ARTEMIS	**HoliDes** **Holi**stic Human Factors **Des**ign of Adaptive Cooperative Human-Machine Systems	HoliDes

D9.5 - Modelled and Model-based Analysis of the Automotive AdCoS

Project Number:	332933
Classification:	Public with Public and Confidential Annexes
Work Package(s):	WP9
Milestone:	M3
Document Version:	V1.0
Issue Date:	30.06.2015
Document Timescale:	Project Start Date: October 1, 2013
Start of the Document:	Month 20
Final version due:	Month 21
Deliverable Overview:	**Main document:** D9.5 – Modelled and Model-based Analysis of the Automotive AdCoS – Public **Annex I:** D6.5-7.5-8.5-9.5 Common Annex for Modelled and Model-based Analysis of AdCoS (public) **Annex II:** D9.5 - Feedback and HF-RTP Requirements Definitions Update (confidential)
Compiled by:	Mark Eilers - OFF
Authors:	Mark Eilers - OFF Thierry Bellet - IFS Dominique Gruyer – IFS Stefan Griesche - DLR Svenja Borchers – TWT Denis Martin - TWT Martin Krähling – IAS Fabio Tango – CRF Marco Beccuti – UTO Marco Botta – UTO Elisa Landini – REL Gert Wellert - TAK

Reviewers:	Peter van der Meulen - PHI Carmen Perea – ATO
Technical Approval:	Jens Gärtner, Airbus Group Innovations
Issue Authorisation:	Sebastian Feuerstack, OFF

DISTRIBUTION LIST		
Copy type[1]	Company and Location	Recipient
T	HoliDes Consortium	all HoliDes Partners

[1] Copy types: E=Email, C=Controlled copy (paper), D=electronic copy on Disk or other medium, T=Team site (AjaXplorer)

RECORD OF REVISION		
Date (DD.MM.JJ)	Status Description	Author
30.04.2015	Structure representation with WP9 contents	Mark Eilers
20.05.2015	Incorporated contributions of CRF, REL, TAK, TWT, and OFF	Mark Eilers
28.05.2015	Incorporated contributions of IAS	Martin Krähling
02.06.2015	IFS section on V-HCD & MOVIDA-AdCoS	Thierry Bellet, Dominique Gruyer
08.06.2015	Incorporated contributions of DLR	Stefan Griesche
15.06.2015	Incorporated contributions of TWT	Denis Martin, Svenja Borchers
17.06.2015	Feedback and Conclusion CRF	Fabio Tango
18.06.2015	Compiled public and confidential parts	Mark Eilers
19.06.2015	Incorporated contribution of UTO	Marco Beccuti, Marco Botta
21.06.2015	Compiled general conclusion from partner contributions	Mark Eilers
22.06.2015	Public part ready for review	Mark Eilers
25.06.2015	Review	Peter van der Meulen
25.06.2015	Review	Carmen Perea
29.06.2015	Incorporated review comments	ALL
30.06.2015	Compiled final version	Mark Eilers

Table of Contents

1 List of figures

2 List of tables

3 Glossary

ACC	Adaptive Cruise Control
ADAS	Advanced Driving Aid Systems
AdCoS	Adaptive Cooperative Human-Machine System
ATO	ATOS Spain SA
BAD MoB	Bayesian Autonomous Driver Mixture-of-Behaviors
CAN	Controller Area Network
CM&A	Collision mitigation and Avoidance
CoS	Cooperative Human-Machine System
CRF	Centro Ricerche Fiat
DDDBDVD	Detection of driver distraction based on data on vehicle dynamics
DIR	Driver intention recognition
DLR	Deutsches Zentrum für Luft- und Raumfahrt e.V.
FA	Full Automation
FCW	Forward Collision Warning
HD	Headway
HEE	Human Efficiency Evaluator
HF	Human Factors
HF-RTP	Human Factor Reference Technology Platform
HMI	Human Machine Interface
IAS	Ibeo Automotive Systems GmbH
IFS	IFSTTAR
LCA	Lane-Change Assistant
LKA	Lane Keeping Assistant
MBD	Model Based Design
MDP	Markov Decision Process
MTT	Methods, Techniques, and Tools
NHTSA	National Highway Traffic Safety Administration
OA	Overtaking Assistant
OFF	OFFIS e.V.
PHI	Philips Medical Systems Nederland BV
REL	RE:Lab S.r.l.
RTP	Reference Technology Platform
SL	Supervised learning
SURT	Surrogate Reaction Task
TAK	TAKATA
TTC	Time-To-Collision
TWT	TWT GmbH Science & Innovation
UML	Unified Modelling Language

UTO	Universita Degli Studi Di Torino
WIS	Warning and intervention strategies
WP	Work Package

4 Introduction

Deliverable D9.5 describes the progress and provides feedback concerning the modelling and model-based analysis of Adaptive Cooperative Human-Machine Systems (AdCoS) using the methods, techniques, and (software) tools (MTTs), provided by the Human Factors Reference Technology Platform (HF-RTP) developed in HoliDes. It follows the deliverable D9.4 [11], in which the tailoring of available MTTs to the automotive domain and use-cases has been described. D9.5 consists of a *public* and a *confidential* part. The public part describes the overall modelling efforts and contains the general conclusions about the use of MTTs for AdCoS modelling. The confidential part provides specific feedback concerning the different MTTs that have been used for AdCoS modelling in the automotive domain so far and provides further details and information related to how the AdCoS and HF-RTP will be implemented in the project. The deliverable is furthermore supplemented with a common annex, providing a cross-domain introduction and conclusion to AdCoS modelling.

4.1 Objective of the document

As a reminder, we start with a short reiteration of the definition of MTTs. In our definition, a *method* is a general way to solve a problem. This could e.g. be the use of task analysis to answer a general design question. The term *technique* refers to a concrete instantiation of such a method, i.e., a specific development and analysis techniques, like e.g. model-based or contract-based design of hardware and software. Finally, a *tool* is a technique, which has been realized as either hard- or software. The general purpose of the MTTs provided by the HF-RTP is to support engineers in their development and analysis tasks. Especially in the human factors domain, a lot of techniques are not strictly formalized and not yet supported by specific tools. Results are often stored in descriptive paper form or excel sheets. One of the major goals of the HF-RTP developed in HoliDes is to close the gap between the engineering disciplines which are to a large extent supported by tools and computer aided techniques, and the human factors discipline.

As described in D2.4 [1], HoliDes promotes a *model-based design* (MBD) approach for AdCoS development by defining new and extending existing modelling languages that allow designers to model and analyse isolated aspects of the AdCoS and/or the AdCoS as a whole, including system behaviour, human behaviour, and adaptation. In general, MBD is a method for addressing problems associated with designing complex systems and is based on syntactically and semantically (e.g.

mathematically) defined abstractions of the system, the environment, and the interactions between them. As such, the development process is based upon a system (or in the case of HoliDes, an AdCoS) model, that is constantly evolved from requirement specification, through design, implementation, and testing. When following the MBD approach thoroughly, the AdCoS should be modelled as an executable specification that is continually refined and elaborated throughout the development process [15]. Simulation of the model allows for easier testing and thus improvement of the product quality, while gaining shorter development times at the same time. This is strengthened by the use of automatic code generation, and support for model-based analysis, i.e. verification and validation. In this sense, modelling must be understood as an activity accompanying the whole development process of an AdCoS.

AdCoS development in HoliDes is based on a common development process with phases for requirements, design, implementation, testing, and evaluation. As depicted in D9.4 [11], within WP9, most MTTs are used during the implementation phase. In contrast to other application WPs, where MTTs are primarily directly used by the AdCoS owners during the design phases, WP9 heavily focusses on the implementation and integration of MTTs that intent to provide functionality to the AdCoS itself, i.e. as components of an AdCoS or AdCoS modules, e.g., in terms of adaptive HMIs, assistance functionalities, or as a means for context assessment. As such, much focus was laid on establishing a unified framework enabling a seamless acquisition and exchange of data and exchange and integration of module prototypes across the different modelling partners both within and across the different AdCoS.

This document aims at providing an overview of the models developed and analysis results obtained during the overall development process of the AdCoS in the automotive domain thus far. As such, it focusses on how the tailored MTTs have been used for the modelling and performing model-based analysis of the different AdCoS and AdCoS modules. The remaining document is structured as follows: Section 5 provides an overview of the four AdCoS addressed in WP9 and describes how the provided MTTs have been used modelling the respective AdCoS and (were already available) have been used for model-based analysis. Section 6 provides a general conclusion concerning the use of the tailored MTTs from the automotive perspective.

5 AdCoS modelling

As described in D9.4 [11], WP9 focusses on the development of four distinct AdCoS:

- AdCoS Adapted Assistance
- AdCoS Adapted Automation
- V-HCD Platform for the virtual design of MOVIDA-AdCoS
- AdCoS Adaptive HMI

The following subsections will provide a general overview of these different AdCoS and present the results of the overall modelling process using the tailored MTTs provided by the HF-RTP.

5.1 AdCoS Adapted Assistance

5.1.1 Description of the AdCoS

As described in details in the previous deliverables D9.3 [10] and D9.4 [11], the Centro Ricerche Fiat (CRF) AdCoS is named *Adapted Assistance* and constitutes in a Lane-Change Assistant (LCA) system, which is able to adapt to the internal and external scenarios. This means that the "optimal" manoeuvre is suggested from machine-agent to human-agent, by means of specific warnings, advice and information, according to the visual state[2] and intentions of driver, as well as to the external environment.

5.1.1.1 Operational definition

The LCA system comprehends the following functionalities:

- Lane-Change Assistant (LCA) and Overtaking Assistant (OA)
- Forward Collision Warning (FCW).

Being an Adapted Assistance system, the "trigger" for the adaptation is represented by the cognitive state of the driver (if he/she is distracted or not, which is her/his intention), based on which the strategies of the AdCoS are modified. The Adapted Assistance system is implemented on the CRF test-vehicle (TV), which is a Fiat 500L, with the following sensors installed on-board:

- One external camera to detect the edges of the lanes on the road and the relative position of the ego-vehicle in the lane.
- Four Laser-scanner sensors, installed on the front, on the rear and on the two lateral sides of the vehicle, in order to detect and to

[2] The cognitive state is still under discussion.

reconstruct the surrounding scenarios and to select the obstacle(s) of interest.
- One internal camera to detect the head position of the driver (and where he/she is looking at).

All these sensors are used for the detection of both the external and the internal environment.
The AdCoS architecture is based on this configuration; more details are illustrated in the deliverable D9.4 [11].

5.1.1.2 Modelling techniques employed

As a starting point, Figure 1 gives an updated overview of the different MTTs (to be) used during the different steps of development process of the AdCoS Adapted Assistance, as described initially in D9.4 [11]. The blocks in dashed line are the ones whose implementation and use are still under definition.

The legend is the following:
- DDDBDVD = Detection of driver distraction based on data on vehicle dynamics
- DDC = Driver Distraction Classifier
- DIR = Driver Intention Recognition
- DM = Driver Model

The different MTTs are associated with a large set of modelling formalisms and techniques and are accompanied by the application of standard software and system development techniques. Abstract modelling of the different use-cases by terms of sequence diagram in the UML was already described in D9.3 [10]. RTMaps is used as the primary MTT to support the integration and enable joint simulations of the overall AdCoS and AdCoS modules. As such, the core formalisms and techniques used at this level are RTMaps Diagrams (some examples are provided in a next section), which can be interpreted as executable functional flow block diagrams. For data acquisition, supporting the development of the different AdCoS modules, several MTTs provided by WP5 were used. Finally, the development of the different AdCoS modules is heavily supported by task-modelling, and especially modelling techniques for human operator models provided by WP2.

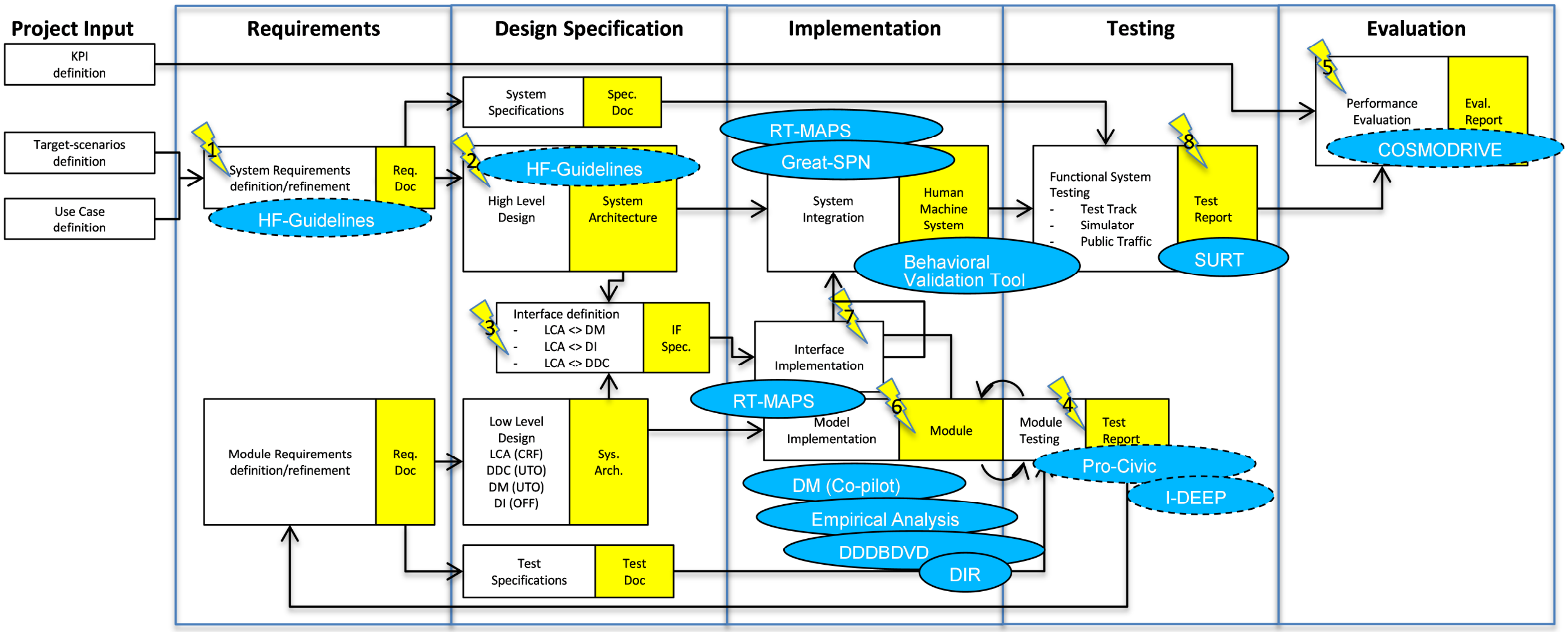

Figure 1: Development process for AdCoS Adapted Assistance.

5.1.1.3 Input to the modelling process from other work packages

Table 1 provides an overview over the input to the modelling process from other WPs in terms of MTTs included in the HF-RTP.

Table 1: Overview of MTTs used during the AdCoS development process.

MTT	WP	Use
Bayesian Autonomous Driver Mixture-of-Behaviours (BAD MoB) Models	2	**Module Implementation** The driver intention recognition module utilizes a BAD MoB model as a model of the human driver describing the probabilistic relations between the driver's intentions, behaviour, control actions, and the perceived environment as a means to answer probability queries of interest.
Driver intention recognition (DIR), (OFF)	3	**Module Implementation** The driver intention recognition is used within the CRF AdCoS to adapt the decision making process to the preferences and intentions of the human driver. In fact, selecting manoeuvres that are most suitable to the estimated intention of the human driver could improve the acceptance of the human operator.
RTMaps, (INT)	4	**Interface and Module Implementation, System Integration:** The tool used for the interface and module implementation, as well as for the system integration, is RTMaps. It has been selected from the HF-RTP, since it solves the issue on synchronization, operating system dependencies and gateway installations. RTMaps is well suited, since it provides a lot of interfaces to standard protocols, like the Controller Area Network (CAN), Ethernet, etc. In addition, it provides also the interfaces and the related blocks of many sensors used inside the CRF AdCoS and even when not originally present, their implementation was quite easy. Most of the selected modules used within this AdCoS will be developed in C++, and RTMaps allows easy integration of C++ modules. The usage of RTMaps is expected to highly decrease the interface implementation effort.
Great-SPN for	4	**Module Implementation**

MTT	WP	Use
MDP, (UTO)		Great-SPN is used in this AdCoS to model task allocation and transition between human-agent and machine-agent. Moreover, it is used to develop and implement the co-pilot concept, which the "core" of the AdCoS.
Surrogate Reaction Task (SuRT),(DLR)	5	**System Design, Module Implementation** It is a tool to create a kind of visual distraction in the user. In our AdCoS, it has been adapted to be used inside the vehicle and integrated in RT-MAPS. The goal is to induce a form of visual distraction in the driver and collect the data to develop the related classifier.
Empirical analysis of cognitive and communication process (SNV)	5	**System Design, Module Implementation** This is the used process to design the experiments for the driver distraction. A dedicated test-site has been selected, with segments where the SuRT was activated and others where it was deactivated.
Detection of driver distraction based on data on vehicle dynamics (DDDBDVD), (UTO)	5	**Module Implementation** This module MTT represents the classifier of the visual driver's distraction and it is based on ML techniques. The basic idea is to use only the vehicle dynamic data, the environment data and the ego-vehicle positioning data (with respect to the lane) an inputs to the classification algorithm.

Table 2 provides an overview over the potential input from other WPs in terms of MTTs included in the HF-RTP that are planned to be used or for which there is interest in use in the future.

Table 2: Overview of MTTs potentially used during the AdCoS development process in the future.

MTT	WP	Issue
COSMODRIVE (IFS)	2	**Module Testing and Evaluation** This simulation tool may be used to test the

MTT	WP	Issue
		AdCoS in two specific conditions: • Risky situations, for which tests on real vehicle can be dangerous. • Preliminary evaluation before the final implementation, to tune parameters, define thresholds, etc. •
Driver distraction model (DDM), (TWT)	2	**Module Implementation** This model estimates the distraction degree of the driver based on audio, video and behavioural driving information. Since a driver distraction classifier is also developed in CRF AdCoS from UTO partner, an exchange of information can be very useful, in particular if the model can be chosen as providing an additional input for the adaptation.
Driver distraction classifier (DDC), (TWT)	2	**Module Implementation** This tool classifies the distraction degree of the driver and gives as output the information whether the driver is cognitively distracted or not. Since a driver distraction classifier is also developed in CRF AdCoS from UTO partner, an exchange of information can be very useful, in order to compare results.
MagicPED (OFF)	2	Magic-PED will be used to formalize the task modelling of the Lane Change manoeuvre in place of Microsoft Power Point.
HF-Guideline, (EAD-IW-DE)	1	**Requirements Definition / System Design** The Human Factors Guideline could be used to define the system and all relevant aspects comprehensively and identifying potential issues in the system design at an early stage in the project. The number of iterations for designing the system could be reduced. Therefore both the CRF and IAS AdCoS applications are interested.
I-Deep, (INT)	4	**Module Testing:**

MTT	WP	Issue
		The evaluation of the overall AdCoS needs to be done in simulations, since no reproducibility is given for the real vehicle. The tools used for testing single modules depend on the module itself. However for the functional testing of modules, I-Deep is expected to be usable, since it is a server based application which allows to run RTMaps projects with variable parameters and stores the results in a database. It is therefore expected that I-Deep will allow an extensive testing of the AdCoS module functions. This opportunity will be investigated by the CRF AdCoS team.
Cosmo-SiVIC, (IFS)	4	Cosmo-SiVIC is a common platform involving the virtual environment, the vehicle, the cognitive driver modelling with its functionalities (eye modelling, gaze direction, mental driving environment representation …). It is possible to model and simulate the main AdCoS functionalities.
ProSivic, (CVT)	4	**Module Testing** ProSivic may be useful for generating simulation scenarios for offline testing. The interface to RTMaps allows using all of the AdCoS modules to be tested with simulated data. Most critical scenarios can be simulated in this way, without any risks.

5.1.2 AdCoS and Module Models

Figure 2 represents a high level model of the AdCoS Adapted Assistance and its modules from a functional point of view. RTMaps is used as the primary MTT to support the integration and enable joint simulations of all modules. RTMaps provides a graphical editor which can be used to model the flow of data samples between functional blocks called components. As such, RTMaps may be seen as a modelling tool for executable functional flow block diagrams. Consequently, the different modules are modelled respective implemented in RTMaps. Abstract modelling of the different use-cases by terms of sequence diagram in the UML was already described in D9.3 [10].

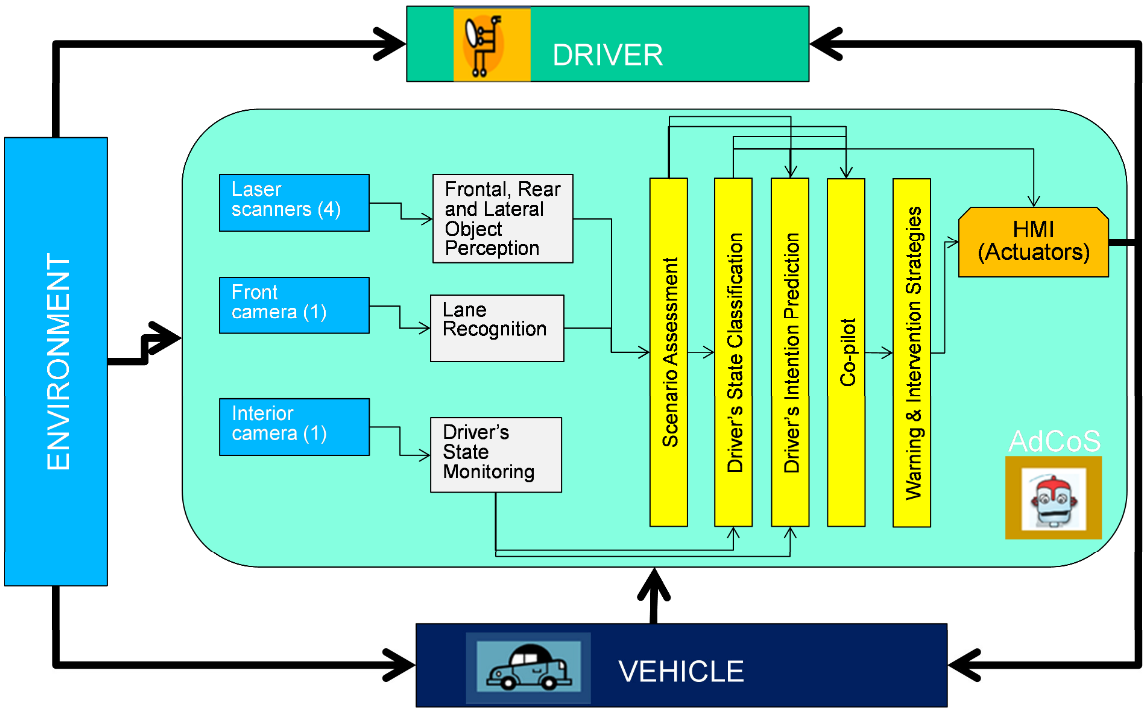

Figure 2: High level AdCoS model of the AdCoS Adapted Assistance, indicating the main functional blocks and information flow.

5.1.2.1 Co-Pilot

The artificial intelligence for the adaptive assistance driving is implemented within the machine-agent, where the driving process can be broken down into four stages:

- the perception of the traffic environment around the host vehicle in real-time,
- the interpretation and assessment of the current traffic situation,
- the planning of appropriate manoeuvres and actions and
- the action to control the vehicle and guide it safely along the planned trajectory.

For the Co-pilot, the basic idea is to adopt a statistical approach; in particular, the AdCoS is modelled as an MDP (Markov Decision Process), in order to construct optimal warning and intervention strategies (WISs).

An instantiation of the MDP COPILOT framework presented in D3.4 [3] will be described in the context of use-case UC4 (c.f., D9.3 [10]). Then, each component module of the MDP COPILOT is tailored to this use case and briefly outlined below. However, the proposed MDP techniques could be applicable in other domains and use-cases.

5.1.2.1.1 Instantiation of the Distraction module

The distraction classifier, called SLFN is trained by using data related to distraction and vehicle dynamic collected by means of dedicated experiments. In particular, the vehicle dynamic data considered are the following:

- Speed [m/s]
- Time To Collision [s]
- Time To Lane Crossing [s]
- Steering Angle [deg]
- Street Curvature [deg]
- Lateral Position [m]
- Lane width [m]
- Position of the accelerator pedal [%]
- Position of the brake pedal [%]
- Turn indicator [on/off]
- Yaw rate [deg/s]

These values are directly available on the prototype CAN bus. The frequency of data collection is about 20 Hz (1 data-point each 0.05 s). Values are then averaged over a period of 1.8 s in order to be consistent with the target variable (distracted or not-distracted). For the time being, we just considered only two possible levels of driver distraction.

Because of the way the training set was designed, we consider here the visual distraction (eyes off the road). Although we cannot directly address other types of distraction (e.g. cognitive) by this experiment, nonetheless visual distraction is associated with greater odds to crash-relevant conflict than cell phone conversation (cognitive distraction).

Thus, the classifier module provides the decision maker with the distraction state of the driver.

5.1.2.1.2 Instantiation of the Cognitive module

The cognitive framework derives the information of the human driver status, based on the cognitive architecture CASCaS. Perception and motoric information are provided by internal camera sensors that point to the human. The goal is to provide distraction level that is necessary for the decision process, which relays on this data to take different decisions and provide different warnings.

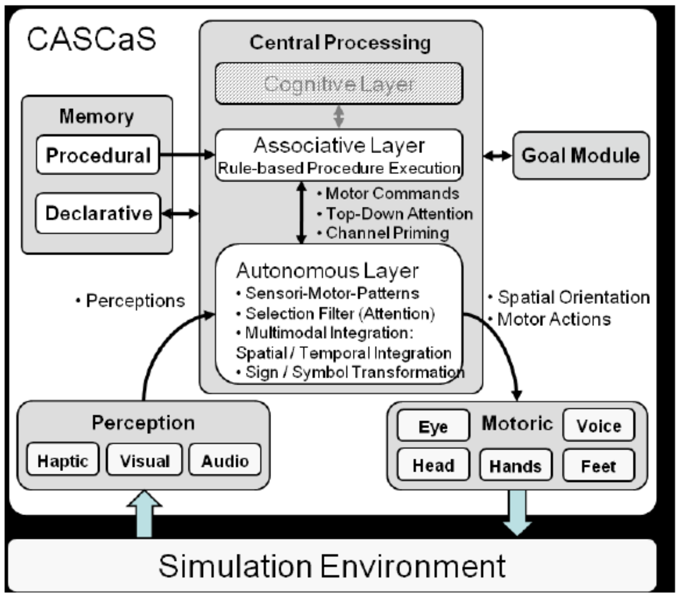

Figure 3 : Cognitive architecture CASCaS.

5.1.2.1.3 Instantiation of the MDP module

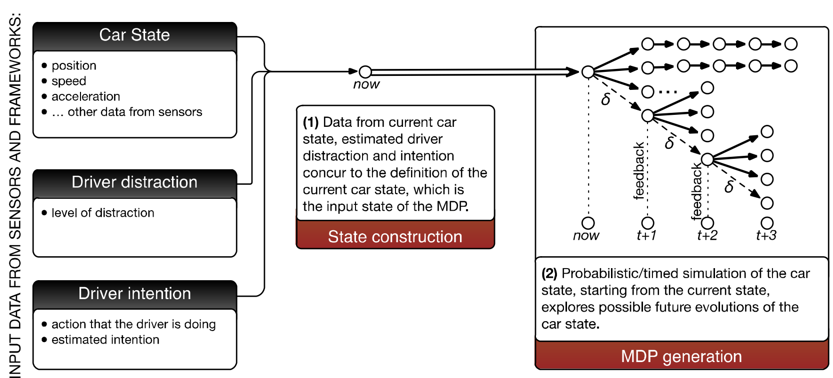

Figure 4 : MDP State

According to Figure 4, the MDP state is described in terms of:

- Car state taken from the car sensors (see variables for the classifier). Uncertainty in the measurements will be considered and

appropriate models (such as bounded parameter MDP) used in the reasoning process
- Human driver intention, predicted by the DIR module
- Human driver distraction level

MDP macro actions are defined according to the Automotive Workpackage specification, and are drive, accelerate, decelerate, brake, change lane, overtake, which will be decomposed in a sequence of lower actions during their actuation.

5.1.2.1.4 Instantiation of the Actuation and Adaptive module

In the use-case UC4 context (c.f., D9.3 [10]), actuation takes as input macro actions (drive, accelerate, decelerate, brake, change lane, overtake), and actuates them by checking the actual actions done by the human driver. Figure 5 illustrates the relation between the macro actions (strategy) decided by the MDP, and its actuation in terms of low-level actions.

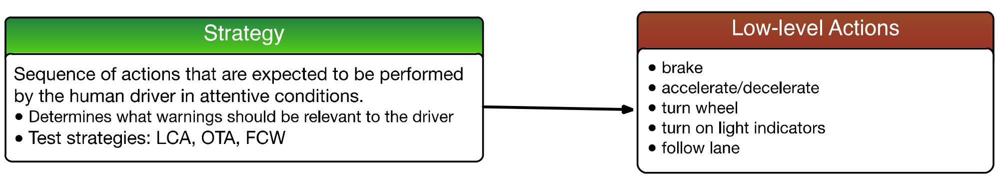

Figure 5 : MDP strategy to low level action

5.1.2.2 Driver Intention Recognition

The Driver Intention Recognition (DIR) module is a system component within the overall AdCoS application for Adapted Assistance (c.f., Figure 2) that provides other components with online-assessments of the intentions and behaviors of a single human driver. These assessments will be used to adapt the decision making process of the AdCoS to the preferences and intentions of the human driver. As the name suggests, the DIR module solely focusses on the automotive domain addressed in WP9. However, the general techniques used are domain-independent and should be applicable in other domains and use-cases.

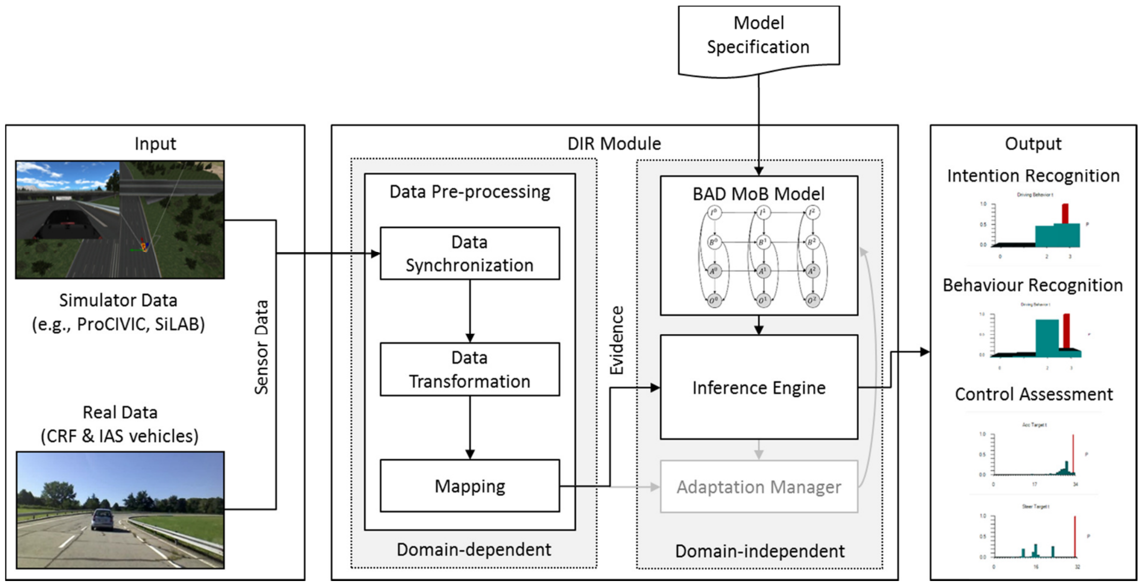

Figure 6: Schematic overview of the DIR module.

Figure 6 shows a schematic overview of the DIR module. It consists of two primary components, a domain-*dependent* part (tailored to the actual system architecture and specification of the AdCoS for adapted assistance) that primarily deals with pre-processing of the available sensor input, and a domain-*independent* part consisting of an *inference engine* that enables the DIR module to answer probabilistic queries in respect to a probabilistic model of the human operator defined in an XML-based specification. Both parts are modelled and provided as separate RTMaps packages, so that the domain-independent inference engine can potentially be used in different domains utilizing different probabilistic models.

5.1.2.2.1 Modelling

In respect to the DIR module, modelling refers to both model-based design of the DIR module itself and the development of the probabilistic model of the human operator used within the DIR module. So far, the modelling efforts mainly resulted in:

- Modelling/implementation of an initial data pre-processing component in RTMaps based on exemplary datasets provided by CRF and experimental data performed for the driver distraction classifier (a screenshot is given in Figure 8).
- Modelling/implementation of a first version of the domain-independent inference engine of the DIR module in RTMaps (a screenshot is given in Figure 9).

- Prototyping of initial human operator models for driver intention recognition based on experimental data obtained in simulator studies prior to HoliDes.

5.1.2.2.1.1 Development of the DIR module

The primary MTT used for modelling the DIR module is RTMaps. RTMaps is a framework for real time data processing that enables component-based development of software and system applications. It provides a graphical editor which can be used to model the flow of data samples between functional components, which may be hierarchically broken down into simpler components. As such, RTMaps can be seen as a modelling tool for executable functional flow block diagrams.

We used RTMaps to develop the DIR module as an executable model that can be used for model-based analysis and model-in-the-loop simulations prior to the actual implementation on target architectures. Where necessary, we used the C++ API provided by RTMaps to implement new functional components. A schematic overview of the DIR module as modelled in RTMaps is shown in Figure 7.

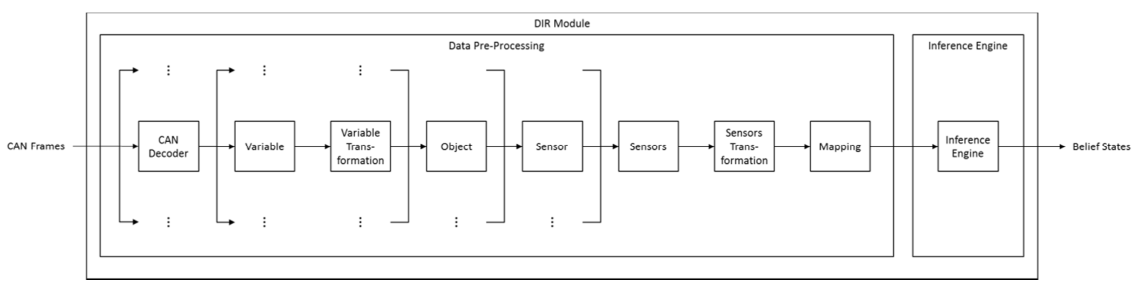

Figure 7: Schematic overview of the DIR module modelled in RTMaps.

The DIR module is designed as a single RTMaps component that is composed of a *Data Pre-Processing* and an *Inference* component. The DIR module requires input in form of CAN Frames and provides output in form of marginalized belief states over the current intentions, behaviors, and lateral resp. longitudinal control actions of the human driver.

Within the *Data Pre-Processing* component, the CAN Frames are decoded to data values using a set of *CAN Decoder* components (provided by RTMaps) according to CAN database files (provided by CRF). According to exemplary datasets provided by CRF, different sensors will require separate *CAN Decoders* that will provide sensor measurements at different frequencies and in different data formats. Furthermore, as each sensor might fail to produce valid measurements due to absence of objects to measure, high noise, or sensor failure, each measured value might not be defined at any point in time. Each measurement that is of interest in the context of the DIR module is connected to a separate *Variable* component that samples its input with a user-defined frequency (e.g. each 50ms). By matching the sampling periods of each *Variable*, the DIR module is able to synchronize the different measurements. Now synchronized, each measurement is sent to a *Variable Transformation* component that may annotate a measurement with temporal derivations (i.e., delta values). In the following, semantically related outputs of the *Variable Transformations* are hierarchically combined to objects (*Object* components), sensors (*Sensor* components) and finally a single data structure containing all measurements (*Sensors* component). The data structure is send to a *Sensors Transformation* component, where new measurements are derived as a function of others (e.g., time-to-collision or time-headway). Lastly, the now complete set of measurements is send to the *Mapping* component, which orders them in three vectors, one containing all continuous measurements, one containing validation flags to indicate whether or not a measurement is valid, and one containing all discrete measurements. A screenshot of the *Data Pre-Processing* component modelled in RTMaps is shown in Figure 8.

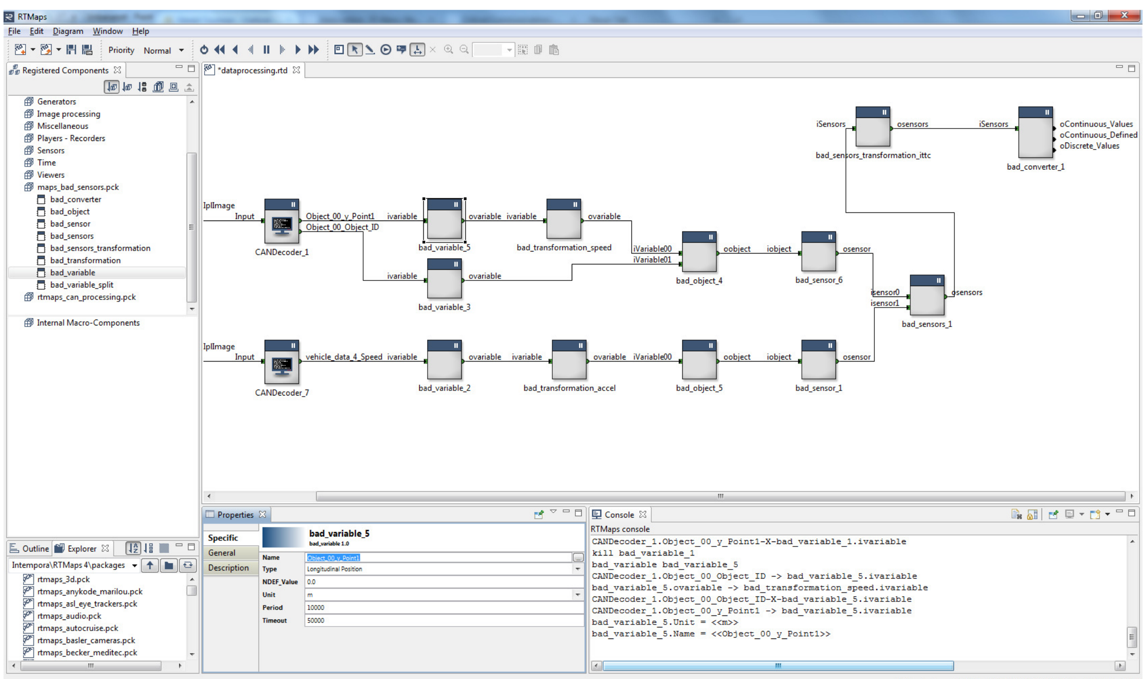

Figure 8: Screenshot of a snippet of the data pre-processing component of the DIR module modelled in RTMaps.

The *Inference* component currently only consists of a single RTMaps component, called the *Inference Engine*. It interfaces to the *Data Pre-Processing* in that it requires input in form of three vectors for continuous values, corresponding validation flags, and discrete values. The *Inference Engine* utilizes a probabilistic human operator model, called a Bayesian Autonomous Driver Mixture-of-Behavior (BAD MoB) model. A BAD MoB model is a Dynamic Bayesian Network (DBN) that defines a conditional probability distribution (CPD) over sets of discrete and continuous random variables[3] representing *(control) actions* $\boldsymbol{A}$, *behaviours* B, *intentions* I, and *(context) observations* $\boldsymbol{O}$ according to the following general factorization for an arbitrary number of T time slices:

$$P(\boldsymbol{A}^{1:T}, B^{1:T}, I^{1:T} | \boldsymbol{o}^{1:T}) = P(\,I^1 | \boldsymbol{o}^1) P(B^1 | I^1, \boldsymbol{o}^1) P(\boldsymbol{A}^1 | B^1, I^1, \boldsymbol{o}^1)$$
$$\prod_{t=2}^{T} P(\,I^t | I^{t-1}, \boldsymbol{o}^t) P(B^t | B^{t-1}, I^t, \boldsymbol{o}^t) P(\boldsymbol{A}^t | \boldsymbol{A}^{t-1}, B^t, I^t, \boldsymbol{o}^t).$$

The model describes, how the perceivable environment conditions the formation of intentions $P(\,I^t | I^{t-1}, \boldsymbol{o}^t)$, how intentions and the perceivable environment conditions the selection of skills/behaviours $P(B^t | B^{t-1}, I^t, \boldsymbol{o}^t)$ and how the intention, selected behaviour, and the perceivable environment conditions the selection of control actions $P(\boldsymbol{A}^t | \boldsymbol{A}^{t-1}, B^t, I^t, \boldsymbol{o}^t)$. The *Inference Engine* provides an interface to select an XML model-specification that defines the BAD MoB model used during runtime. The BAD MoB model must be specified in terms of:

1. The definition of all template-variables (variables that will be instantiated for each time-slice t) $\boldsymbol{\mathcal{X}}$ in the model. Currently, these consist of variables representing the lateral and longitudinal control actions of the human driver $\boldsymbol{A} = \{A_{Lat}, A_{Lon}\}$, behaviors B, intentions I, and a set of observation variables $\boldsymbol{O} = \{O_1, \dots, O_m\}$: $\boldsymbol{\mathcal{X}} = \{\boldsymbol{A}, B, I, \boldsymbol{O}\}$.

2. Under the assumption of a first-order Markovian system, a factorization for the initial time-slice $P(\boldsymbol{A}^1, B^1, I^1 | \boldsymbol{O}^1)$, e.g.

[3] Variables will be denoted by capital letters, such as X, Y, Z, while specific values taken by those variables will be denoted by corresponding lowercase letters x, y, z. The set of values that a random variable X can take will be denoted by $Val(X)$. We use boldface type capital letters $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$ to denote sets of random variables (e.g., $\boldsymbol{X} = \{X_1, \dots, X_n\}$) and corresponding boldface lowercase letters $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$ to denote assignments of values to the variables in these sets (e.g., $\boldsymbol{x} = \{x_1, \dots, x_n\}$).

$$P(\boldsymbol{A}^1, B^1, I^1 | \boldsymbol{O}^1) = P(I^1 | \boldsymbol{O}^1) P(B^1 | I^1, \boldsymbol{O}^1) P(A^1_{Lat} | B^1, \boldsymbol{O}^1) P(A^1_{Lon} | B^1, \boldsymbol{O}^1),$$

and a factorization for the 2TBN $P(\boldsymbol{A}^t, B^t, I^t | \boldsymbol{A}^{t-1}, B^{t-1}, I^{t-1}, \boldsymbol{O}^t)$, e.g.

$$P(\boldsymbol{A}^t, B^t, I^t | \boldsymbol{A}^{t-1}, B^{t-1}, I^{t-1}, \boldsymbol{O}^t) \\ = P(I^t | I^{t-1}, \boldsymbol{O}^t) P(B^t | B^{t-1}, I^t, \boldsymbol{O}^t) P(A^t_{Lat} | A^{t-1}_{Lat}, B^t, \boldsymbol{O}^t) P(A^t_{Lon} | A^{t-1}_{Lon}, B^t, \boldsymbol{O}^t).$$

3. A set of parameters $\theta_{X,\boldsymbol{Pa}(X)}$ for each CPD $P\big(X|\boldsymbol{Pa}(X)\big)$ that is sufficient to identify a function $f\big(x, \boldsymbol{pa}(x), \theta_{X,\boldsymbol{Pa}(X)}\big) = P\big(x|\boldsymbol{pa}(x)\big)$ that given $\boldsymbol{pa}(x)$ and x, returns the conditional probability $P\big(x|\boldsymbol{pa}(x)\big)$ according to the parameters $\theta_{X,\boldsymbol{Pa}(X)}$.

Given a BAD MoB model that defines the CPD $P(\boldsymbol{A}^{1:T}, B^{1:T}, I^{1:T} | \boldsymbol{o}^{1:T})$, it can be used to answer any probability query over subsets of $\{\boldsymbol{A}^{1:T}, B^{1:T}, I^{1:T}\}$ conditioned on $\boldsymbol{o}^{1:T}$. By now, we restrict our focus on the following set of probability queries which can be obtained from the joint belief state over the current actions, behaviors, and intentions of the human driver $P(\boldsymbol{A}^t, B^t, I^t | \boldsymbol{a}^{1:t-1}, \boldsymbol{o}^{1:t})$:

- $P(I^t | \boldsymbol{a}^{0:t-1}, \boldsymbol{o}^{0:t})$: The marginalized belief state of the current intentions, given all available information about the driver's actions and the environment.
- $P(B^t | \boldsymbol{a}^{0:t-1}, \boldsymbol{o}^{0:t})$: The marginalized belief state of the current behaviours, given all available information about the driver's actions and the environment.
- $P(A^t_{Lat} | \boldsymbol{a}^{1:t-1}, \boldsymbol{o}^{1:t})$: The marginalized belief state of the current lateral control actions (i.e., steering wheel angles), given all available information about the prior driver's actions and the environment.
- $P(A^t_{Lon} | \boldsymbol{a}^{1:t-1}, \boldsymbol{o}^{1:t})$: The marginalized belief state of the current longitudinal control actions (i.e., a combined acceleration-braking pedal position), given all available information about the prior driver's actions and the environment.

For each of these queries, in the case of discrete query variables, the *Inference Engine* (and subsequently the DIR module) provides a vector containing the probabilities for the different assignments of the variable, or, in the case of continuous query variables, the mean and variance of CPD. A concrete summary for the current version of the DIR module is given in Table 3.

Table 3: Output of the DIR module.

Output	Description
Current Intentions	Vector containing the probabilities $P(i^t | \boldsymbol{a}^{1:t-1}, \boldsymbol{o}^{1:t})$ for each intention

	$I^t = i^t$
Current Behaviours	Vector containing the probabilities $P(b^t \mid \boldsymbol{a}^{1:t-1}, \boldsymbol{o}^{1:t})$ for each behaviour $B^t = b^t$
Expected steering wheel angle	Vector containing the mean and variance of a Gaussian representing $P(A_{Lat}^t \mid \boldsymbol{a}^{1:t-1}, \boldsymbol{o}^{1:t})$
Expected position of a combined acceleration-braking pedal position	Vector containing the mean and variance of a Gaussian representing $P(A_{Lon}^t \mid \boldsymbol{a}^{1:t-1}, \boldsymbol{o}^{1:t})$

Although currently not included in the actual DIR module, we implemented a number of additional RTMaps components that provide functionality for an actual classification of this output, which, given a vector containing the probabilities $P(Y \mid \boldsymbol{E} = \boldsymbol{e})$ for a discrete variable Y as input, provides the assignment with the highest probability $\arg\max_y P(y \mid \boldsymbol{E} = \boldsymbol{e})$ as output. A screenshot of the *Inference Engine* component model in RTMaps is shown in Figure 9.

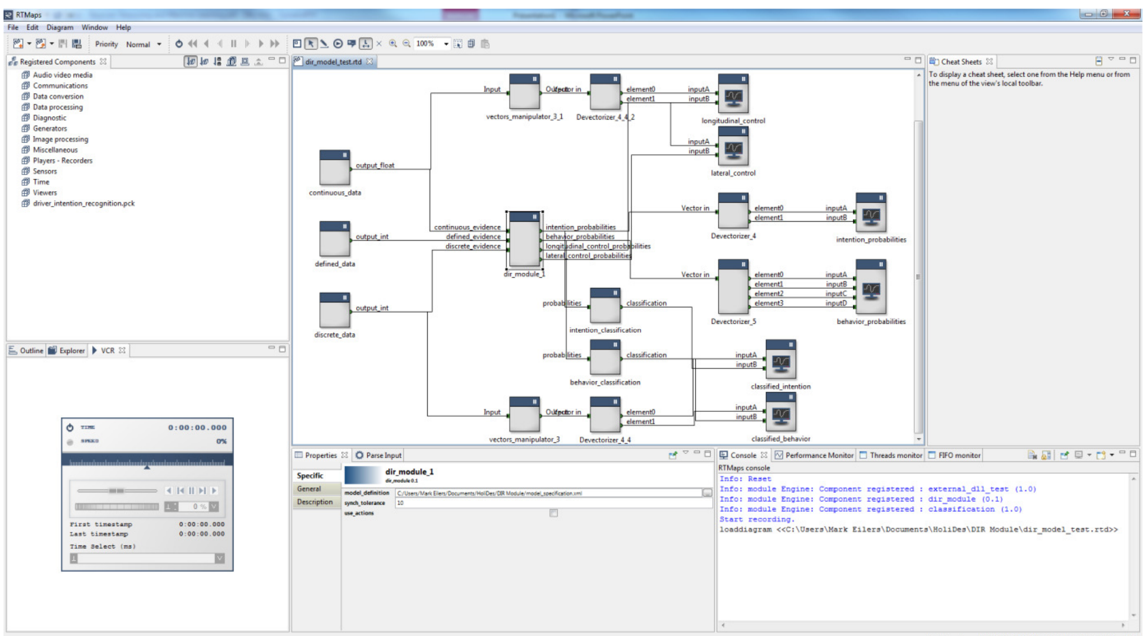

Figure 9: Screenshot of the inference engine of the DIR module modelled in RTMaps.

5.1.2.2.1.2 Prototypical development of BAD MoB models

In order to provide any output, the DIR module requires a probabilistic model of the human operator. These models need to be learnt from multivariate time-series of human behaviour traces. As dedicated experiments to obtain these time-series for driver intention recognition will be conducted in July-August 2015, by now, we rely on experimental data obtained prior to HoliDes in simulator experiments that focused on comparable highway scenarios [12]. We used this data to learn a prototypical human operator model according to the techniques described in [12] that can be utilized in the DIR module to prepare a workflow for test and validation. The model utilizes a set of approx. 100 observation variables that can be obtained from detailed information about traffic participants in the vicinity of the driver, the future path of the road, the state of the driver's vehicle, the driver's control behavior, and general contextual information (a detailed description of the required input can be found in the confidential part of D3.5 [4]).

The general development workflow for learning and testing BAD MoB models is shown in Figure 10. We adjusted the *Data Pre-Processing c*omponent modelled in RTMaps to transform experimental data in the form necessary for parameter and structure learning. The actual parameter and structure learning is done using a set of internal workflows and tools developed by OFFIS. The general techniques employed are described in [12]. The result of learning is the XML-specification of a BAD MoB model and a set of learning results, which is stored in a database. For testing, we reserve a set of data not used during training that provide an independent measure of performance. Testing itself can be performed using the DIR module within RTMaps, where we replace the *Data Pre-Processing* component by the already pre-processed test data. The result of testing is a set of test results stored in a database. Both learning and test results are analysed and visualized in R, a free software environment for statistical computing and graphics.

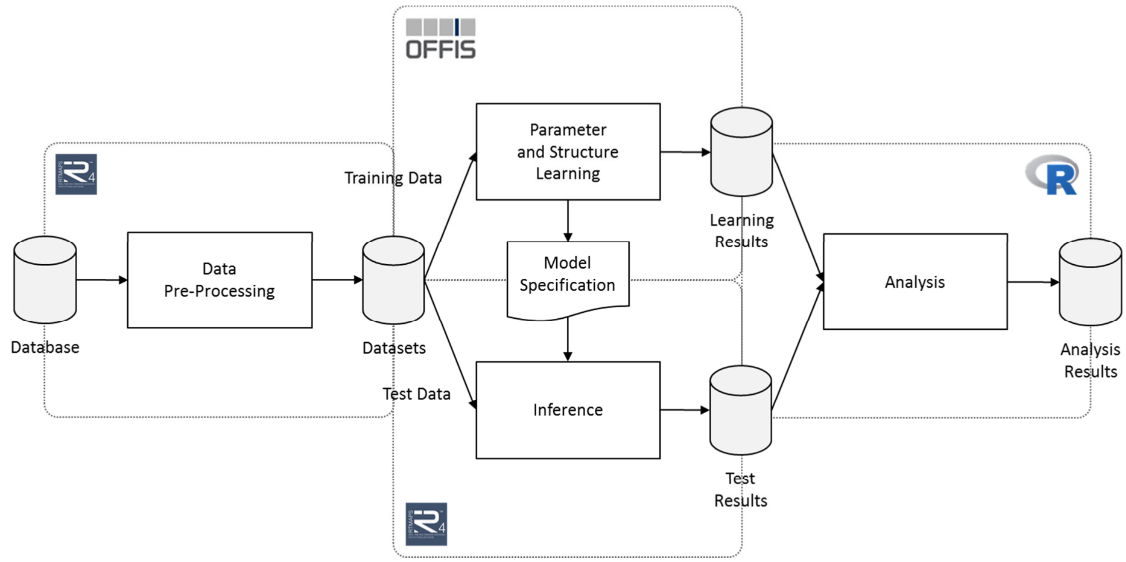

Figure 10: Workflow for prototypical development of BAD MoB models for intention recognition.

5.1.2.3 Driver Distraction Classifier

In AdCoS Adapted Assistance, the detection of the driver distraction is based mainly on the data of vehicle dynamics.
Since a machine learning (ML) approach has been selected, a specific and dedicated experimental phase has been carried out from CRF and SNV partners, based on the Empirical analysis of cognitive and communication processes (as described in the deliverable D5.4 [7]).

Table 4: List of variables used for the Distraction classifier development.

ID	Signals	Meaning	Unit of Measure
1	Time	time stamp (from RT-MAPS)	[ms]
2	Offroad	"1" means, the driver is looking off the road "0" means, the driver is looking in front of te road	[#]
3	VTSD	Visual Time Sharing Distraction - distraction level in %	[ms]
4	VDD	Visual Distraction Detection - duration of the current diustraction [ms]	[ms]
5	Xpos	X, Y, Z head position	[mm]
6	Ypos		
7	Zpos		
8	Xangle	head orientation along X, Y, Z	[deg]
9	Yangle		

ID	Signals	Meaning	Unit of Measure
10	Zangle		
11	Head Quality	confidence of head data (referred to six measures before)	[%]
12	Left Eye Opening Measurements	how much left/right eyes are opened	[mm]
13	Right Eye Opening Measurements		
14	Left Eye Opening Quality	confidential level of measures before	[%]
15	Right Eye Opening Quality		
16	lateral distance	distance between longitudinal axis of the HV and the center of the lane	[m]
17	curvature	lane curvature of road ahead (30-50)m	[m^-1]
18	DTCT Distance	distance at which the camera is looking the curvature data are referred to it.	[m]
19	Lane Width	Width of the lane	[m]
20	Heading angle	Yaw-angle between the longitudinal axis of HV and the tangent line to the lane	[deg]
21	Turn indicator	if indicators are used: 0 = not used; 1: right 2: left	[#]
22	Lws_angle	steering angle	[deg]
23	brake pedal	if the pedal is pressed or not 0: no 1: yes	[#]
24	gas pedal	accelerator pedal position	[%]
25	vehicle speed	velocity of HV	[km/h]
26	yaw-rate	yaw-rate of HV	[deg/s]
27	Num_of_task	number of SURT tasks presented to the user	[#]
28	Num_of_correct_ans	number of corrected answers of users	[#]
29	Num_of_incorrect_ans	number of not corrected answers of users	[#]
30	target_position	position of the SURT target	[#]
31	action_position	where the user has pressed	[#]
32	Show_task	if the task is actually shown to the user 0 = baseline; 1 = secondary task ON	[#]
33	X_pos_selObj	X, Y position of selected obstacle	[m]
34	Y_pos_selObj		
35	X_speed_selObj	X, Y velocity of selected obstacle	[m/s]
36	Y_speed_selObj		

ID	Signals	Meaning	Unit of Measure
37	ID_selObj	Provide a vector (size=number of objects) containing the ids of each object	[#]
38	Ages_selObj	Provide a vector (size=number of objects) containing the ages of each object	[#]
39	Class_selObj	Provide a vector (size=number of objects) containing the class of each object	[#]
40	Age_Class_selObj	Provide a vector (size=number of objects) containing the classification age of each object	[#]
41	Class_Confidence	Provide a vector (size=number of objects) containing the classification confidence of each object	[#]
42	Maneuver	Maneuver planned (**Overtaking** for us): 0: No Overtaking (OV) 1: OV	[#]
43	Status	Status of maneuver (OV) 0: running / in progress 1: abort	
44	Sub_maneuvers	Elementary maneuvers, constituting the OV: 1: Lane Keeping (LK) 2: Lane Change (LC) left 3: Passing 4: LC right 5: Second LK (coming back to the original lane)	[#]
45	TTC	Time-To-Collision (defined as the ratio between distance and relative velocity)	[s]
46	HD	Headway (defined as the ratio between distance and HV velocity)	[s]

The data are collected by using RTMaps tool with a frequency of 50ms.

Time-To-Collision (TTC) and Headway (HD) are computed in post-processing.

Starting for these data, specific datasets will be derived; in particular, following the ordinary procedure for supervised learning (SL), each data set has been split in three different subsets:

- Training data (around 60% of the whole dataset), which are presented to the network during training and the network is adjusted according to its error.

- Checking data (around 15% of the whole dataset), which are used to measure network generalization and to halt training when generalization stops improving.
- Testing data (around 25% of the whole dataset), which have no effect on training and so provide an independent measure of network performance during and after training:

In addition, data of distraction constitute the target set (SL approach). More details about the experimental procedure can be found in the deliverable D5.4 [7], here only a summary is reported.
The distraction has been induced by means of a secondary visual research task, called SURT (Surrogate Research Task), reproduced on an in-vehicle display system (7'' TFT touch screen installed on the right-hand side of the car cabin) and whose SW has been integrated in RT-MAPS as well, which is the tool used for the data collection.

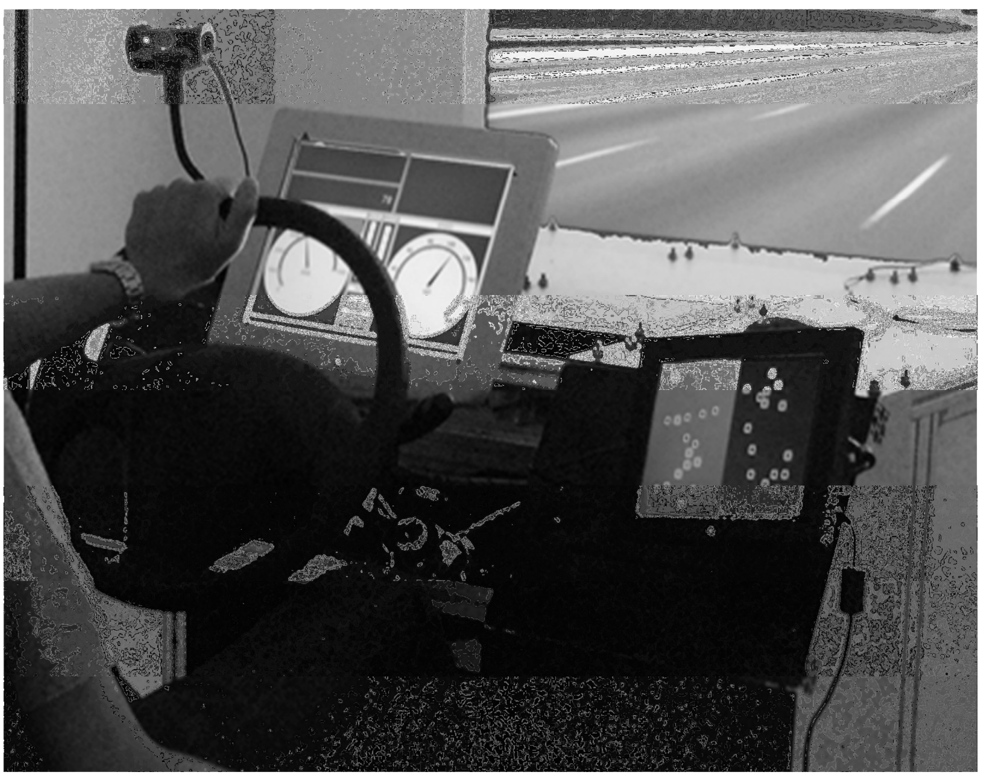

Figure 11: SURT display on the right part of prototype vehicle cockpit

Participants were asked to drive for around 60 minutes on a pre-defined test-site, which comprehended both extra-urban/rural roads and motorways. During this driving phase, each participant was asked to complete some secondary task sessions: when SURT is activated the display shows a black/grey screen with 30 symbols (each 1.4 cm high), specifically: 14 blue circles, 15 red squares and 1 red circle. The screen is equally divided into two vertical sides and each time the SURT is presented the driver is asked to touch the side where the red circle is located. The time interval between two consecutive screens was pseudo-randomized between 3 and 9 seconds

It is worth to note here that only the vehicle dynamic and scenarios variables constitute the inputs of the classifiers; the head position, as provided by the internal camera, does not appear, since it has been used only to construct the target set.

More details on that can be found in the deliverables D2.5 [2] and D3.5 [4].

5.1.2.4 HMI of the Lane Change Assistant

As detailing described in D9.4 [11], even though the HMI is part of the AdCoS Adapted Assistance, the HMI is implemented following a different development process compared to the LCA.

Figure 12 shows the description of the development process of the HMI for the AdCoS Adapted Assistance (the Android app) and the MTTs that will be employed to improve the development process and the overall quality (a description is given in the confidential part of D9.5 Feedback and HF-RTP Requirements Definitions Update).

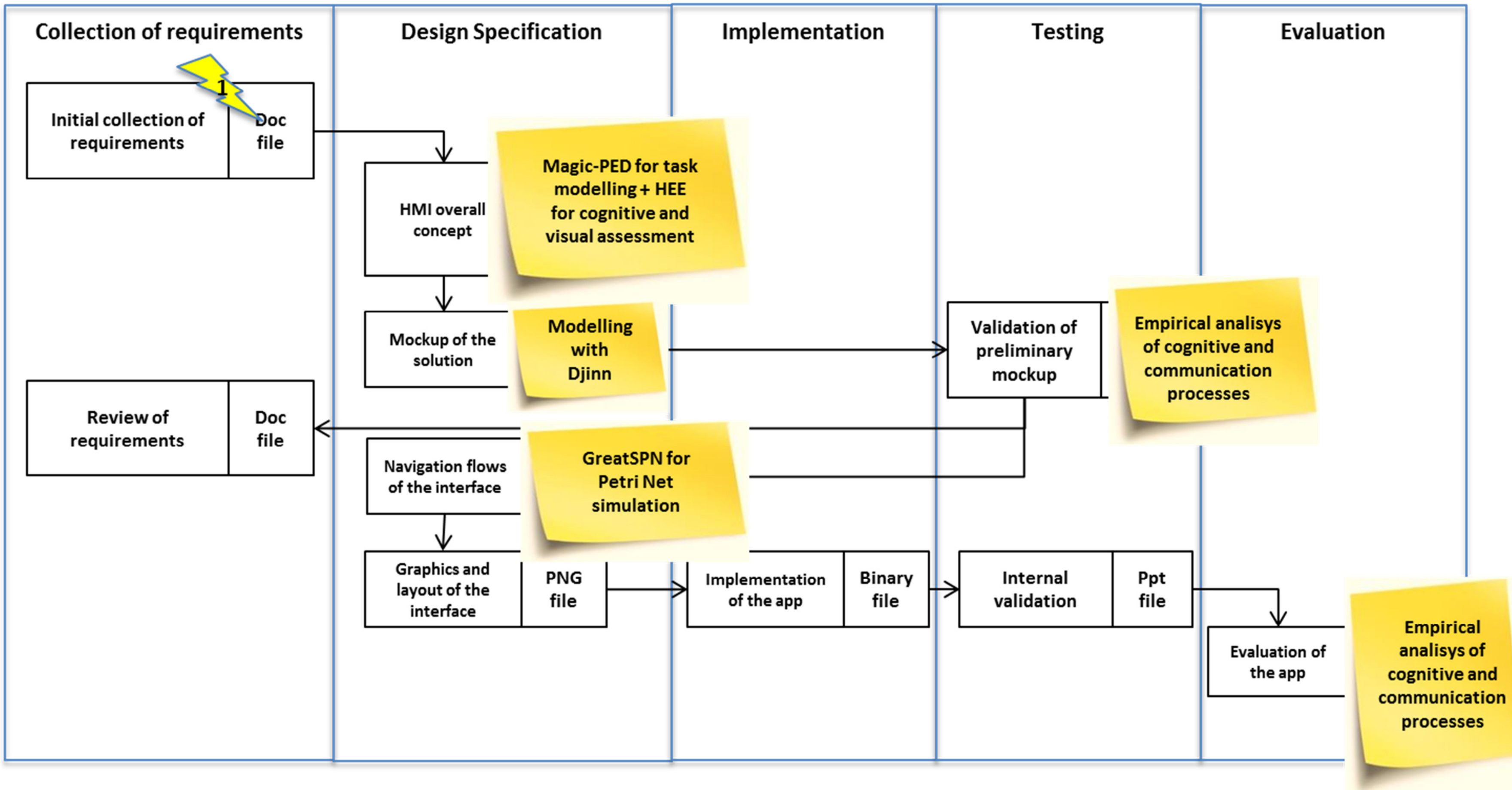

Figure 12: Selection of MTTs to improve the development process of the HMI of the Lane Change Assistant developed by CRF.

So far, for the development of the HMI for the LCA of CRF REL has carried out the following activities:

- Modelling of the tasks the driver must perform to complete the lane change and the overtaking manoeuvre: the tasks modelling has been performed by using Microsoft Power Point, so the next step will be employing Magic-PED to assess the improvements and the benefit it can bring
- Analysis of the tasks in order to identify the cognitive and visual load of each of them: the MTT Human Efficiency Evaluator (HEE) will be tested to check if it can provide an automatic evaluation of the cognitive and visual load of the driver during the lane change and overtaking manoeuvre.
- By considering the cognitive and visual loads, the HMI concept has been defined
- A preliminary draft of the graphical interfaces has been completed, in order to associate different messages according to the state of the driver (cognitive and visual loads) and the intention of the driver (provided by the Driver Intention module).

The results of these activities have already been detailed in D9.3 [10].

The activity conducted by REL was meant to identifying the cognitive and visual load of each task to provide appropriate information to the driver (by also considering the most suitable interaction modality to allow the driver processing this information in continuously changing conditions).

Therefore, REL mainly focused on the “what”, i.e. what the driver should do in each condition (e.g. “keep the lane”, “change the lane”, etc.).
However, recent studies [13] has shown the importance of providing “why” information describing reasoning for actions to achieve better driving experience in (semi-autonomous) adaptive vehicles. The explanation of the “why” also affects the driver’s attitude and safety performance.

Therefore, in collaboration with WP3, REL is improving the preliminary HMI concept by including innovative communication strategies to describe the “why” in a way that is suitable to the cognitive and visual load of the driver, by also considering additional multi-modal (visual, acoustic, haptic and tactile) interfaces (that will be embedded in the vehicle to complement the information provided by the app).

In particular, alternative HMI concepts and solution will be developed to provide the information about the “why”, e.g. direction haptic feedback in the seat when a car is approaching from the back and the “keep the lane” message is displayed (the haptic feedback intuitively explains that the driver should keep the lane because a car is approaching from the back).

Moreover, in collaboration with SNV, experiments in real environment and simulation will be conducted at the end of this cycle to assess the effectiveness of each multi-modal message for each driving task and subtask, in order to identify the best set of feedback to be employed for the HMI of the AdCos and in similar safety-critical conditions.

Finally, for the actual development of the app, REL is starting integrating the Djnn libraries into the third-party software QT used for the development of the Android app to test the chance of easily improving the interaction with the app.

5.2 AdCoS Adapted Automation

5.2.1 Description of the AdCoS

Today the development of highly automated driving is the research focus of many OEMs and research institutes. A major need regarding automated vehicles is an increased usability and operability for the human driver. This encompasses cooperation and adaption of the machine agent to the human driver and other road users, with a human-centred design process as the foundation of the system development. The main challenges are the development of a fluent, yet transparent task allocation and transition between human driver and the machine agent and at the same time integrating the host vehicle into the flow of other road users, where a number of agents are acting in a shared space with shared resources. This aims at increasing the confidence of the human driver in a highly automated system, as described by vehicle automation level 3, which is defined by the National Highway Traffic Safety Administration (NHTSA).

The novelty of the automated driving approach presented here is the advanced interaction with a human driver and adaptation to his or her capabilities, needs and preferences, to other road users and the environmental conditions.

5.2.1.1 Operational definition

The main claim of vehicle automation systems is providing more freedom to the driver to work or relax while travelling and at the same time increase the safety. But, as many user studies have shown, the trust of the user in the automated system is a major issue. The human-machine interaction in current systems is often complex and the behaviour of the automated system remains untransparent to the driver.
To increase the user acceptance in vehicle automation this AdCoS provides an intuitive interaction concept. Transitions between manual and

automated driving are fluent, allowing the driver to interact with the automated system at any time and with minimal control inputs. Intentions of the driver are anticipated at a manoeuvring level, while the automation system takes care of the low level vehicle control. An HMI provides situation dependent information about planned manoeuvres to the driver. The interaction is adaptive to the characteristics of the human driver as well as his current situation awareness.
The driving characteristics of the automated system are adapted to the preferences of the individual driver and the level of distraction. Adaptation to the external context improves the driving performance in complex traffic situations.

5.2.1.2 Modelling techniques employed

Figure 13 shows the development process for the Adapted Automation AdCoS and applied MTTs. The graph shows the main tasks for the modelling of the overall system as well as the MTTs which are applied in the development process.
System requirements were derived from the use cases and the KPIs, which were provided as input to this project in the technical annex of the project proposal. Standard software and system development techniques were applied for the definition and verification of these requirements. A high level system architecture was derived, as it is illustrated in Figure 14.
There are only very few MTTs available to support the first two process steps. Most MTTs applied in this development are not part of the development process, but part of the product to be developed, such as the DDM or CONFORM. RTMaps serves as a framework for prototyping during the implementation of the AdCoS.

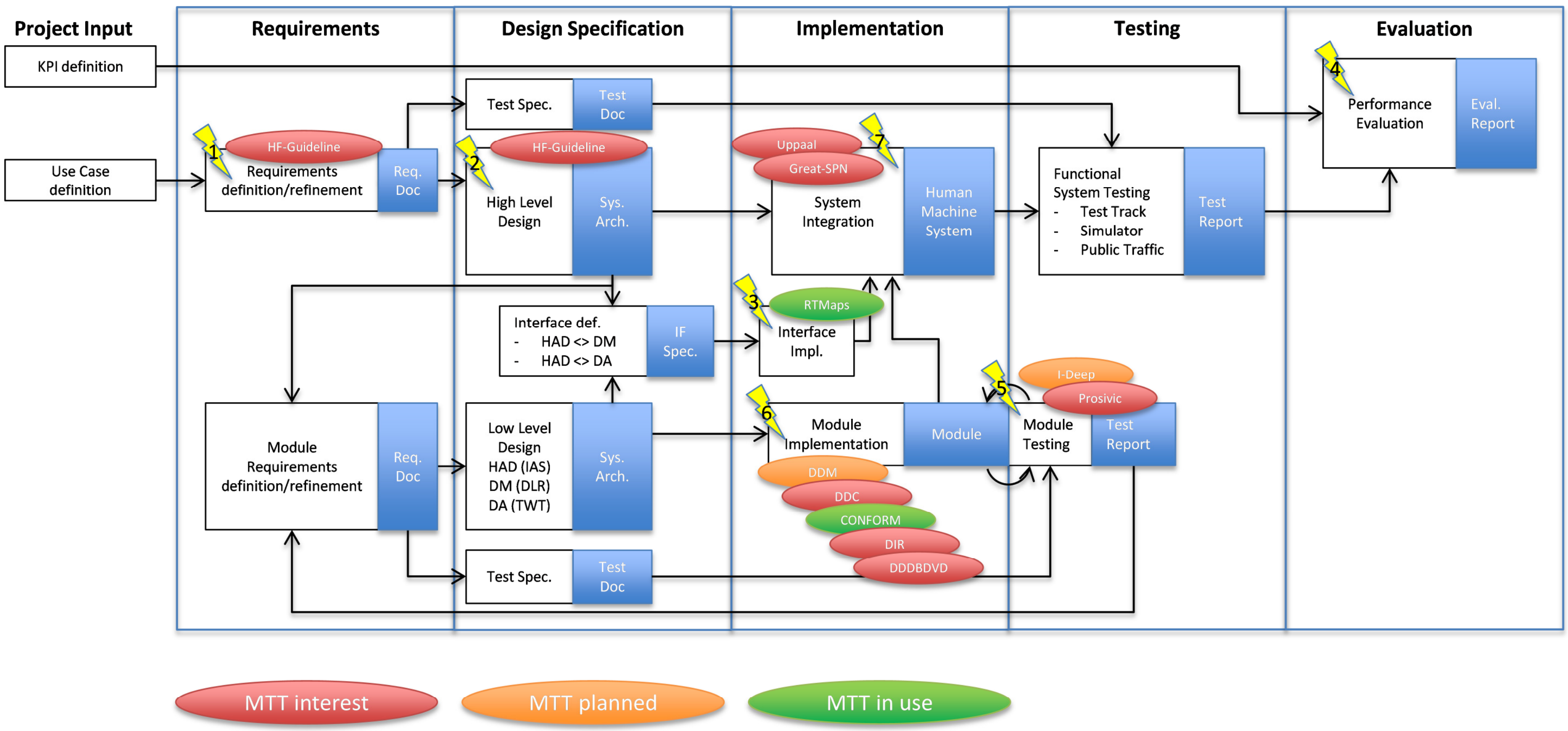

Figure 13: Development process for AdCoS Adapted Automation.

5.2.1.3 Input to the modelling process from other work packages

Table 5 provides an overview over the input to the modelling process from other WPs in terms of MTTs included in the HF-RTP. The MTTs can be divided into two categories: (1) MTTs that support the development process of the AdCoS and (2) MTTs that are part of the product which is under development in this process. These tools are labelled with PROC and PROD, respectively in the following table.

Table 5: Overview of MTTs used during the AdCoS development process.

MTT	WP	Issue
CONFORM, (DLR) ***PROD***	3	**Module Implementation** This driver model is implemented in the AdCoS to characterize the individual driving style of the human driver in real time. This information is used in the real time system to adapt the driving style of the automation according to the individual driver. The relation between manual driving style and preferred automation characteristics is determined through user studies.
RTMaps, (INT) ***PROC/PROD***	4	**Interface Implementation:** The tool used for the interface implementation is RTMaps. It has been selected from the HF-RTP, since it solves the issue on synchronization, operating system dependencies and gateway installations. RTMaps is well suited, since it provides a lot of interfaces to standard protocols, like CAN, Ethernet, etc. Most of the selected modules used within this AdCoS will be developed in C++, and RTMaps allows easy integration of C++ modules. The usage of RTMaps is expected to highly decrease the interface implementation effort.
Methods and techniques for the driver adaptive parameterization of a highly automated driving system, (DLR) ***PROC***	5	**Requirements Definition / System Design / Module Testing** This activity encompasses the empirical studies necessary to determine driver styles and design appropriate automation driving styles. Data from the experiments are used to implement the CONFORM-module.

Table 6 provides an overview over the potential input from other WPs in terms of MTTs included in the HF-RTP that are planned to be used or for which there is interest in use in the future.

Table 6: Overview of MTTs potentially used during the AdCoS development process in the future.

MTT	WP	Issue
COSMODRIVE (IFS) ***PROC***	4	**Module Testing** COSMODRIVE is indented to be used for testing purposes for the driving style classification as well as the driver distraction classification.
Driver distraction model (DDM), (TWT) ***PROD***	2	**Module Implementation** The driver distraction model will be used within the AdCoS for adaptation of the autonomous driving style. When the driver is distracted, a more "defensive" style will be selected. The model was chosen as providing an additional input for the adaptation. The implementation of the model will provide an RTMaps interface, so it can be integrated without much effort.
Driver distraction classifier (DDC), (UTO) ***PROD***	5	**Module Implementation** The driver distraction classifier could be used as a reference for the driver distraction estimation which is implemented in this AdCoS by TWT.
Uppaal (UTO) ***PROC***	5	**Module Implementation** The time automata could be of interest in this AdCoS to model the task allocation and transitioning between manual and automatic driving.
Driver intention recognition (DIR), (OFF) ***PROD***	3	**Module Implementation** The driver intention recognition could be used within the AdCoS to adapt the decision making process of the automated driving according to the preferences of the human driver. Selecting

MTT	WP	Issue
		manoeuvres that are most suitable to the estimated intention of the human driver could improve the acceptance of the human operator.
HF-Guideline, (EAD-IW-DE) ***PROC***	1	**Requirements Definition / System Design** The Human Factors Guideline could be used to define the system and all relevant aspects comprehensively and identifying potential issues in the system design at an early stage in the project. The number of iterations for designing the system could be reduced.
I-Deep, (INT) ***PROC***	4	**Module Testing:** The testing of the overall AdCoS needs to be done in simulations, since no reproducibility is given for the real vehicle. Since a simulator is available at DLR which is involved in the AdCoS development, it will be used instead of the simulators available in the HF-RTP. The tools used for testing single modules depend on the module itself. However for the functional testing of modules, I-Deep is expected to be usable, since it is a server based application which allows to run RTMaps projects with variable parameters and stores the results in a database. It is therefore expected, that I-Deep will allow an extensive testing of the AdCoS module functions.
COSMO-CIVIC, (IFS) ***PROC***	4	**Module Testing:** COSMO-CIVIC can be used together with COSMODRIVE to test core functionalities of the AdCoS. The integration is intended to be done with COSMODRIVE, Pro-SiVIC and RTMaps.
Great-SPN, (UTO) ***PROC***	2	**Module Implementation** Great SPN could be of interest in this AdCoS to model the task allocation and transitioning between manual and automatic driving.

MTT	WP	Issue
ProSivic, (CVT) ***PROC***	4	**Module Testing** ProSivic may be useful for generating simulation scenarios for offline testing. The interface to RTMaps allows using all of the AdCoS modules to be tested with simulated data.
CPM-GOMS task analysis of a Lane Change for manual and automated driving, (DLR) ****PROC****	2	**Requirements Definition / System Design** This task analysis method will be used further to understand the cognitive, perceptual and motor actions of the human driver during lane changes. It consists of a critical path analysis of the operators applied by the human driver during this task. This information will provide critical input to the human-machine-interaction strategy used to design the handover-of-controls between human and machine.
Theatre Technique for acceptance tests during AdCoS design, (DLR) ****PROC****	5	**Requirements Definition / System Design** The Theatre Technique is used to explore design alternatives for the fluent task handover between human driver and automation. Due to its Wizard-of-Oz approach, this technique will allow the designers and human factors expert to explore possible functions without the necessity of implementation.
Detection of driver distraction based on in-car measures, (TWT) ***PROD***	2	**Module Implementation** The tool can detect driver distraction using a microphone, a camera, and driving data from the CAN bus. Together with the distraction model mentioned above, a value is calculated which is an estimation of the current degree of distraction of the driver. If the value indicates a distracted driver, more defensive manoeuvers are selected.
Detection of driver	5	**Module Implementation** This module could be used either as a

MTT	WP	Issue
distraction based on data on vehicle dynamics (DDDBDVD), (UTO) *PROD*		reference for the driver distraction estimation which is implemented in this AdCoS by TWT or as an additional source of information for driver distraction.

5.2.2 AdCoS and Module Models

The overall AdCoS model is illustrated in Figure 14. The main modules of the AdCoS include

1. Machine agent: Providing the artificial intelligence for automated driving on highways.
2. Driver Model: Estimating the characteristics and intentions of the human driver.
3. Driver Analysis: Estimating the distraction of the human driver.
4. HMI: Providing situation dependent information to the driver.

The latter three modules close the loop between human and machine agent.

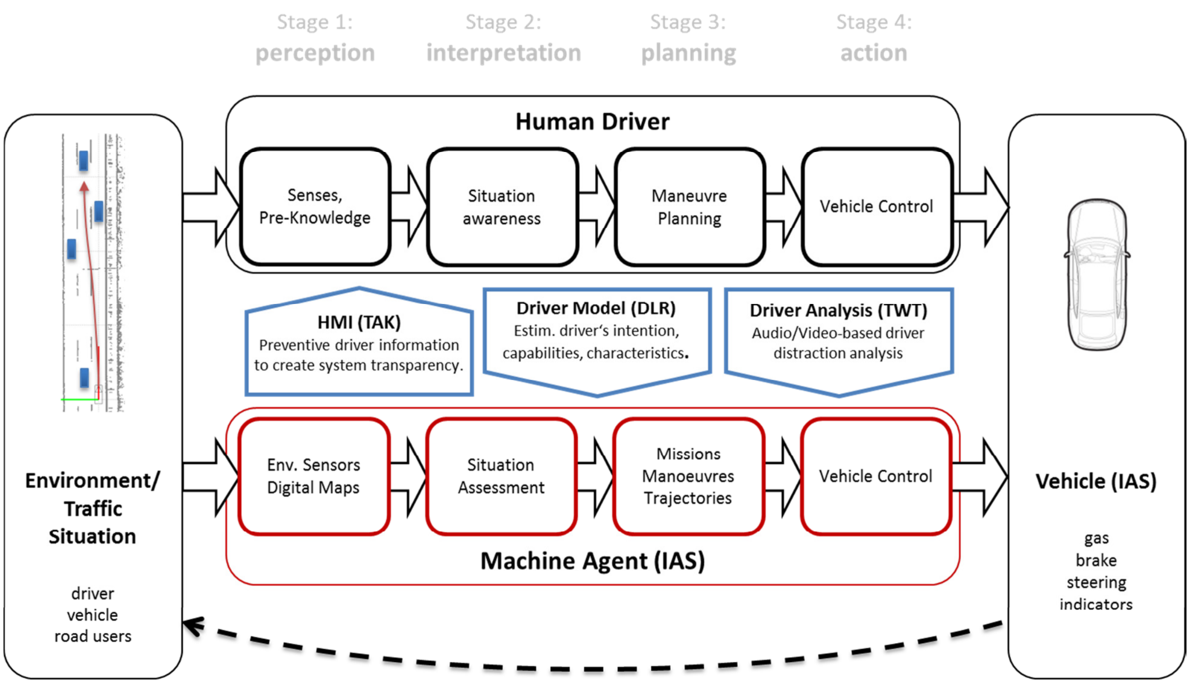

Figure 14: Adapted Automation AdCoS Model

5.2.2.1 Machine Agent for Highly Automated Driving

The model of the machine agent is a process of four sequential process steps.

1. Perception
2. Interpretation
3. Planning
4. Action

The Perception of the machine agent is provided by various sensors that are mounted around the vehicle to provide a 360 degree view of the local traffic environment. Other road users are detected and tracked by the system. The interface is a feature based representation of surrounding objects, dynamic and static infrastructure, including accurate and detailed information about position, motion, shape, uncertainties, constant object IDs over time and object classification.

Additionally a grid-based representation is provided to obtain probabilistic information about occupied and free space.

A localization algorithm provides an accurate estimate of the current position of the vehicle relative to a given map. This map is used as an additional source of information, very similar to a physical sensor. The map provides information about the road layout, legal road information as well as static background information.

Lane information is additionally estimated from the real-time sensor data.

The interpretation layer analyses the overall traffic situation. It provides a prediction of how the situation will evolve over a prediction horizon of a few seconds. It considers coupling between different road users and the host vehicle. This prediction is input to the planning of a suitable trajectory for the automated vehicle.

The planning module is the core of the machine agent. Inputs to the motion planning are

1. The current motion state of the host vehicle,
2. The current traffic situation including other road users (position, motion, classification, etc.) and the road layout (lanes, road boundaries, etc.).
3. A prediction of how the current traffic situation will evolve over time.
4. Estimated characteristics of the human driver (provided by the Driver Model by DLR)
5. Estimated level of distraction (provided by the Driver Analysis by TWT)
6. Estimated intention of the human driver, e.g. a maneuver preference.

The output of this module is the best trajectory with respect to a defined cost function and the given inputs.

The action module takes the calculated trajectory as input and controls the vehicles actuators, e.g. gas, brakes, indicators and steering wheel, accordingly. Also, this module handles the fluent transition between manual and automatic driving.

5.2.2.2 CONFORM

The MTT CONFORM is used together with the IAS test vehicle to realize a driver adaptive automated driving. Therefore CONFORM has to exchange data with the machine agent of the automated vehicle which is responsible for the situation assessment, trajectory planning and vehicle control. The MTT CONFORM was already described in detail in D3.4b [3] and D9.3 [10]. For this deliverable we will concentrate on the data flow between the IAS machine agent and CONFORM and we will provide an update to the previous specifications.

Figure 15 shows the updated data flow. Compared to the previous version we introduce a three phase's data interaction depending on if automated driving is active and depending on if we collect data to train CONFORM. The three phases are then defined as follows:

Phase 1: CONFORM offline training

In phase one experiments are conducted within WP5 (see D5.4 [7]) to **collect data for different drivers** in different manual overtaking scenarios to train CONFORM. This correlates to "automated driving active - No" and "Data Collection – Yes" in Figure 15. In this phase CONFORM only receives data and does not send data to the machine agent. The required input data can be distinguished in environment and human behavior related parameters. Both are more precisely specified in Figure 15. At the current point of development the inputs for the human behavior related parameters are not yet final decided. But CONFORM provides an interface to a csv file named "Data image specification" where system designer define:

- The inputs from the CAN data stream to be considered
- The property they want to use (moving mean, moving standard deviation, moving median, first/second derivative)
- The min/max value of each data

This can be added directly in the csv file or as shown in D3.4 [3] with support of the CONFORM GUI.

At the end of phase 1 the collected data images for the different overtaking scenarios are used to cluster different driving styles in the module "Clustering". CONFORM does this by applying standard image processing methods. These clusters will be used as input in phase 2.

Phase 2: Manual driving

In phase two **COMFORM learns the natural driving behavior of the current driver** based on manual driven overtaking scenarios. Therefore CONFORM receives the same inputs as in phase one. This correlates to "automated driving active - No" and "Data Collection – No" in Figure 15. In this phase the module "Conflict Analyzer" is additional applied to compare for each situation state the data image of the current driver with the data image clusters from phase one. The output of phase two is the classification of the driving style for each considered situation state, a trajectory corresponding to the driving style and a confidence value of the classification. These outputs are sent to the machine Agent. The machine agent analysis the trajectories and adapts the weights of the cost function for the trajectory planning to influence the behavior of the automated vehicle with regard to the classified driving style.

Phase 3: Automated driving

In phase three the automated driving mode is active. CONFORM does not receives or sends any data. Based on the output of phase two the **driving style of the automated vehicle (= automation style) is adapted** to the current driver. At the current state of development a further online learning of CONFORM and adaptation of the automation style through CONFORM is not planned. Nevertheless there will be an adaptation based on the driver state (driver distraction). This will be done by TWT.

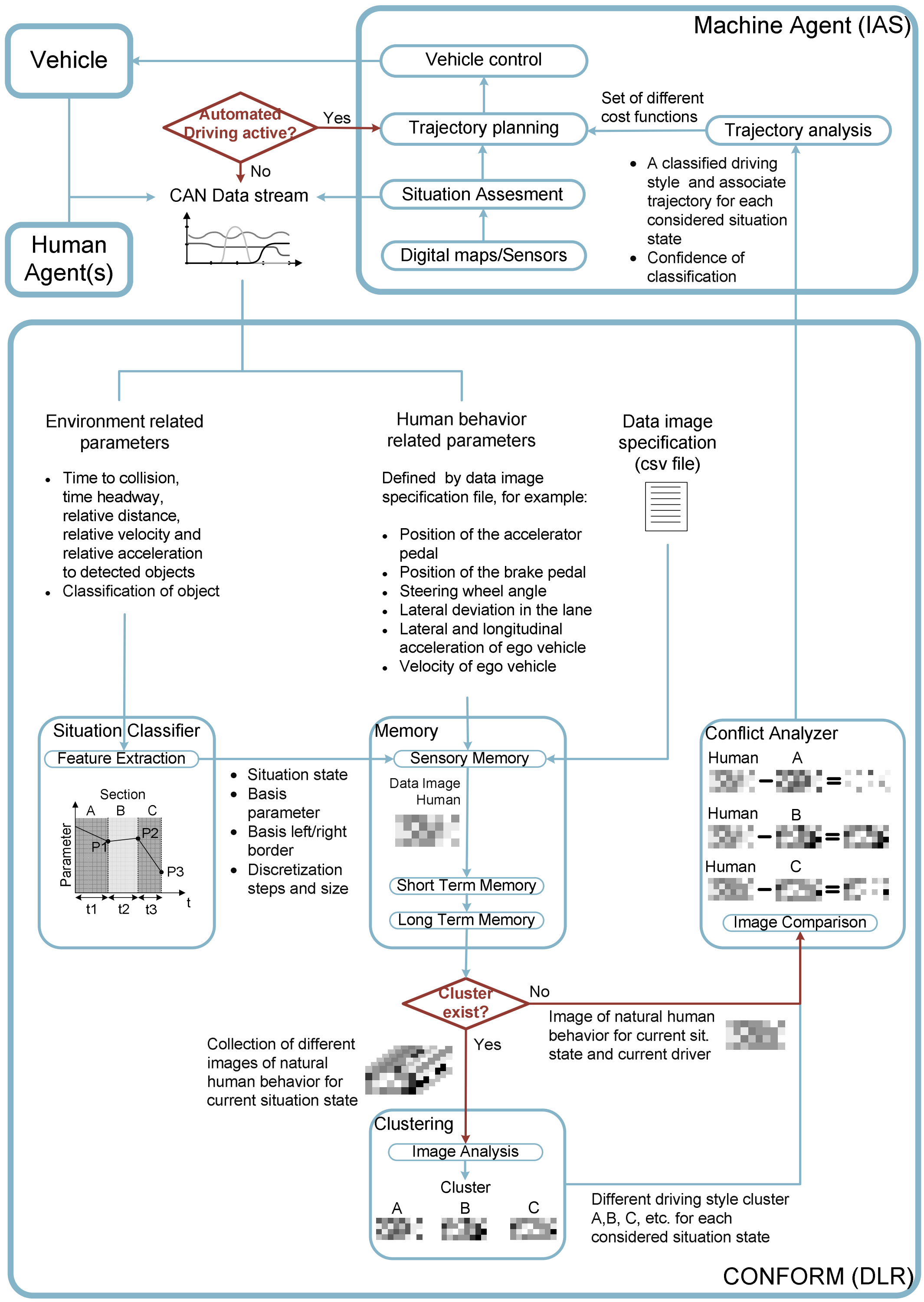

Figure 15: Information flow between CONFORM and IAS machine agent

5.2.2.3 Driver Distraction Recognition

A driver distraction model will be used within the AdCoS for adapting the autonomous driving style based on the current state of the driver. State, in this context, refers to the level of attention on the driving task. The driver's attention on the driving task may vary over time due to, for instance, drowsiness or distraction by another task. With the selected model, we focus on cognitive distraction. When the driver is distracted, a more defensive style will be selected. Besides CONFORM, this model was chosen in order to provide an additional input for adaptation based on the current state of the driver, and not only based on the long-term driving style of the driver.

Compared to other distraction models that are only based on driving parameters or visual distraction, this model has the potential to cover a broader range of distraction and to even function when a single input source is not available. In particular during autonomous driving, driving parameters cannot be used since those are not directly influenced by the driver.

To obtain relevant parameters for inferring cognitive distraction, we used the cognitive driver model described in D2.5 [2], Section 3.8, which is depicted in Figure 16:

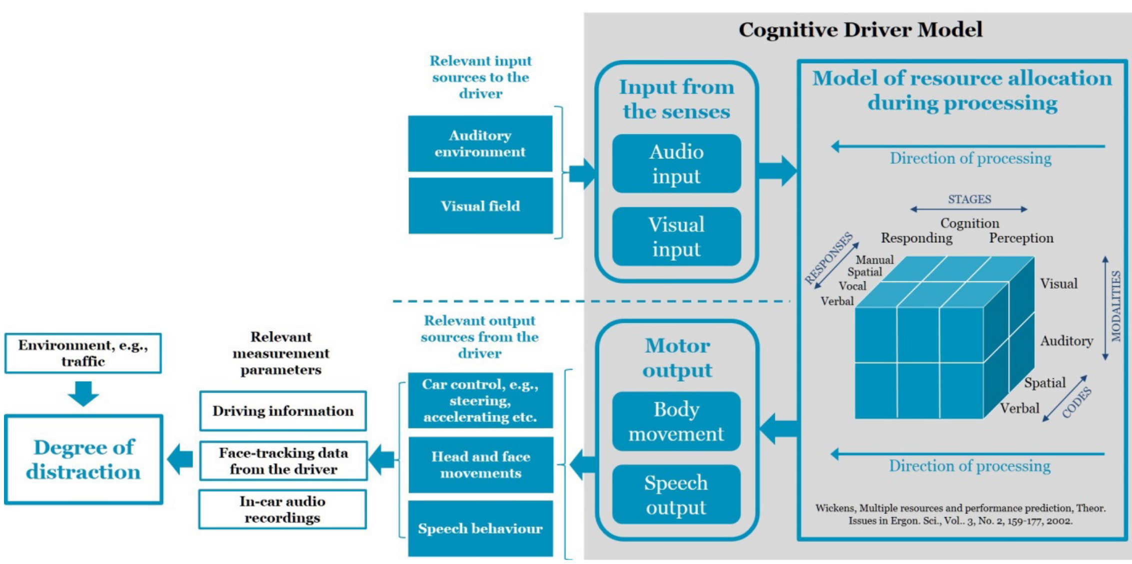

Figure 16: Cognitive Driver Model (from [2], Section 3.8)

Figure 16 shows the architectural design of a simplified cognitive distraction estimation model. We take the driver's auditory and visual perception into consideration and compute his/her distraction degree based on a resource allocation model. This model from [14] states that the more a secondary task takes up the same or similar sensory modalities (auditory vs. visual), codes (visual vs. spatial) and processing stages (perceptual, cognitive, responses), the more the secondary task leads to distraction from the primary task. The measured parameters derived from in-car audio recordings, face-tracking information of the driver, behavioural car information (e.g. driving parameters) and environmental information like the distance to the pace car to be followed will lead to conclusions about the allocation of the driver's resources and therefore enable the computation of his distraction degree.

For this distraction model, additional sensory information from the car cabin is needed in order to feed audio data (from the whole cabin) and video data (from the driver) to the model. Behavioural car information and environmental information can be obtained from other AdCoS components. A specific challenge here is that in case of autonomous driving, the model cannot use driving parameters to infer distraction but has to rely on auditory and visual input.

The estimated distraction degree output by the model then feeds into the Machine Agent for Automated Driving (Section 0) which eventually decides whether to choose a driver-specific driving style or a generic defensive driving style.

5.3 V-HCD Platform for the virtual design of MOVIDA-AdCoS

5.3.1 Description of the AdCoS

The AdCoS designed and developed by IFS is an integrative co-piloting device combining several simulated Advanced Driving Aid Systems (ADAS). All these ADAS are centrally managed in and adaptive and cooperative way, by a set of monitoring functions implemented by the MOVIDA module (for Monitoring of Visual Distraction and risks Assessment), in accordance with drivers' visual distraction status and the external situational risks (assessed from ADAS and car sensors).

To support the virtual design, prototyping and then test of this MOVIDA-AdCoS, the aim is to use in WP9 a Virtual Human Centred Design platform (so-called V-HCD) being jointly developed by IFS, CVT and INT in WP4, as an example of a tailored HF-RTP based on RTMaps software, specifically dedicated to dynamic simulations of virtual MOVIDA-AdCoS in HOLIDES (see detailed description in D4.4 [5] and D4.5 [6]).

As presented in detail in D9.4 [11], this V-HCD platform integrates 4 main modelling and simulation tools: (1) a COgnitive Simulation MOdel of the car DRIVEr (named COSMODRIVE) able to visually explore the road environment from a "virtual eye" (from COSMO-SIVIC functionalities) and to drive (2) a virtual car (simulated with Pro-SIVIC) (3) equipped with virtual ADAS (Advanced Driving Aid Systems) and the MOVIDA-AdCoS (simulated with RTMaps and Pro-SIVIC), for dynamically progressing in (4) a virtual 3-D road environment (simulated with Pro-SIVIC).

From this HF-based design approach (supported here by the COSMODRIVE simulation model), it is expected to better integrate end-users' needs (i.e. real car drivers) since the first step of the AdCoS virtual design process. Moreover, this V-HCD integrative platform will provide one of the WP9 simulation Demonstrators (i.e. a virtual human centred design platform of AdCoS), as a concrete example of a tailored HF-RTP approach applied to automotive domain.

5.3.1.1 Operational definition

The main driving situations to be supported by the AdCoS based on MOVIDA concern collision risk avoidance and lane change manoeuvre decision support, to be applied in HOLIDES on driving scenarios occurring on a two-lanes Inter-Urban Highway limited to 90 km/h (Figure 17).

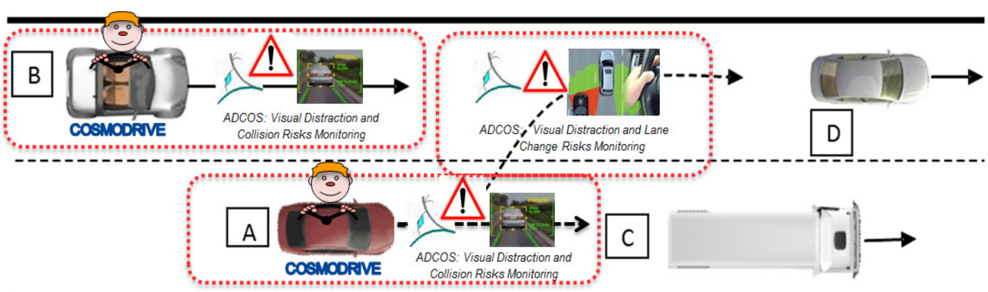

Figure 17: Driving situation supported by the MOVIDA AdCoS

In this driving context, MOVIDA is designed in order to support both drivers in Car A and in Car B. For drivers in Car A, the aim is to assist them in an adaptive way in case of critical visual distraction while approaching a slower vehicle (vehicle C), by managing the collision with it and/or by supporting a Lane Change manoeuvre. For Car B driver, this AdCoS will be mainly in charge to support collision risk in case of critical lane change of Car A, more particularly if this Car A overtaking occurs when car B driver is visually distracted.

In this traffic conditions, MOVIDA has (1) to observe and monitor COSMODRIVE's driving behaviors (in car A and/or in Car B) and (2) to diagnose risky behaviors and/or to assess situational risk (e.g. intention to implement the lane change in a critical time, or collision risk with a followed vehicle due to a visual scanning of the left mirror) and (3) to adapt the AdCoS in the right way to adequately support the different drivers, as simulated by several instances of COSMODRIVE (i.e. in car A and/or in Car B) for avoiding the accident.

5.3.1.2 Modelling techniques employed

A large set of modelling formalisms and technics are employed at both MOVIDA-AdCoS and V-HCD platform levels (from *UML* language for modelling visual scanning or driving task as *driving schema* in COSMODRIVE (Figure 18), to *state-transition graphs* for modelling and implement MOVIDA monitoring functions).

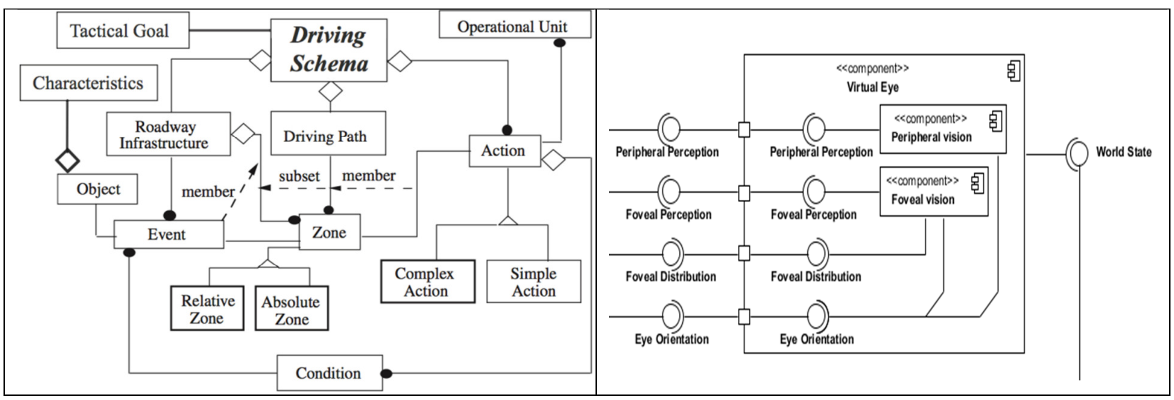

Figure 18: Example of UML modeling of the driving task (as *driving schemas*; on the left) and of COSMODRIVE's Virtual Eye (on the right)

Nevertheless, the core challenging objective (detailed in D4.4 [5] and D4.5 [6]) of this HF-based virtual design approach of AdCoS is the dynamic simulation of the functioning of this driving aid in a realistic way, and the simulation of its interaction with the end-users, as simulated through COSMODRIVE model.

At this V–HCD overall level, RTMaps is used as an integrative software to support joint simulations of complex interactions between a virtual diver, a set of vehicles (ego car and others' vehicles), several simulated ADAS supervised by the MOVIDA-AdCoS, and the Road Environment. The core formalisms and techniques used at this V-HCD level are RTMaps Diagrams (some examples are provided in a next section).

In this Virtual HCD platform, the core MTTs used are:

- COSMODRIVE and COSMO-SiVIC for simulating car drivers' perception (e.g. visual scanning of the road environment or of AdCoS HMI, for instance), cognition (like Situation Awareness or decision making) and driving behaviours (like drivers' actions on vehicle controls, or interactions with the AdCoS)
- Pro-SiVIC for the road environment simulation (3-D model), the vehicles (including the Ego-car to be driven by COSMODRIVE), and the car sensors modelling.
- RTMaps to support dynamic simulation and the data flow management (coming from sensors or other sources) and the ADAS & AdCoS prototyping.
- i-DEEP for the scenario monitoring with the management of the variation of critical parameters in the scene.
  - MOVIDA for the monitoring of driver distraction and risk assessment.

Figure 19 gives an overview of the use of these different during the 5 main phases of the design and development process of the MOVIDA-AdCoS, from the Virtual HCD Platform (i.e. Requirement, Design Specification, Implementation, Testing and Evaluation).

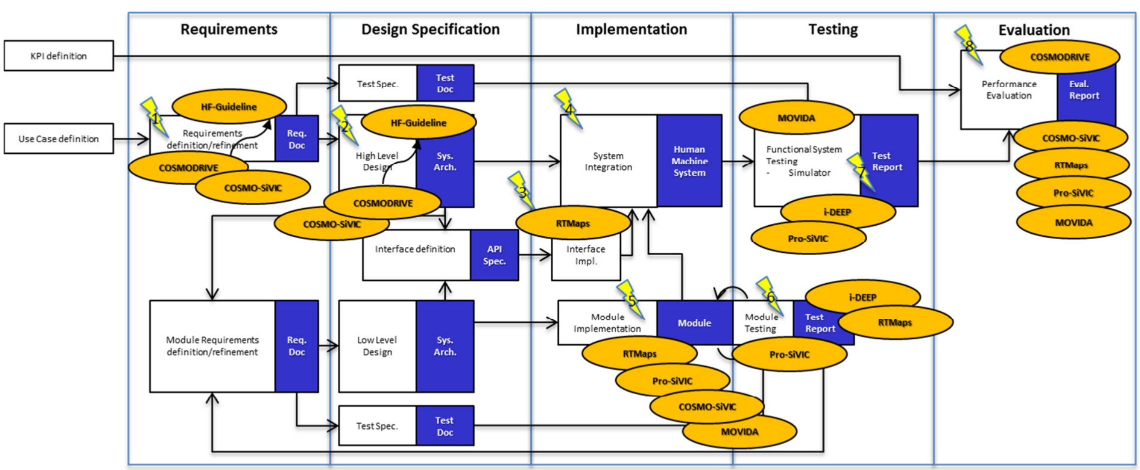

Figure 19: Development process of AdCoS Virtual HCD Platform.

5.3.1.3 Input to the modelling process from other work packages

Figure 20 presents the origin of the main MTT to be integrated in the V-HCD. COSMODRIVE simulation model is developed in WP2 and the

MOVIDA-AdCoS is developed in WP3. Modelling and simulation of dynamic interactions between these 2 first components is supported by the V-HCD platform, developed by IFS, INT and CVT in WP4, and then use in WP9 (as an example of a tailored HF-RTP; cf. D4.4 [5]) in order to virtually develop and test MOVIDA module and the MOVIDA-AdCoS as a whole.

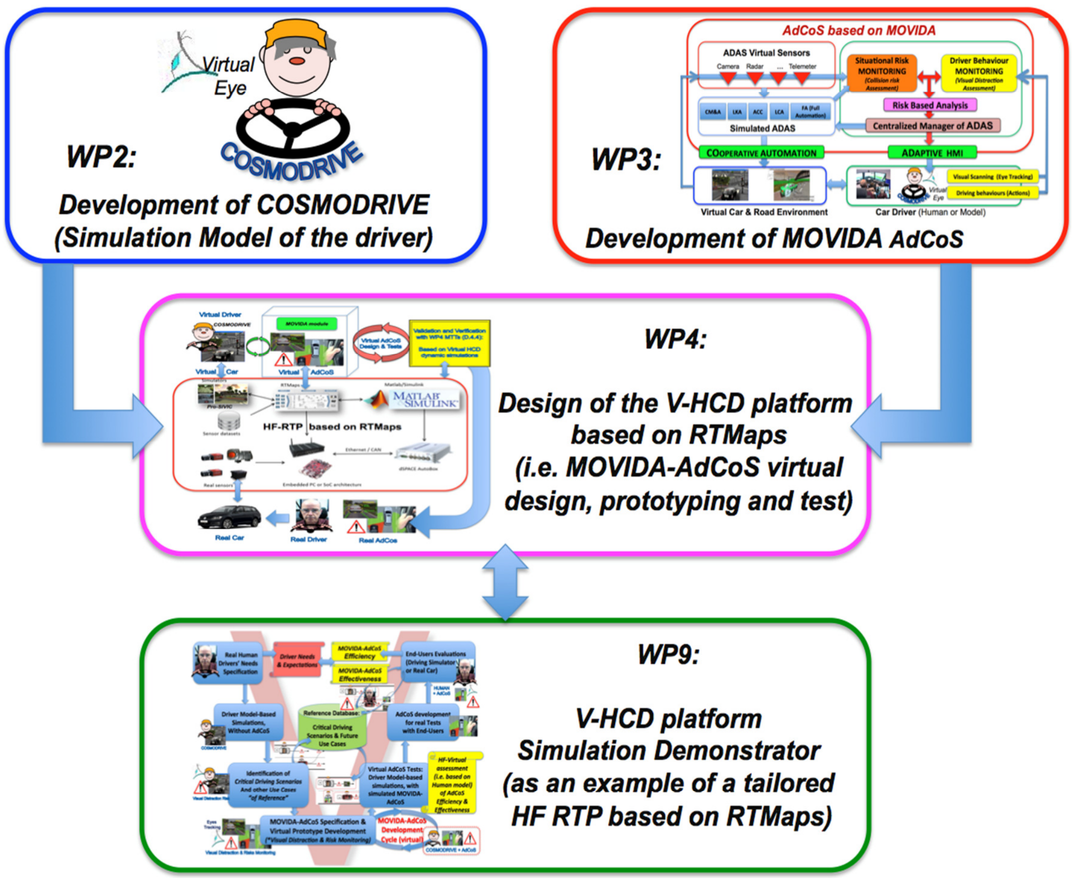

Figure 20: WPs design process of MTTs to support MOVIDA-ADCoS design

Table 7 provides an overview over the input to the modelling process from other WPs in terms of MTTs included in the HF-RTP.

Table 7: Overview of MTTs used during the AdCoS development process.

MTT	WP	Issue
COSMODRIVE & COSMO-SIVIC (IFS)	2,4	COSMODRIVE (COgnitive Simulation MOdel of the DRIVEr) and COSMO-SIVIC (integrative tool supporting COSMODRIVE's visual scanning in a Pro-SIVIC environment) provide simulation of end-users of the AdCoS based on MOVIDA. These simulations may be used at different

MTT	WP	Issue
		phases of the design process: • At the Requirement and design levels: COSMODRIVE and COSMO-SiVIC will be used to simulate drivers' performance and risk when driving without ADAS or AdCoS. Critical scenario identified from these unassisted driving simulations can be used as outputs to specify the AdCoS, in association with other HF-Guideline MTT. • At the Testing and Evaluation levels, COSMODRIVE will be used to simulate the end-user to be monitored by MOVIDA and assisted by the AdCos. COSMODRIVE visual scanning will be monitored by MOVIDA and combined with external risk of collision. Then, critical scenarios identified for unassisted driving will be applied, in order to test the effective support of the MOVIDA-AdCoS (in terms of both AdCoS efficiency and effectiveness).
Pro-SiVIC (CVT)	4	Pro-SIVIC is a software platform able to simulate embedded Sensors, infrastructure, roadside equipment, Vehicles, and mobile objects (pedestrians), in a virtual 3D Road Environment. • At the Implementation level, pro-SiVIC is used to simulate and implement virtual car sensors (physically realistic) needed for the ADAS and for MOVIDA-AdCoS. • At the Testing level, Pro-SiVIC is one of the MTTs used to generate references and ground truth to test and to evaluate some of the modules implemented in the implementation stage of the development process. • At the Evaluation level, Pro-SiVIC will be used as a part of the integrated AdCoS and provides measurements and references in order to test the functional system.
MOVIDA (IFS)	2	MOVIDA (*MOnitoring of VIsual Distraction & risks*

MTT	WP	Issue
		Assessment) is the MTT supporting algorithms in charge to centrally manage ADAS and to interact with the car Driver in an Adaptive and Cooperative way.
RTMaps (INT)	4	RTMaps is a software platform allowing to easily and efficiently interconnect the data streams of the different tools and models, such as simulators, sensors and actuators, or HMI, and can integrate data processing algorithms, also with capacities for synchronized recording and playback of any kind of streams. These functionalities provide MTT : • At the implementation level, RTMaps will support the virtual development and the implementation of the simulated ADAS, their interconnections, and their centralized management by MOVIDA. In addition, RTMaps is also used as the core integrative software supporting the V-HCD platform as a whole. • At the Testing and Evaluation levels, RTMaps (combined with Pr-SIVIC, COSMODRIVE and COSMO-SIVIC) will be used to support the dynamic simulation of MOVIDA-AdCoS functioning (from car sensors to full AdCoS simulation). In addition, this software will also support dynamic simulation of the MOVIDA-AdCoS use by / interaction with a human driver.
iDeep (INT)	4	In order to test and to evaluate either the overall AdCoS or a specific component of the AdCoS, it is relevant to use software which allows to monitor a great set of scenario and which provide the variation of critical parameters in the scene. This MTT is iDeep. This MTT needs to be used with a simulation platform, since no reproducibility is given for the real vehicle. iDeep is perfectly adapted to work with the Virtual HCD platform which is involved in the AdCoS

MTT	WP	Issue
		development. The MTT usable for the testing of a single module depends of the module itself. However for the functional testing of modules, I-Deep is expected to be efficient, since it is a server based application which allows to run RTMaps, Pro-SiVIC, CSOMODRIVE and Cosmo-SiVIC MTTs with variable parameters and stores the results in a database. It is therefore expected, that I-Deep will allow an extensive testing of the AdCoS module functions.

Table 8 provides an overview over the potential input from other WPs in terms of MTTs included in the HF-RTP that are planned to be used or for which there is interest in use in the future.

Table 8: Overview of MTTs potentially used during the AdCoS development process in the future.

MTT	WP	Issue
HF-Guideline, (EAD-IW-DE)	1	The Human Factors Guideline could be used to define the system and all relevant aspects comprehensively and identifying potential issues in the system design at an early stage in the project. The number of iterations for designing the system could be reduced.
ERG		Eye tracking technics should be used by MOVIDA module to monitor drivers' visual scanning.
GreatSPN-MPD, Driver distr. classif. (UTO)	5	Liable to be used to support interactions with teams involved in CRF demonstrator, regarding drivers' distraction and risk issues
Djnn (ENA)	2	Liable to be used to support HMI verification of IFS AdCoS based on MOVIDA
Bad MoB & Driver Intention Rec. (OFF)	2,3	Liable to be used in association with COSMODRIVE model to support interactions with teams involved in CRF demonstrator, regarding drivers' distraction and situational risk issues.
LEA, APA & AMAS (EAD-F)	3	Liable to be used to support the design and development of drivers Monitoring Functions to

MTT	WP	Issue
		be integrated in MOVIDA module
Analysis of cognitive and communication process. (SNV)	5	Liable to be used to support interaction with teams working on CRF demonstrator, more particularly regarding methodological issues, or empirical/simulated data sharing.

5.3.2 AdCoS and Module Models

As presented in introduction, the MOVIDA based AdCoS is an integrative co-piloting system combining several simulated ADAS. The core ADAS sub-systems integrated in this AdCoS are (c.f., Figure 21) a Collision Mitigation and Avoidance (CM&A) system, a Lane Keeping Assistant (LKA), a Lane Change Assistant (LCA), an Adaptive Cruise Control (ACC). In addition, Full Automation (FA) may be also simulated by combing the preceding ADAS. All these ADAS are centrally managed by MOVIDA module, according to drivers' visual distraction status and the external situational risks (assessed from ADAS and car sensors).

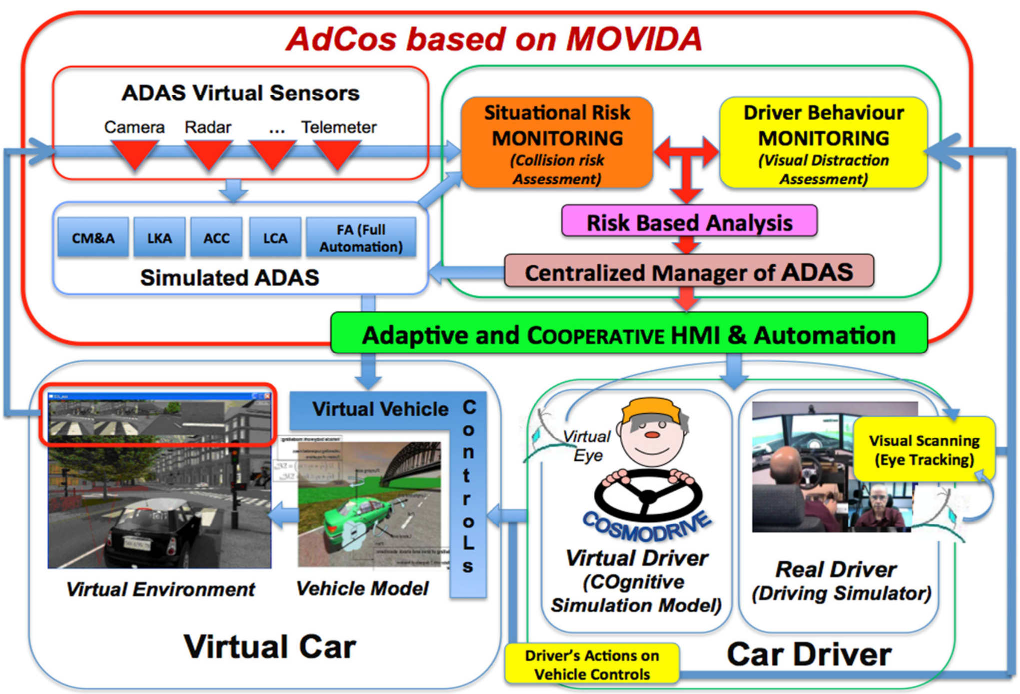

Figure 21: Functional architecture of the AdCoS based on MOVIDA

All these ADAS and the global AdCoS based on MOVIDA as a whole, are interfaced and/or simulated with RTMaps platform and Pro-SIVIC software. In order to run the different ADAS and AdCoS versions, the designer needs either to replay sensors databases (from real or virtual sensors), or may use real time sensors data flow coming from embedded sensors, or virtual embedded sensors available in the pro-SiVIC platform.

Regarding Human-Machine Interaction modalities issues, the MOVIDA-AdCoS is liable to interact with the driver from 2 main modalities: warning and vehicle control taking (via partial or full automation). Adaptive and Cooperative abilities in this AdCoS are supported by MOVIDA monitoring functions in charge to supervise the drivers' activity (by assessing their behaviours and their visual distraction state) and, from the other side, to evaluate the risk of accident in the current traffic situation (forward collision risk of Car A with the slow vehicle C ahead, or rear collision risk with rear vehicle B. From these MOVIDA monitoring functions, Risk-based analysis algorithms and a Centralized Manager of ADAS modules will be implemented, in order to provide an adaptive and cooperative support system (based on warning or on vehicle control taking), specifically adapted to the current driver needs, in accordance with both their visual distraction state and the situational risks.

At the V–HCD overall level, as previously explained, RTMaps is used as an integrative software to support dynamic simulations of complex interactions between a virtual driver (simulated with COSMODRIVE), a virtual car equipped with a set of simulated ADAS supervised by MOVIDA (implemented with Pro-SIVIC and COSMO-SIVIC), and the Road Environment (simulated with Pro-SIVIC). The core formalisms and techniques used at the V-HCD level are RTMaps Diagrams (Figure 22).

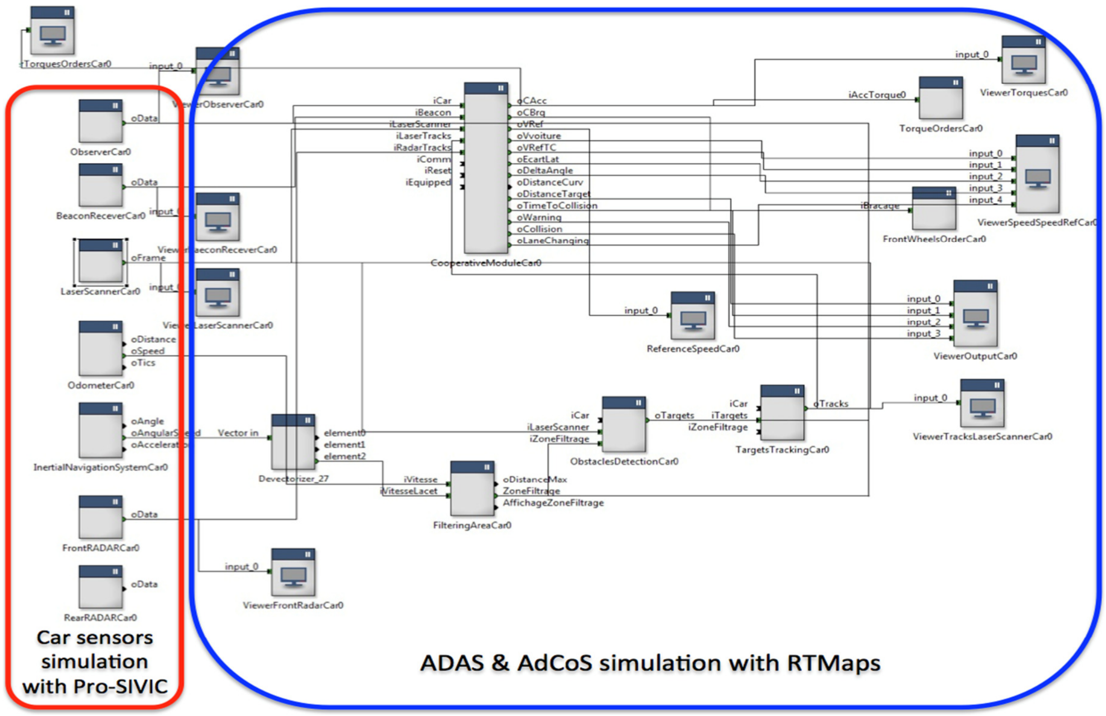

Figure 22: RTMaps diagrams for ADAS & AdCoS simulation and prototyping

In addition, RTMaps diagrams are also used for modelling, to simulate and then to dynamically implement human driver's interactions (a real driver or a simulated driver with COSMODRIVE) with the Pro-SIVIC virtual car, potentially equipped with ADAS and AdCoS (Figure 23).

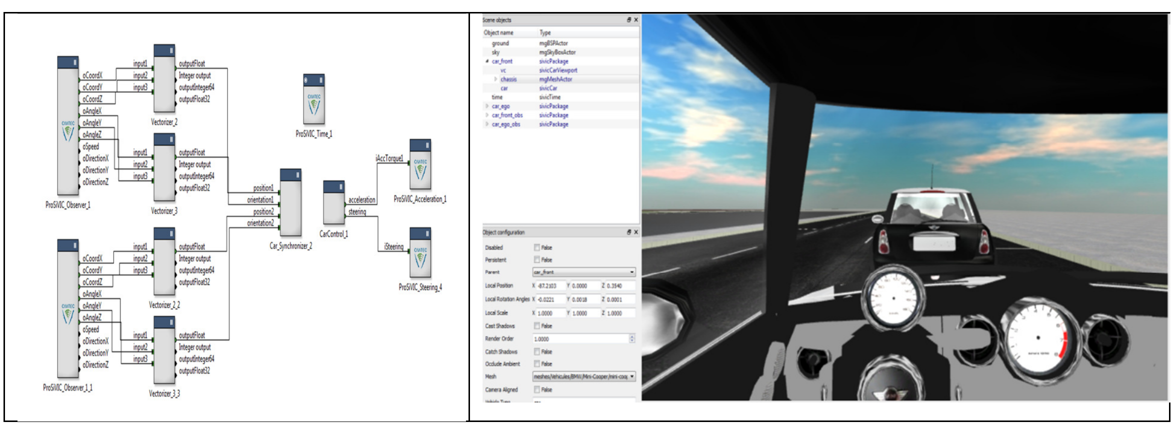

Figure 23: RTMaps diagram supporting COSMODRIVE and Pro-SiVIC car interactions (on the left) and view of a simulation result via COSMODRIVE piloting a virtual car (on the right)

5.4 AdCoS Adaptive HMI

5.4.1 Description of the AdCoS

As described in preceding deliverables of WP9 (Deliverables D9.1 [8], D9.2 [9], D9.3 [10] and D9.4 [11]), the TAKATA AdCoS will be an HMI developed for the WP9 scenario and use cases derived thereof. Some of these use cases comprises automatic driving whereby this automation consists of a combination of longitudinal and lateral control in different levels of automation.

5.4.1.1 Operational definition

Driving is a task that is conducted in a dynamic environment. Besides the environment, the human driver itself is dynamic in the sense that its cognitive and physiological resources change as do its emotions and motivations. This combination suggests that adapting the task to the current state of the vehicle-driver-environment system is a promising way to achieve an optimum task outcome. Such changes in turn require that they are fed back to the driver in order to keep him/her in the loop. This is – amongst others - achieved via an adaptive HMI.
The basis for the adaptation will be driver distraction, both visual and cognitive. The classification of distraction will be achieved by MTTs provided by project partners (see D9.3 [10] for details).
One of the challenges of HMI development is to foresee the acceptance of and the consequences for the user of different design solutions. In order for the process to be fast, these variables must be foreseen as early as possible in the design process. Human behaviour models provide an appropriate means to test user acceptance and foresee negative consequences. This will be achieved by applying CASCaS.

5.4.1.2 Modelling techniques employed

To model the effect of the HMI, different techniques are employed. Beginning with the development process, expert evaluations are used. These expert evaluations can later be supported by the application of the HF-Guidelines. In later steps CASCaS will be applied to foresee the effects of different designs.

Figure 24 gives an overview of the different MTTs (to be) used during the different steps of development process of the AdCoS Adaptive HMI, as described in D9.4 [11]. The different MTTs have been used to create actual models used during the development process.

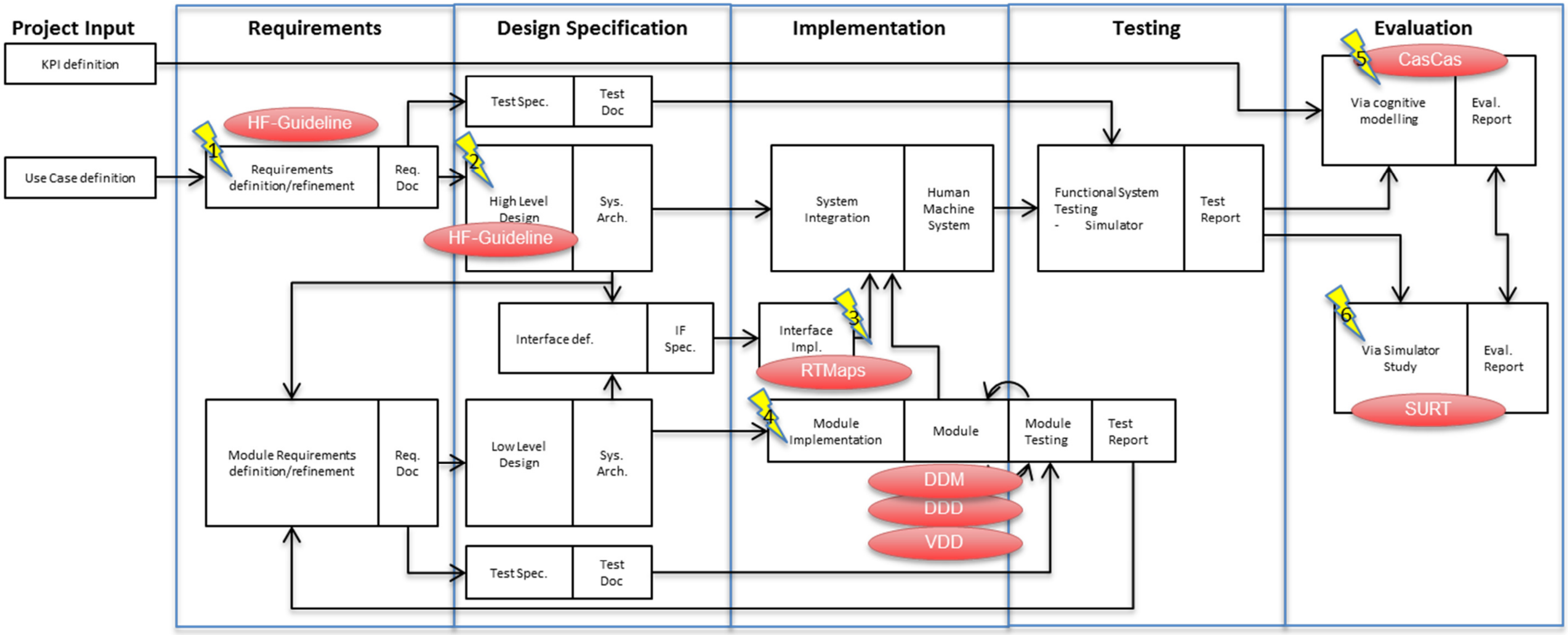

Figure 24: Development process of AdCoS Adaptive HMI.

5.4.1.3 Input to the modelling process from other work packages

Table 9 provides an overview over the input to the modelling process from other WPs in terms of MTTs included in the HF-RTP.

Table 9: Overview of MTTs used during the AdCoS development process.

MTT	WP	Issue

Table 10 provides an overview over the potential input from other WPs in terms of MTTs included in the HF-RTP that are planned to be used or for which there is interest in use in the future.

Table 10: Overview of MTTs potentially used during the AdCoS development process in the future.

MTT	WP	Issue
HF-Guideline, (EAD-IW-DE)	1	**Requirements Definition / System Design** The Human Factors Guideline could be used to define the system and all relevant aspects comprehensively and identifying potential issues in the system design at an early stage in the project. The number of iterations for designing the system could be reduced.
SuRT (DLR)	5	**Evaluation** The SuRT can help to reliable induce visual distraction. It is therefore an important part in the evaluation of the AdCoS.
Visual Distraction Detection (n.n.)		**Module** Visual Distraction Detection is needed to reliably detect visual distraction.
Driver distraction model (DDM), (TWT)	2	**Module** The driver distraction model will be used within the AdCoS for adaptation of the HMI. The implementation of the model will provide an RTMaps interface, so it can be integrated without much effort.
Detection of Driver Distraction (DDD) (TWT)	2	**Module** The detection of (cognitive) driver distraction is a prerequisite for the AdCoS. Without DDD the AdCoS would not be possible.
CASCaS (OFF)	2	**Evaluation** CASCaS is used to reduce the simulator studies with real participants and speed up the development process of the HMI

5.4.2 AdCoS and Module Models

The AdCoS consists of different parts: the instrument cluster, the center screen with the entertainment system and the mirrors that are used as displays. The center screen that is originally intended to house the entertainment system is used both to present the distraction task (SURT) and the warning that is displayed to inform the driver of imminent danger. Its aim is to prevent the driver from further engaging in the potential distraction task and to redirect its attention to the road. The instrument cluster provides the driver with information and warnings regarding its position on the road and its distance to other traffic participants. Finally, the mirrors are used to inform (when no lane change is intended) and warn (in the case of an intended lane change) of overtaking vehicles in the left lane.
This HMI concept is based on the MTTs provided by the partners as described in preceding deliverables. Additional MTTs are used to reliably distract the driver and to model the effect of this driver distraction.

5.4.2.1 Driver Distraction Recognition <TWT>

Since the Adaptive HMI AdCoS primarily focuses on the evaluation of HMI adaptation techniques, it is employed mainly during the system evaluation step of the development process. Driver distraction plays an essential role during this phase in order to determine whether proposed changes to the HMI increase the actual distraction of the driver or whether they help to decrease it. In addition, the driver distraction delivers an additional input to new HMI adaptation techniques: If the driver is concentrated on the driving task, non-intrusive indications of events can be displayed in the general dash-board. If, however, the driver is distracted such indications may escape the current attention of the driver and should be reinforced by stronger signals (e.g., acoustic feedback), depending on their importance.

With the selected distraction model, we focus on cognitive distraction. To obtain relevant parameters for inferring cognitive distraction, we used the cognitive driver model described in D2.5 [2], Section 3.8, which is depicted in Figure 16. For more information on the architectural design, please see Section 5.2.2.3.

The model gives as output the estimated distraction degree of the driver. This estimated distraction degree is recorded during the whole simulation which can then be evaluated for specific HMI events. In addition, the distraction level is provided in real-time to simulator components as needed. It describes the estimated distraction based on input parameters collected during the last 3-10 seconds.

6 Conclusions

We will now attempt to provide a general conclusion concerning the use of MTTs provided by the HF-RTP for AdCoS modelling in the automotive domain thus far. In contrast to other application WPs, where MTTs are primarily directly used by the AdCoS owners during the design phases, WP9 heavily focusses on the implementation and integration of MTTs that intent to provide functionality to the AdCoS itself, i.e. as components of an AdCoS or AdCoS modules, e.g., in terms of adaptive HMIs, assistance functionalities, or as a means for context assessment. These MTTs are not used by the AdCoS owners in the traditional sense; instead the different AdCoS components are developed simultaneously by the respective MTT owners. Consequentially, the primary focus had to be laid on the development and establishment of a common unified framework enabling a fast and robust integration of the different AdCoS modules into the overall AdCoS.

For all AdCoS in the automotive domain, RTMaps was selected as the primary MTT to establish a common framework for development and integration. As such, the different MTTs were implemented as RTMaps component and functional AdCoS modules were modelled as RTMaps diagrams, representing functional flow block diagrams. By connecting these different modules, RTMaps constitutes a simulation environment to run the whole system, where the different blocks representing the MTTs can operate in real-time and online. This is perfectly in line with the original requirements for such a type of framework and can be interpreted as a successful realization of the MBD approach promoted by HoliDes.

Throughout the development, RTMaps allowed to easily adapt the interfaces of the different AdCoS components. Under this point of view, the framework served its purpose and greatly helped developers and engineers with the overall AdCoS development. Furthermore, the available MTT RTMaps has already reached a level of maturity that allows an efficient integration in the overall AdCoS development process.

As apparent from this document, many AdCoS modules rely on the availability of experimental data, both for development (e.g., by means of machine-learning) or to perform model-based analysis. RTMaps reduced time and effort for the synchronization of multimodal data recording and collection during experiments. Providing the data via RTMaps made it possible to immediately exchange, assess and visualize the test data among partners. During the next phases of HoliDes, the common framework will be prepared to perform model-based analysis for isolated AdCoS modules and the integrated AdCoS as a whole. As an alternative to model-based analysis based on experimental data, the V-HCD Platform for the virtual design of MOVIDA-AdCoS provides a virtual simulation environment that can be integrated in the overall modelling process.

As a general conclusion, we denote that the availability and utilization of a unified framework for data acquisition, exchange, and modelling provides a great benefit to all partners.

7 References

[1] HoliDes consortium (2015). D2.4 – Modelling Techniques and Tools V1.0. HoliDes deliverable.

[2] HoliDes consortium (2015). D2.5 – Modelling Techniques and Tools V1.5. HoliDes deliverable.

[3] HoliDes consortium (2014). D3.4 – Techniques and Tools for Adaptation Vs1.0 incl. Handbooks and Requirements Analysis Update. HoliDes deliverable.

[4] HoliDes consortium (2015). D3.5 – Techniques and Tools for Adaptation Vs1.5 incl. Handbooks and Requirements Analysis Update. HoliDes deliverable.

[5] HoliDes consortium (2014). D4.4 – Techniques and Tools for Model-based Analysis Vs1.0 incl. Handbooks and Requirements Analysis Update. HoliDes deliverable.

[6] HoliDes consortium (2015). D4.5 – Techniques and Tools for Model-based Analysis Vs1.5 incl. Handbooks and Requirements Analysis Update. HoliDes deliverable.

[7] HoliDes consortium (2015). D5.4 – Techniques and Tools for Empirical Analysis Vs1.5 incl. Handbooks and Requirements Analysis Update. HoliDes deliverable.

[8] HoliDes consortium (2014). D9.1 – Requirements Definition for the HF-RTP, Methodology and Techniques and Tools from an Automotive Perspective. HoliDes deliverable.

[9] HoliDes consortium (2014). D9.2 – Tailored HF-RTP and Methodology Vs0.5 for the Automotive Domain. HoliDes deliverable.

[10] HoliDes consortium (2015). D9.3 – Requirements & Specification & first Modelling for the Automotive AdCoS and HF-RTP Requirements Definition Update (Feedback). HoliDes deliverable.

[11] HoliDes consortium (2015). D9.4 – Tailored HF-RTP and Methodology Vs1.0 for the Automotive Domain. HoliDes deliverable.

[12] Eilers, M., and Möbus, C. (2014). Discriminative Learning of Relevant Percepts for a Bayesian Autonomous Driver Model. In: Publications of the Sixth International Conference on Advanced Cognitive Technologies and Applications (COGNITIVE 2014), pp. 19-25, May 2014, Venice, Italy.

[13] Koo, J, Kwac, J, Ju, W, Steinert, M, Leifer, L, and Nass, C (2014). Why did my car just do that? Explaining semi-autonomous driving actions to improve driver understanding, trust and performance. In: International Journal of Interactive Design and Manufacturing. Springer.

[14] Wickens, C.D. (2002). Multiple resources and performance prediction. In: Theor. Issues in Ergon. Sci., Vol. 3, No. 2, pp. 159-177.

[15] Vijayagopal, R., Shidore, N., Halbach, S., Michaels, L., and Rousseau, A. P. (2010): Automated model based design process to evaluate advanced component technologies. SAE Technical Paper, No. 2010-01-0936.