. Applying those roadworks information dynamically during the simulation, it requires map update techniques in the respective simulation models.

From the microscopic point of view, the traffic simulation is also affected by the change of the road geometry due to roadworks. However, in the microscopic traffic simulation in SUMO different lane widths do not have influences on the traffic flow, only the number of lanes have. As a result, lane drops, lane gains, and lane restrictions are updated in the traffic simulation, whereas changes in the lane geometry, such as narrowing lanes, are ignored.

5.4 Test Simulation Scenarios

In order to test the coupling of ICOS and VSimRTI, a very simple simulation scenario is used. The simulation scenario consists of only one road and two vehicles driving along. Next to ICOS and SUMO, no other simulation tools are involved in the simulation scenario. The simulation scenario is kept as simple as possible in order to keep testing and debugging during the development process as simple as possible. The digital map consists of an artificial road segment provided by an OpenDrive file and serves as the playground for the simulation scenario. The map is imported to VSimRTI, SUMO, and ICOS accordingly. In the configuration of the scenario in VSimRTI, two vehicles are defined: one to be simulated by SUMO, the other one by ICOS. Both vehicles are inserted one after another onto the same road segment. The vehicle simulated by SUMO should therefore follow the vehicle simulated by ICOS (see Figure 17) without crashing into it. Furthermore, the traffic simulator SUMO is configured to work with low-frequent simulation time steps (1 Hz), whereas the solver of the simulation of the vehicle behaviour in Carmaker is triggered in a higher frequency (100 Hz).

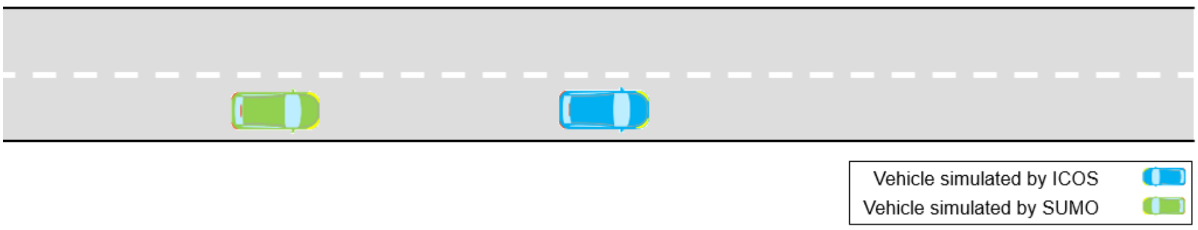

Figure 17 – Simulation scenario used to test the coupling

Figure 18 shows the outcome of this first test. It shows a screenshot from SUMO GUI taken roughly at 200 s simulation time. Both vehicles are driving on the same lane with the same heading. This means that the coupling of ICOS and VSimRTI as described in section 33 works as expected. To conclude, data is successfully exchanged and the built-up co-simulation framework is capable of handling the INFRAMIX simulation scenarios.

Figure 18 – Result of the testing scenario (output of SUMO GUI)

Depending on the outcomes of the discussions to be held in WP4, further test scenarios are going to be implemented in order to improve the coupling between ICOS and VSimRTI.

6. Conclusion

The INFRAMIX co-simulation environment presented in this document will help to assess and validate the proposed techniques and innovative methods in traffic control and management of mixed traffic. All use cases of the three traffic scenarios *dynamic lane assignment, roadwork zones,* and *bottlenecks* will be evaluated utilizing both simulation frameworks VSimRTI and ICOS, which have been successfully linked together. All aspects required to properly simulate the proposed use cases are covered by the INFRAMIX co-simulation environment, from simulation of traffic, communication among vehicles, road infrastructure, and the traffic management centre, through the sub-microscopic simulation of the vehicle behaviour of one vehicle. Next to pure virtual testing, the INFRAMIX co-simulation environment is prepared to work with hybrid testing scenarios, which integrate the given simulation models with vehicles, services, and infrastructure of the real world.

The INFRAMIX co-simulation environment will be developed and extended continuously during the project. Depending on the outcomes of Task 2.3, further test scenarios are going to be implemented which may require extensions on the interface between VSimRTI and ICOS, or even improvements of the integrated simulation models. Furthermore, final decisions on the setup of the simulations and the hybrid testing scenarios will be done in the frame of WP4.

7. References

[1] INFRAMIX, “INFRAMIX - D2.1 - Requirements catalogue from the Status Quo Analysis,” 2018.

[2] R. Protzmann, B. Schünemann and I. Radusch, “Simulation of Convergent Networks for Intelligent Transport Systems with VSimRTI.,” in *Networking Simulation for Intelligent Transportation Systems: High Mobile Wireless Nodes*, John Wiley & Sons, 2017, pp. 1-28.

[3] D. Krajzewicz , J. Erdmann , M. Behrisch and L. Bieker, “Recent Development and Applications of {SUMO - Simulation of Urban MObility},” *International Journal On Advances in Systems and Measurements,* pp. 128-138, 3&4 December 2012.

[4] ETSI EN 302 637-2 (V1.3.2), “Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service”.

[5] ETSI EN 302 637-3 (V1.2.1), “Intelligent Transport Systems (ITS);Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service”.

[6] ETSI TS 103 301 (V1.1.1), “Intelligent Transport Systems (ITS);Vehicular Communications;Basic Set of Applications; Facilities layer protocols and communication requirements for infrastructure services”.

[7] M. Dupus and et al., “OpenDRIVE® Format Specification, Rev. 1.4”.