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

D9.2 Tailored HF-RTP and Methodology Vs0.5 for the Automotive Domain

Project Number:	332933
Classification:	Public
Work Package(s):	WP9.1
Milestone:	M2
Document Version:	V1.0
Issue Date:	11.09.2014
Document Timescale:	Project Start Date: October 1, 2013
Start of the Document:	Month 9
Final version due:	Month 10
Deliverable Overview:	**Main document:** D9.2 Tailored HF-RTP and Methodology Vs0.5 for the Automotive Domain
Compiled by:	Sara Saleri - REL
Authors:	Leandro Guidotti- REL , Sara Saleri – REL Fabio Tango – CRF, Bellet Thierry – IFS Martin Krähling – IAS, Nico Holst – IAS Gert Weller – TAK, Mark Eilers – OFF Stefan Griesche – DLR, Svenja Borchers – TWT, Cristóbal Curio – TWT, Jordi Fonoll Masalias – ATOS
Reviewers:	Denis Martin – TWT David Käthner – DLR
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
23.06.14	Draft structure of deliverable	Sara Saleri – REL
04.07.14	First draft of deliverable	Sara Saleri – REL
20.08.14	Second draft of deliverable with contributions from IFS, IAS, TAK, OFF, DLR, TWT, ATOS	Sara Saleri, Leandro Guidotti – RE:Lab
22.08.14	Edited the document, contributions	Svenja Borchers, Cristóbal Curio – TWT
26.08.14	Corrections and comments	David Käthner – DLR
27.08.14	Comments and contributions	Mark Eilers – OFF
28.08.2014	Comments and contributions	Fabio Tango – CRF
09.09.2014	Third version of deliverable	Sara Saleri, Leandro Guidotti – REL
12.09.2014	Definitive version of deliverable, including contributions from Denis Martin (TWT)	Sara Saleri – REL

Table of Contents

List of Acronyms

ACC = Adaptive Cruise Control

ADAS = Advanced Driving Assistance Systems

AdCoS = Adaptive Cooperative Human-Machine Systems

AUT = Automotive

CAS = Collision Avoidance Systems

COSMODRIVE = COgnitive Simulation MOdel of the DRIVEr

DAS = Driving Assistance Systems

EV = Ego Vehicle

FCW(S) = Forward Collision Warning (System)

HF = Human Factors

HF-RTP = Human Factors Reference Technology Platform

HMI = Human Machine Interaction

HMS = Human Machine Systems

HoliDes = Holistic Human Factors Design of Adaptive Cooperative Human-Machine Systems

LCA = Lane Change Assistant

LKAS = Lane Keeping Assistance

MOVIDA = Monitoring of Visual Distraction and risks Assessment

MTTs = Methods and Techniques

PADAS = Partially Autonomous Driving Assistance Systems

RTP = Reference Technology Platform

UC = Use Cases

V-HCD (platform) = Virtual Human Centred Design (platform)

WP = Work Package

Executive Summary

The following document describes the process of adaptation of the HF-RTP, which is being developed in the HoliDes project (WP1), to the Automotive domain, with a special focus on the description of the AdCoS and tool chains developed by the partners. The HoliDes project is a European Union Project, which aims at the development and qualification of Adaptive Cooperative Human-Machine Systems (AdCoS). The methodology is described as well as the used terminology.

1 Introduction

This deliverable describes how the HF-RTP methodology Vs0.5 and the HF-RTP, which are being developed in WP1, are adapted to the Automotive domain. In particular, it focuses on the process of development of an AdCoS and on the tool chains elaborated for each test-vehicle and driving simulator.

1.1 D9.2 in the context of HoliDes project

HoliDes is an European Project which addresses development and qualification of Adaptive and Cooperative Human-Machine Systems (AdCoS), where many humans and many machines act together, cooperatively, in a highly adaptive way.
Currently, a lack of adequate means of compliance with human factors and safety regulations, especially in the health, aeronautics, control rooms, and automotive market, shows the huge potential that an AdCoS could have on increasing overall safety. In order to improve this situation and unleash this potential, HoliDes is researching affordable means of compliance which enable to formalize adaptation strategies on global many humans – many machines levels and local HMI levels in a coordinated way. This is achieved by developing techniques and tools and integrating them in a Human Factors Reference Technology Platform (HF-RTP) to foster interoperability and to support human factors along the whole engineering life-cycle. All techniques and tools are then used to develop and qualify AdCoS in four domains: Health, Aeronautics, Control Room and Automotive.
Work Package (WP) 9 aims at the development and qualification of Adaptive Cooperative Human-Machine Systems (AdCoS) in the Automotive domain, using the tailored HF-RTP and methodology developed in WP1 to demonstrate the added value for industrial engineering processes in terms of reduced cost and less needed development cycles.
Deliverable D9.2 is the first step in the HF-RTP tailoring to the automotive domain, since it defines the specific application of the first version of the HF-RTP and Methodology (version 0.5) to this domain, according to the work done in WP1 (see especially D1.3).

1.2 Structure of the document

This document is divided into 5 chapters. After introducing the deliverable in Chapter 1, Chapter 2 summarizes the starting points for the process of tailoring the HF-RTP to the Automotive domain: the HF-RTP requirements, the scenarios and use cases. Chapter 3 describes the HF-RTP. Chapter 4 describes the tailoring process, focusing on the tools and techniques applied in the Automotive context. Chapter 5 draws some conclusions, starting from the work done so far.

2 Tailoring HF-RTP to the Automotive Domain: starting points

2.1 HF-RTP requirements in the Automotive Domain

The requirements in WP9 are focused mainly on functional requirements, although some non-functional requirements have been added also.
As the HoliDes project is mainly focused on the human factors in the development process, requirements deal with different constructs such as driver distraction, mental workload, multimodal information display, situation and mode awareness, trust in automation, and of course adaptive automation and their integration into development requirements.
Technically, the requirements deal with different systems and sensors. A complete list can be found in Annex II of D9.1.

2.2 Scenarios and Use Cases in the Automotive Domain

The main Use Case (UC) for the Automotive domain concerns a complete overtaking manoeuvre with 4 traffic participants (see schematic display in Figure 1).

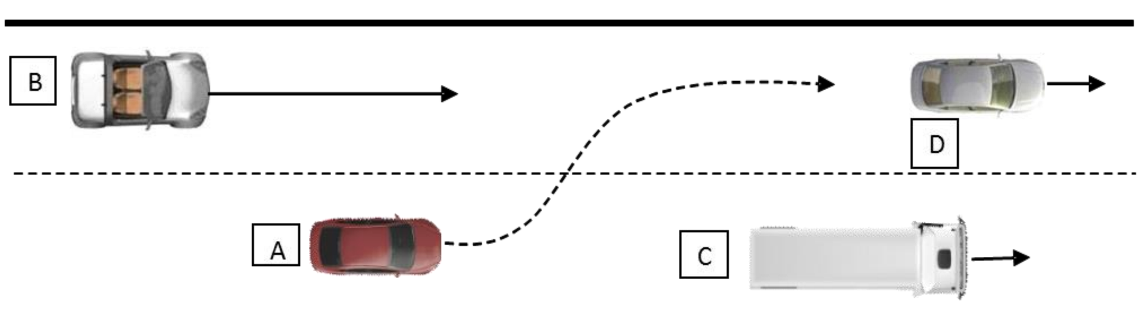


Figure 1 - Use Case.

The AdCoS here consist of four cars with machine agents (e.g. ADAS/PADAS) and human agents (drivers) in a two/three lanes road with the same driving direction. In particular, Car A (coloured by red in figure) wants to change the

lane to overtake the truck (C). During this manoeuvre a collision against car B has to be avoided.
Several situational variables can be specified for this use case in order to fix situation as much as possible and avoid variables not managed.
The potential risky behaviour is that the driver on vehicle A is not aware of the incoming vehicle in the adjacent lane and thus a collision with agent B is possible, especially if Ego Vehicle (EV, agent A) changes lanes suddenly.
Then, of course, many variations are possible, depending on different weather conditions, internal and external scenarios and types of road. There can be different conditions (e.g. extra-urban/motorway) and situations (e.g. middle traffic density, sunny day, with clear and good visibility). A pre-condition is the respect of the traffic laws.
Current PADAS work without mutual interaction and adaptation. In HoliDes these systems have to be managed as a "unique supporting system" which adapts to the behaviour of the different agents.
Agents have to interact: cooperation or competition (due to the limited and sharable resources) have to mutually understand their intentions and goals, in order to increase safety and traffic efficiency. Thereby, the system should support the human when needed.
The development of adaptive harmonized systems addresses a twofold aspect to:

- Generate warnings/advice.
- Shift to longitudinal and lateral control of the car from human to machine (and back) according to the capacity, load and intentions of driver (LOA).

In the AUT domain, AdCoS will be developed and qualified on test-vehicles provided by CRF and IAS, as well as on driving simulators provided by DLR, IFS, REL. Table 1 describes them in detail.

Demonstrator	Partners involved	Role of partner
Vehicle 1 - Frontal Collision scenario	**CRF**	Demonstrator owner; AdCoS implementation; support to the distraction classifier, driver's intention and co-pilot development; evaluation phase of the demonstrator with AdCoS.
	REL	HMI design and implementation.
	SNV	Test methodology for distraction

		data gathering; test methodology for AdCoS evaluation.
	OFF	Research Partner WP2 and WP3, Driver Intention Recognition module developer
	UTO	Integration and implementation of driver's distraction classification algorithms.
	INT	Support in using RT-MAPS (and PRO-CIVIC).
	TWT	Driver Distraction Model based on Multi-Modal Audio Analysis
Vehicle 2 - Driver model for partial automated driving	**IAS**	Industrial Partner WP9, Test Vehicle Owner
	TWT	Integration of driver distraction model
	DLR	Research Partner WP3, Driver Model Developer
Simulator 1 - Virtual HCD Platform	**IFS**	Research Partner, Driver Model, ADCOS-MOVIDA and COSMO-SIVIC main Developer
	INT	SME Partner WP4 & WP9, RT-MAPS support for V-HCD
	CVT	SME Partner WP4 & WP9, Pro-SIVIC support for V-HCD
	ERG	SME Partner WP3 and WP9, Eye Tracking System (To be confirmed)
	EAD-FR	Industrial Partner WP3, partnership on Monitoring Functions design (To be confirmed)
	ENA	Research Partner WP4, partnership on Validation Methods (To be confirmed)
Simulator 2 - Development and model-based evaluation of	**TAK**	Industrial Partner WP9, ADCOS Developer
	OFF	Research Partner WP2, Driver Model Developer
	ERG	SME Partner WP3 and WP9

an adaptive HMI	INT	SME Partner WP9
	TWT	SME Partner WP9

Table 1: List of test-vehicles and simulators developed in WP9

In addition, ATOS partner proposes an application to monitoring the UC simulation proposal in WP9 (i.e. Support Agent in LC manoeuvre). In particular, this systems allows you to view the state of each car and its devices, monitoring alarms, video streaming (e.g. a webcam installed in the vehicle), location and so on, also allowing to show it easily to the customers, partners and others in a demo. It is possible to communicate with a vehicle in real time, receive/send alarms from/to vehicle, receive AdCoS signals from the vehicles and take the control in an emergency situation.
Each use case will develop an AdCoS, which is described in detail in chapter 4.

3 HoliDes HF-RTP: Methodology

HoliDes HF-RTP will provide a set of connectable human factor engineering tools which support HF methods and processes. The main technical challenge in addressing this problem is the lack of open and common interoperability technologies supported by the different tools that generate and provide access to data, and of the underlying IT infrastructure, whether they are based on "Commercial off the Shelf" (COTS) products or custom-built solutions. An Interoperability Specification (IOS) will enable interoperability and collaboration between tools across the entire engineering lifecycle.
An RTP is a set of components which can be integrated using standard web technologies. To date, the internet is the best example of a scalable collaboration which has been successful beyond expectations. So long as you adhere to a set of common standards then anyone on the World Wide Web can produce content that can be consumed by anyone with a web browser. The goal of an RTP is to take those principles that made the internet so successful and apply them to Systems Engineering. The projects CESAR, MBAT and CRYSTAL have demonstrated, and still do, that these principles can be applied to different domains around Systems Engineering. HoliDes is going to take this a step further and demonstrate that these same principles can be used in Human Factors domain. That's why we call it an HF-RTP, a Human Factors – Reference Technology Platform. As it is detailed in D1.3, a major decision in HoliDes was that the HF-RTP should mirror the RTP which came out of the CESAR project.

3.1 Background to the RTP

The RTP is composed of a set of heterogeneous tools and services, further called RTP *components*.

1. The tools which are supported by the RTP are not just tools which are used for typical engineering tasks such as software programming IDEs and modelling software but also managerial software such as change configuration management, quality control and impact analysis.
2. The word 'services' is often used interchangeably with the word 'tools' and 'software'. Whilst this isn't wrong, in HoliDes a service is considered as *a computation process which performs some sort of analysis or data processing on existing data.* A service needs input from e.g. a data repository and delivers a predefined output to the

requester of the service. An example of this might be some statistical analysis tool which pulls its numbers in from a spreadsheet of results. This is in contrast to a modelling tool where the output is generated by an engineer creating drawing models and creating data.

All RTP components are able to be integrated, because they shall be compliant to the IOS. Figure 2 represents such a set of components, collected in the HoliDes RTP. To avoid any confusion about what an RTP is and what it is not: in general it is a bundle / pool of tools and services (components) which all offer a common interface to exchange data between each other. In case of the HoliDes RTP the data exchange format is based on the OSLC standard.

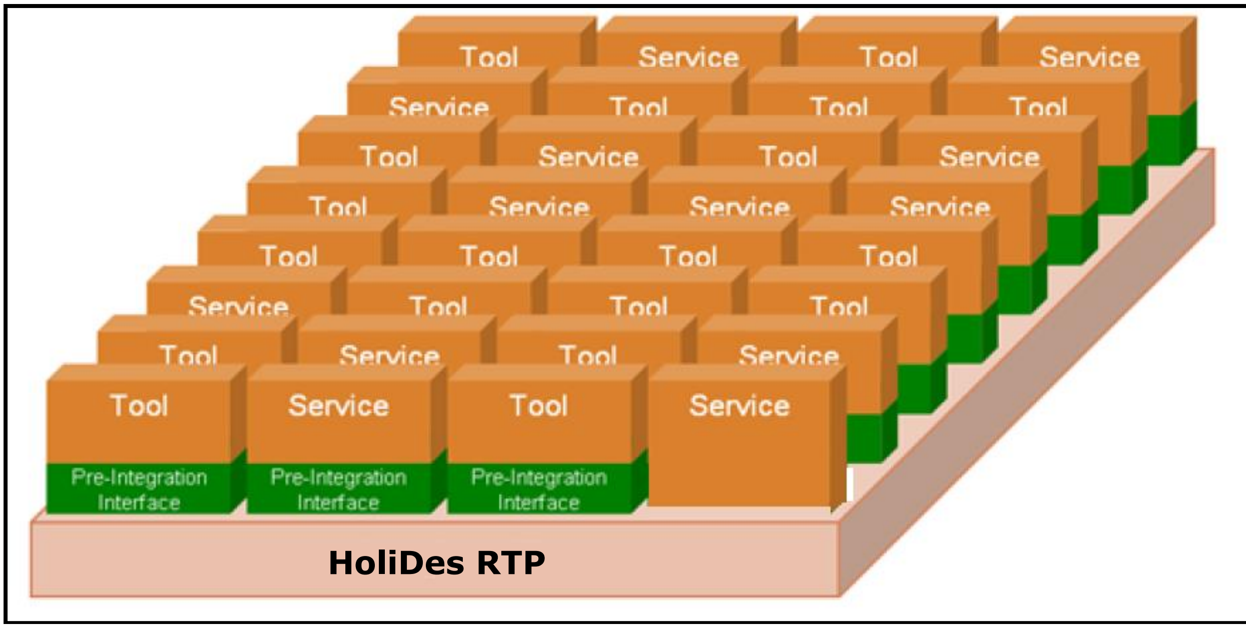


Figure 2 - The HoliDes RTP

Not every component needs to be able to deal with all the data types specified in the OSLC standard, but if for example a tool requests data about requirements from another RTP service, it has to implement the standard of a Human Factors Domain, which has to be defined within HoliDes. With this being said, the RTP itself (as a pool of components) is nothing that can be used directly to develop a product. To fit the specific needs of a certain development process, a proper selection and connection of RTP components needs to be done to form a useful *tool chain*. The selection and integration of the appropriate tools according to the needs of a specific development process is called *tailoring an RTP instance* (Figure 3).

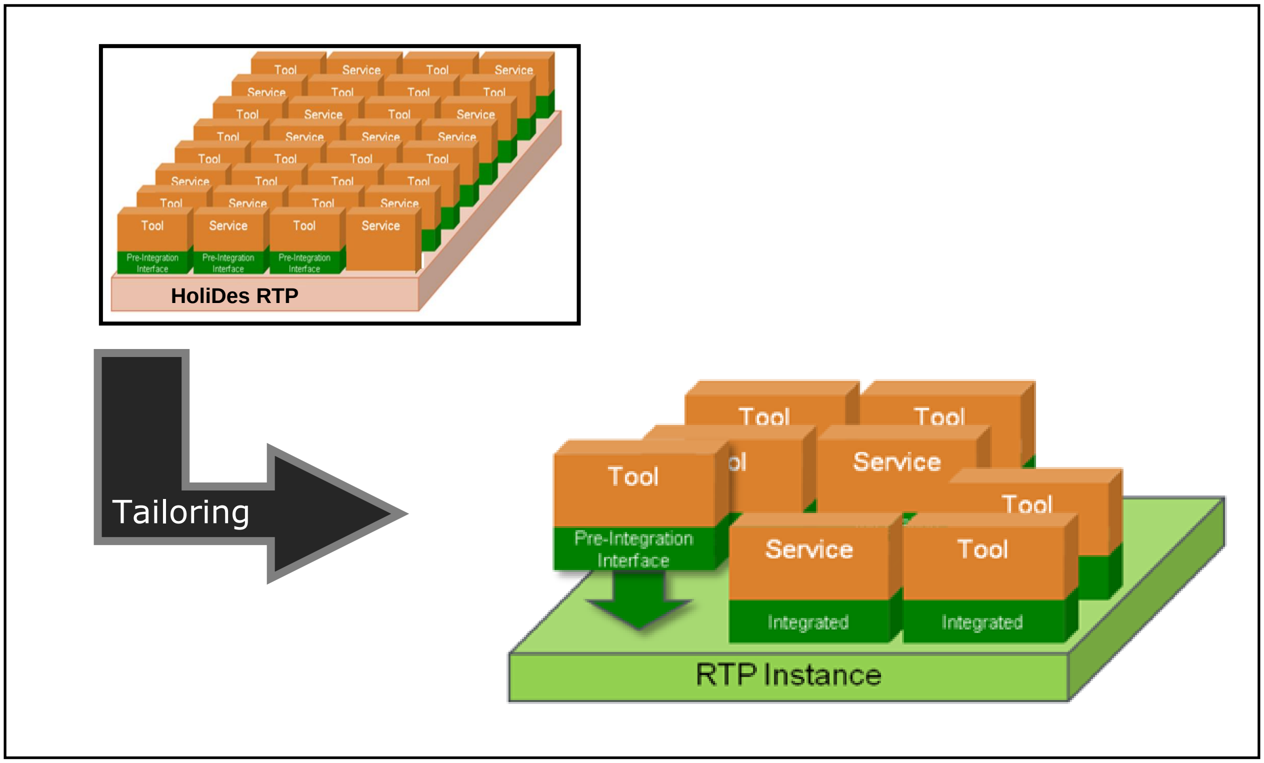


Figure 3 - An RTP being tailored to match the needs of industry

3.2 RTP Reference Architectural Model

The following section describes the structure of an RTP. Figure 4 shows the layers of an RTP. Whilst the model might seem confusing at first, it can be broken up into three basic layers, which are Domain, Application and Interoperability. This plan of the RTP was first defined in CESAR and has been carried over to other projects such as CRYSTAL and MBAT.

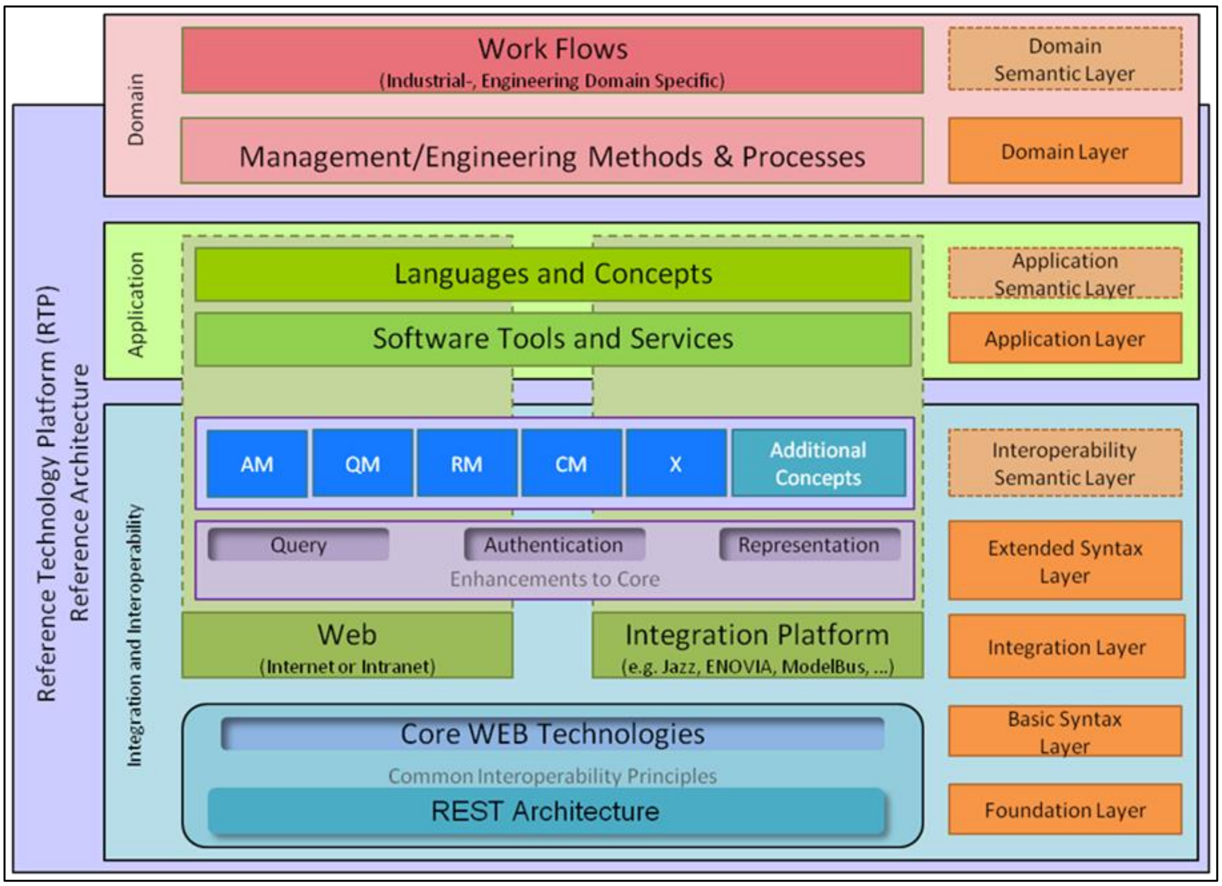


Figure 4 - Structure of an RTP

3.2.1 Domain

The domain layer contains elements which are specific for each application domain. In HoliDes there will be workflows from WP 6 to 9 who are providing the use case i.e. Aeronautical, Control Room, Medical and Automotive.

3.2.2 Application

The application layer is concerned with the actual components of the engineering environment such as software applications and human factors analysis tools. They exist in the RTP in a pre-integrated stage so that they can collaborate and share information needed to enter a tailoring stage.

3.2.3 Interoperability

The Interoperability building-block represents technologies needed to implement the actual interconnection and collaboration between tools. The used communication or data exchange technologies are inspired by the Web and OSLC. This building-block is concerned by the IOS.

4 HF-RTP tailoring process for the Automotive Domain

The following chapter collects the detailed descriptions of the AdCoS developed for each test-vehicle and driving simulator provided in WP9 (see § 2.2, Table 1, for the complete list). Each chapter section follows the same structure: AdCoS description; used tools; linear workflow; non-linear tool chain; covered requirements. This allows collecting tools and defining the workflows on which basis generalizations for the Automotive HF-RTP instance are being made.

4.1 The CRF Test Vehicle (Frontal Collision scenario)

4.1.1 List of partners

List of involved partners	
Partner	**Role**
CRF	Demonstrator owner; AdCoS implementation; support to the distraction classifier, driver's intention and co-pilot development; evaluation phase of the demonstrator with AdCoS.
REL	HMI design and implementation.
SNV	Test methodology for distraction data gathering; test methodology for AdCoS evaluation.
OFF	Development of a driver intention recognition (DIR) module for that provides the AdCoS with assessments and predictions about the internal context of the AdCoS concerning the human driver's current manoeuver intentions and driving behaviour.
OFF	Research Partner WP2 and WP3, Driver Intention Recognition module developer
UTO	Integration and implementation of driver's distraction classification algorithms.
INT	Support in using RT-MAPS (and PRO-CIVIC).
TWT	Driver Distraction Model based on Multi-Modal Audio Analysis

4.1.2 AdCoS introduction

The AdCoS implemented in CRF test-vehicle (TV) is a unique supporting system, adapting to the behaviour of the different agents, depending on the internal and external scenarios.
In particular, the scenario is represented in Figure 5, consisting of four cars with machine agents (e.g. PADAS) and human agents (drivers):

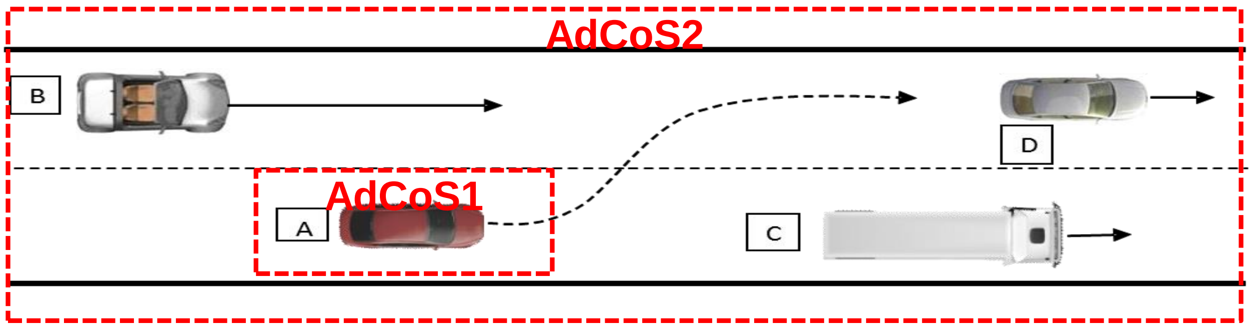


Figure 5 - Representation of the AdCoS scenario of CRF-TV

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 cognitive state and intentions of the driver, as well as external environment.
In the CRF TV, the following functionalities are implemented:

- Lane-Change Assistant (LCA) and Overtaking Assistant (OA)
- Forward Collision Warning (FCW), including assisted braking (and, optionally, automatic emergency braking).

The adaptation is carried out at a twofold level, being based on the external situation (e.g., a vehicle approaching from the rear of the ego-vehicle when its intention is to overtake) and, above all on the internal situation (e.g., the driver is distracted from the on-board infotainment system).
In order to accomplish this idea, we have adopted a statistical approach for the co-pilot, which constitutes the core of the AdCoS: the principle is to model our system as a Markovian Decision Process (MDP), to construct optimal warning and intervention strategies (WISs).

In this context, the classification of driver's cognitive state is the "trigger" for the adaptation. In particular, depending on the fact that the driver is distracted or not, the strategies of the AdCoS are modified, both for LCA and for FCW functions. The following schema sketches how this works:

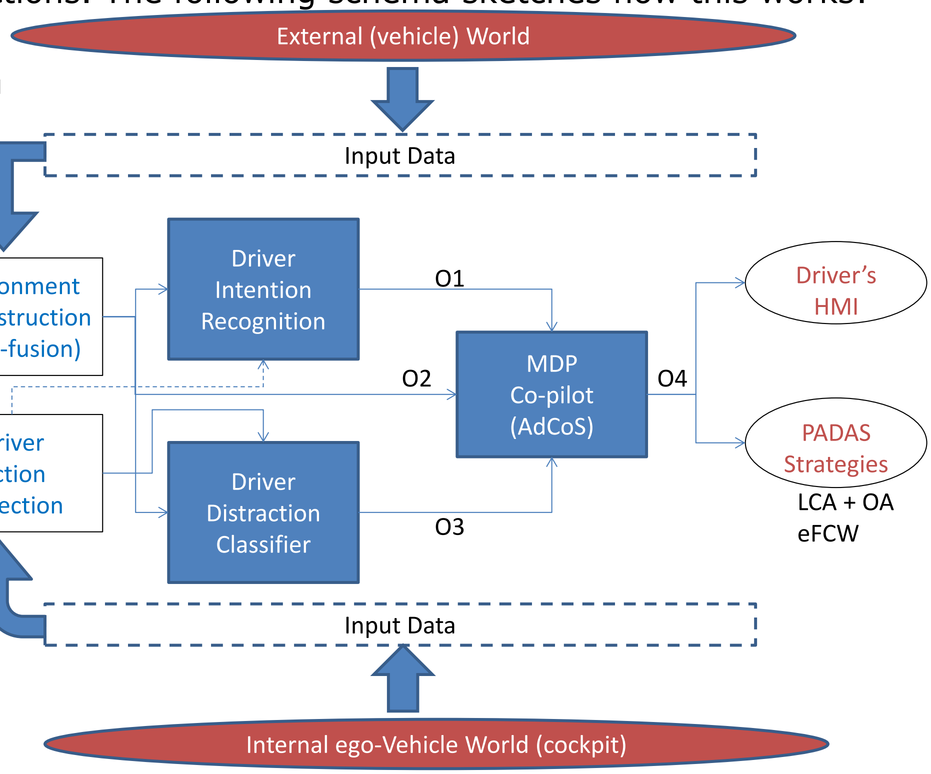


Figure 6 - Sketch of the AdCoS functionality.

With reference to Figure 6, the real world is detected by the on-board sensors (e.g., steering angle, yaw-rate, etc.) and ADAS sensors (e.g., Radar, Camera, etc.), while the detection of the internal world (internal to the test-vehicle, namely the cockpit) concerns the actions of the driver, where he/she is looking at, and so on (pedals position, eye-tracker, etc.). The data from the real-world are put together by the data-fusion (DF) module, which provides the list of obstacles (with a selection of the most relevant ones), the road curvature ahead, the presence of the lanes, and so forth (outputs O2).
In addition, all these data are then used by the **Driver Intention Recognition** (DIR) module and by the **Driver Distraction Classification** (DDC) module, as illustrated in the figure (O1 and O3 outputs, respectively).
The DIR aims at predicting the driver intention that is the manoeuvre which the ego-vehicle – and thus its driver – intends to do. The DDC module

provides information about the driver cognitive state, in particular if he/she is distracted or not (at the moment, it is not defined yet if the classification is on a binary level or on more levels).
O1, O2 and O3 are the inputs to the **Co-pilot** module, whose main output (O4) is represented by the computation of an "optimal" manoeuvre, based on the external situation and on the driver state. This manoeuvre is suggested from machine-agent to human-agent, by means of specific warnings, advice and information, according to the cognitive state and intentions of driver, as well as external environment.

4.1.3 Tools

The following table shows the situation for CRF:

MTTs	Use
CRF test vehicle	Development and test environment
MATLAB (and SIMULINK)	Support to the development Validation
RT-MAPS	Developer kit for AdCoS implementation Framework on the demonstrator vehicle Tool for data collection
PRO-CIVIC	Visualization and Traffic Simulation Software
Standardized questionnaires: NASA-TLX, subjective evaluation, etc. (still in progress)	Evaluation of acceptance/usability
RTMaps SDK	Development tool kit for RTMaps components
COSMODrive	Simulation of road environment and sensors
MS Visual Studio	IDE for component development

MTTs adopted from WP2-5	**Use**
Co-pilot based on Petri Nets Markovian Decision Process (PN-MDP)	Driver Model for the co-pilot. Adaptivity of the AdCoS.
Machine Learning Distraction Classifier	Classification of driver's visual distraction Analysis of driver's state
BAD Model, based on Bayesian Neural Networks	Classification and prediction of ego-vehicle manoeuvres, that is driver's intentions.
BAD MoB model parameter and structure learner	Machine-learning suite for learning the parameters and structures of BAD MoB models.

SURT (Surrogate Research Task)	Methodology to visually distract the user (for the data collection, on which the classifier is developed).

The cognitive state of the driver plays an essential role in the decision making process of the PADAS, e.g., to be able to issue warnings early if the driver is distracted, or to decide for more conservative actions when it comes to autonomous manoeuvres. The tool described in Section 4.5.2 (for TWT partner) can be used in this scenario: it monitors the driver's auditory and visual environment in order to estimate a driver distraction level using a cognitive model of the driver. The PADAS' decisions can then be adapted to this distraction level.

4.1.4 Description of Problems

An agent (the RED car in Figure 5, namely the ego-vehicle – EV) is preparing to overtake/lane-change a slower vehicle ahead (i.e. truck) and entering in collision path with another vehicle on the adjacent lane already overtaking the same EV. Another vehicle can travel ahead on the same adjacent lane.
The precondition is that driving faster and approaching a slower vehicle on a straight/curve road.
The successful end-condition is that the Lane-Change (LC) manoeuvre is performed without risks and without stop/strongly speed reduction of EV (minimum change in traffic flow, namely the function has not to disturb traffic). Alternatively, the EV can follow the slower vehicle ahead without any danger of collision (Car-following situation).
We can guess the following conditions:

- Extra-urban or motorways
- Middle traffic density
- Clear and good weather and visibility
- Sunny lighting conditions

In this context the addressed use-case (UC) is the following: the human-agent in the RED vehicle is distracted and thus the lane-change is performed without considering the vehicle approaching from the back. Thus, the Lane-Change Assistant (LCA) function is able to warn the driver in an appropriate way (that is, taking into account the cognitive state) and moreover it can suggest the right moment to perform the manoeuvre. This is done using the CRF vehicle.

4.1.5 Solution Description

The basic idea is to follow a *Statistical Approach for AdCoS* (PADAS, including FCW and LCA): it is modelled as MDP, in order to construct optimal warning and intervention strategies (WISs) – see Figure 7.

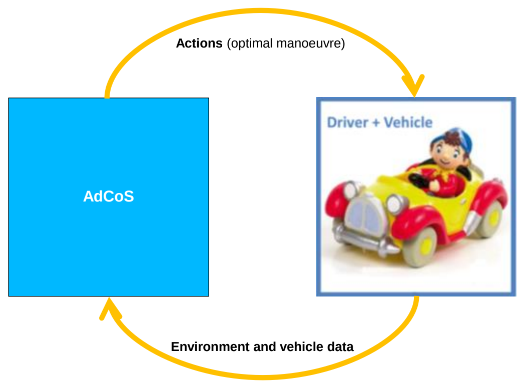


Figure 7 - Sketch of the interaction between AdCoS (system), vehicle and driver

In other words, we want to model the *sequential* decision making of a rational agent, that can be solved to derive the optimal set of actions (i.e. optimal strategy) which maximizes (minimizes) a specified reward (cost), as depicted in Figure 8.

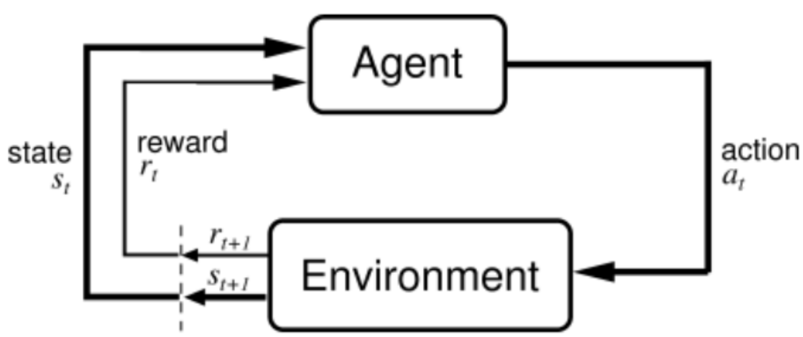


Figure 8 - Sequential decision process to solve an MDP.

So, the objective is to find an **optimal policy**, solving the MDP.

Driver Intention Recognition (DIR) module:

The DIR module will provide the CRF MDP Co-Pilot with predictions about the human driver's manoeuvre intentions and future driving behaviour. In this context, manoeuvre intentions are defined as the unobservable intentions of a human driver to perform one of a set of high-level manoeuvres, denoted by *B*, as defined by a skill hierarchy. Figure 9 shows an exemplary skill-hierarchy applicable for the WP9 use-case, where the complex driving behaviour for driving on highways can be represented by the four manoeuvres for performing lane changes to the left lane, lane changes to the right lane, lane-following, and car-following.

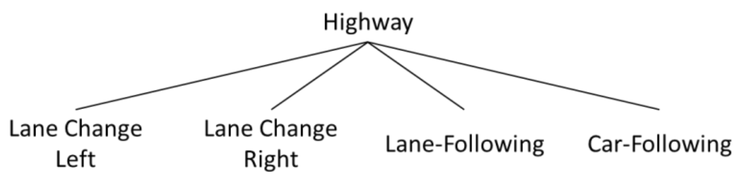


Figure 9 - Exemplary basic skill-hierarchy.

If necessary, each of these manoeuvres could be further decomposed into simpler low-level manoeuvres, such as emergency brakes and distance-keeping in the case of car-following, or decomposing a lane-change into the phases preparation, the lane change itself, and the final realigning of the vehicle in the new lane, as shown in Figure 10.

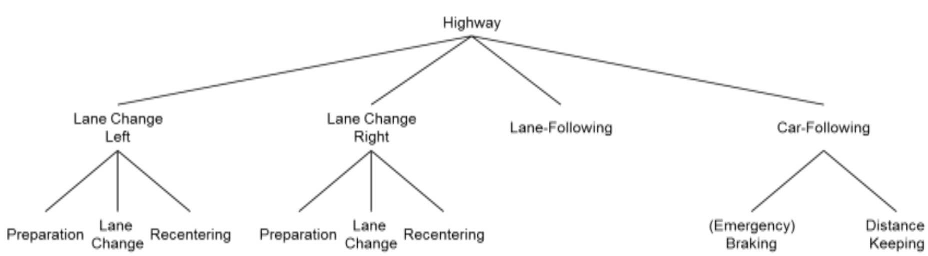


Figure 10 - Exemplary advanced skill-hierarchy.

Driving behaviour is defined as a sequence of actuator actions, denoted by *A* (by now, we assume steering wheel angles for lateral control and acceleration-/braking-pedal positions for longitudinal control) that is expected to be observed during a specific manoeuvre in the skill hierarchy.

Figure 11 shows the expected architecture of the DIR module. The module will primarily consist of three components: A probabilistic model of the human driver, based on and extending previously developed Bayesian Autonomous Driver Mixture-of-Behavior (BAD MoB) models, an inference engine, and an adaption manager.

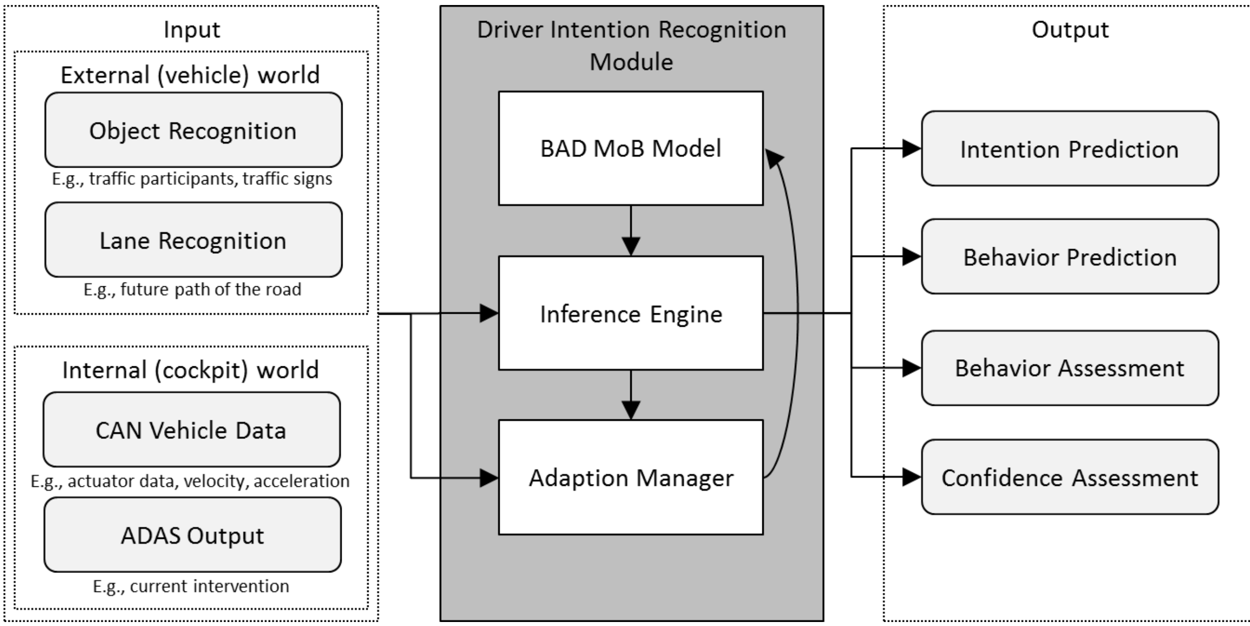


Figure 11 - Overview of the architecture of the DIR module.

BAD MoB Model:

BAD MoB models are probabilistic model of the human driver based on (Dynamic) Bayesian Networks. They will primarily be developed in WP2, while their utilization in the DIR module will be investigated in WP3.

A Bayesian Network (BN) is an annotated directed acyclic graph (DAG) that encodes a joint probability over a set of random variables $X = \{X_1, \ldots, X_n\}$. Formally, a BN B is defined as a pair $B = \{G, \theta\}$. The component G is a DAG, whose vertices correspond to the random variables $X_1, \ldots, X_n$, and whose arcs define the (in)dependencies between these variables, in that each variable X_i is independent of its non-descendants given its (possible empty) set of parents $Pa(X_i)$ in G. The component θ represents a set of parameters that quantify the probabilities of the BN. Given G and θ, a BN B defines a unique joint probability distribution (JPD) over X as:

$$P(X) = \prod_{i=1}^{n} P\big(X_i | Pa(X_i)\big).$$

DBNs extend BNs to model the stochastic evolution over a set of variables $X = \{X_1, \ldots, X_n\}$ over time. A DBN D is defined as a pair $D = \{B^1, B^{\rightarrow}\}$, where $B^1 = \{G^1, \theta^1\}$ is a BN that defines the probability distribution $P(X^1)$ and, under the assumption of first-order Markov and stationary processes, $B^{\rightarrow} = \{G^{\rightarrow}, \theta^{\rightarrow}\}$ is a two-slice Bayesian network (2TBN) that defines the conditional probability distribution (CPD) $P(X^t | X^{t-1})$ for all t. The nodes in the first slice of the 2TBN do not have any parameters associated with them, but each node in the second slice of the 2TBN has an associated CPD which defines $P\big(X_i^t | Pa(X_i^t)\big)$, where a parent $X_j^t \in Pa(X_i^t)$ can either be in time-slice t or $t-1$. The JPD over any number of T time-slices is then given by:

$$P(X^{1:T}) = \prod_{t=1}^{T} \prod_{i=1}^{n} P\big(X_i^t | Pa(X_i^t)\big).$$

A BAD MoB model is a DBN that implements the complex sensorimotor system of human drivers in a modular and hierarchical probabilistic architecture by combining multiple nested DBNs with distinct purposes. A BAD MoB model is based on the assumption that the complex driving competence of a human driver can be described by a skill hierarchy that hierarchically decomposes complex high-level driving behaviour or manoeuvres into simpler, or pure driving behaviours. Each basic skill in the skill hierarchy is realized by a distinct *action*-model that implements the isolated sensorimotor schema of the corresponding driving skill, i.e., the relation between driving actions *A* and the available observations from the environment, denoted by *O*. The appropriateness of a pure or a mixture of basic skills in a given situation is inferred by a *behavior-classification*-model. The functional interaction of action- and behavior-classification-models then allows the context-dependent generation, prediction, and assessment of complex human driving behaviour and intentions.

The primary functionality of a behaviour and intention prediction is to provide an estimate of the driver's current intention in respect to a preliminary defined set of potential intentions.

Inference Engine:

During runtime, the inference engine will utilize the BAD MoB model to answer probability queries about the desired output using the actual input as evidence.

Adaption Manager:
During runtime, the adaption manager will continuously assess the input and output of the DIR module to recalibrate the parameters of the BAD MoB model, in order to adapt the model to the actual driver and, over time, achieve a better predictive performance. For this, new techniques will be developed by OFF in WP3 to assess the current performance of the intention prediction.

Input:
During runtime, at a constant rate of 100ms, the machine agent expects synchronized and pre-processed input from the perception layer of the CRF AdCoS. This input includes but may not be limited to:

- Information about recognized objects like e.g., surrounding traffic participants and traffic signs,
- Information about the future path of the road, including e.g., the distance from lane edges,
- Information about the current state of the car, like e.g., current velocities and accelerations,
- Information about the current state of the actuators, like e.g., steering wheel angles and pedal positions,
- Information about the current state and outputs of other machine agents.

Output:
The primary output of the machine agent consists of sets of temporally evolving belief state estimates in the form of CPDs of the driving behaviour, resp. driving actions, *A* and manoeuvre intentions *B*, given the all current sensor observations *O*. In the following, the planned outputs, the machine agent will provide to the overall AdCoS shall be briefly introduced:

Maneuver Intention Classification / Prediction:
At each time-step t, the context assessment module will provide a maneuver intention prediction via the CPD $P(B^{t+n}|a^{1:t}, o^{1:t})$, where n is a desired anticipatory horizon. If $n = 0$, this can be seen as a classification of the current maneuver intention.

Lateral and Longitudinal Driving Action Prediction:

At each time-step t, the context assessment module will provide an action prediction via the CPD $P(A^{t+1+n}|a^{1:t}, o^{1:t})$, where n is a desired anticipatory horizon

Likelihood of the current driving actions:
At each time-step t, the context assessment module will provide the log-likelihood of the last m chosen driving actions $\log P\left(a^{t-m:t}|a^{1:t-(m-1)}, o^{1:t}\right)$. Under the assumption that the model represents normative driving, this can be used as a measure of normative driving. Low values or sudden drops (under the assumption that certain thresholds are defined) indicate that the driver does not show normative driving behavior and can be used as a further trigger for needed adaptation of the AdCoS.

Confidence:
At each time step t, for each provided output, the context assessment module will provide an assessment of its confidence in the inferred outputs. Note that this confidence does not relate to the probabilities itself (which are obvious from the outputs) but takes into account the confidence in the estimated parameters used for the inference. Higher confidence values indicate the confidence of the context assessment module in the correctness of the inference and should rise in the presence of more available data.

4.1.6 Solution Workflow: Linear

In developing Partial Autonomous AdCoS, the workflow shown in Figure 12 will be used.

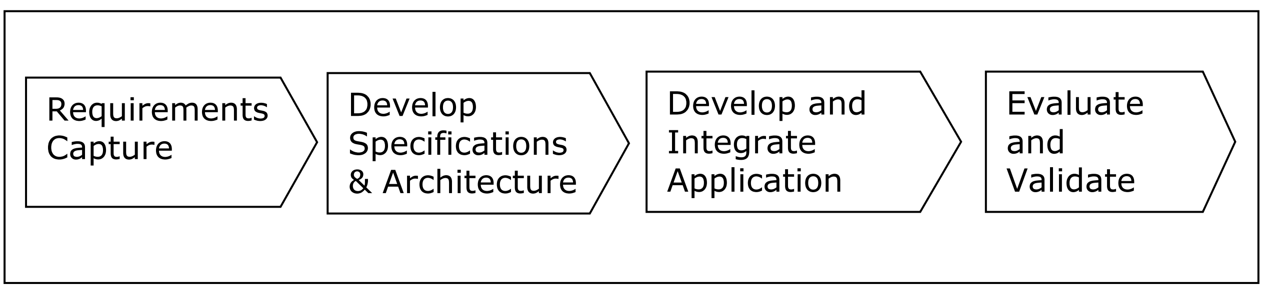


Figure 12 - Linear workflow for AUT AdCoS development

The design workflow used for the development and integration of the DIR module into the CRF AdCoS is shown in Figure 13.

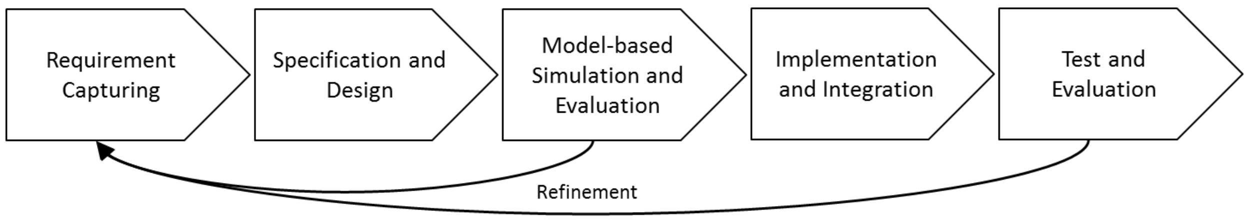


Figure 13- Linear workflow for the development of the machine agent.

4.1.7 Solution Workflow: non linear tool-chain

The following tool chain (Figure 14) shows how the different applications are expected to be used in the development of the Partial Autonomous AdCoS in CRF demonstrator.

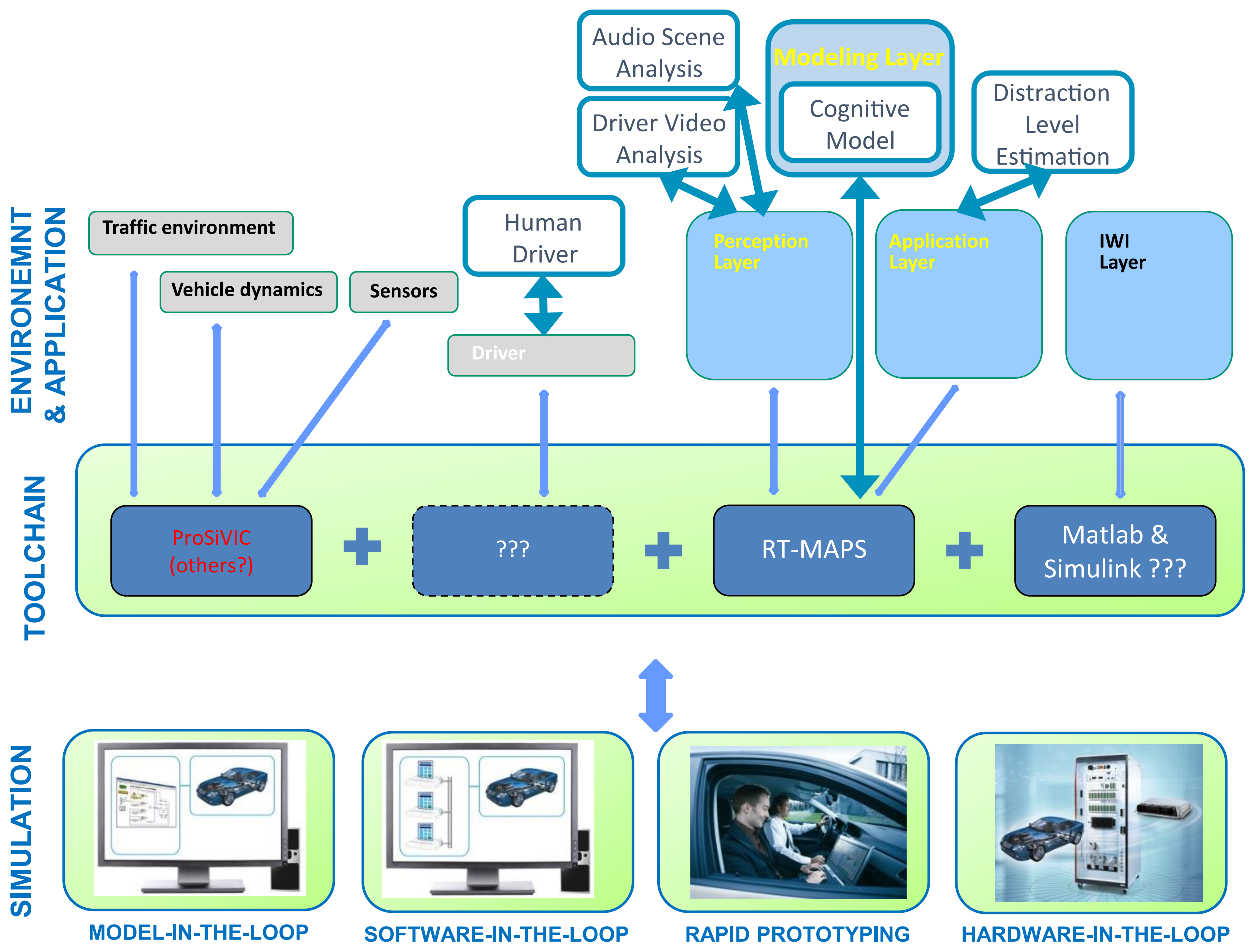


Figure 14 - Adapted AdCoS design from CRF including the driver distraction estimation from TWT.

This is the workflow we expect to follow in creating and design our AdCoS. This can be done with HoliDes technologies, as described in the figure, or also with other technologies if the HoliDes tools are not used yet.

Figure 15 shows the tool-chain expected to be used for the development of the DIR module and its connection to the CRF AdCoS tool-chain. Primarily, the development of the DIR module will be linked to the CRF tool-chain via RTMaps provided by INT as a MTT in HoliDes.

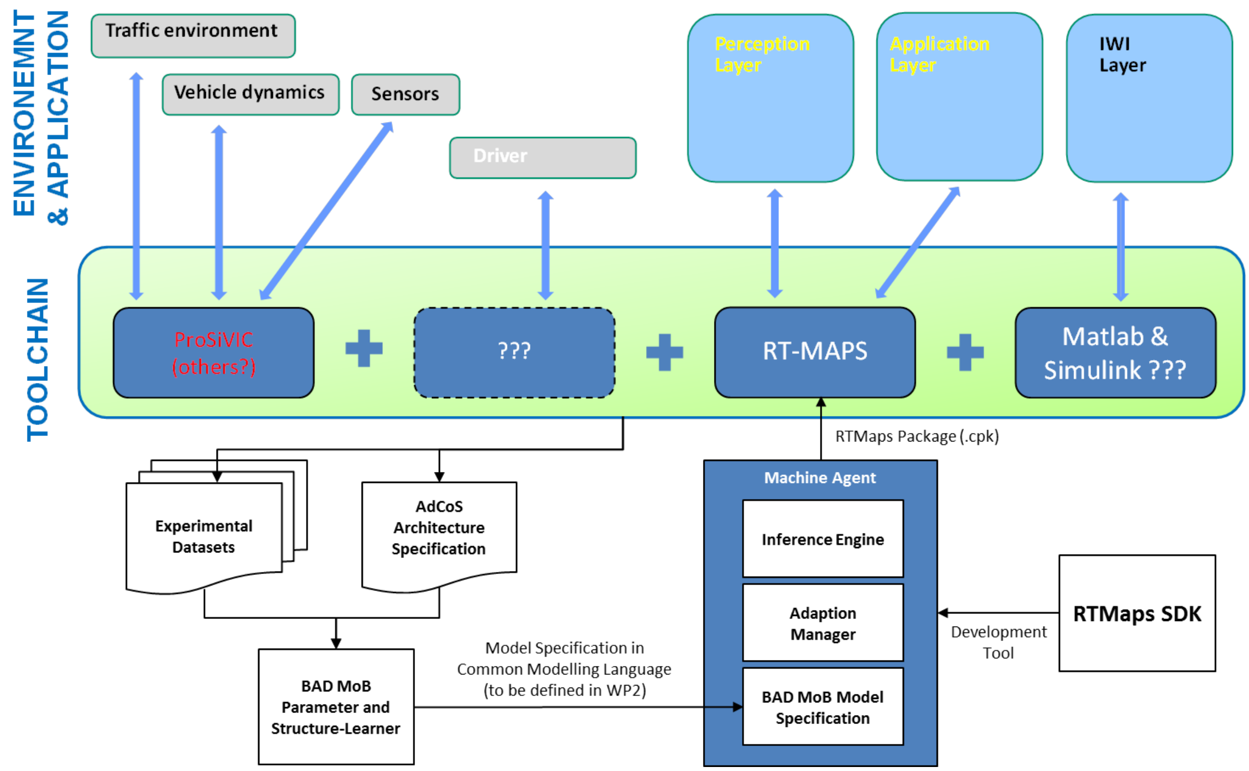


Figure 15 - CRF tool-chain for design, development and testing of AdCoS including the DIR machine agent (before going on real vehicle).

At each stage during the development of the CRF AdCoS, the current specification of the AdCoS architecture defining e.g., available inputs of the perception layer, will be used to derive the potential graph-structures for the BAD MoB model.

The actual graph structure and parameters of the BAD MoB model are learned via machine-learning methods. The algorithms and learning procedures will be implemented in proprietary software developed by OFF (BAD MoB Parameter and Structure-Learner). The output is a fully defined specification of a BAD MoB model, essentially consisting of a description of the graph-structure and the parameters of the model. The specification will

be described using the yet to be defined common modelling language developed in WP2. For learning the structure and parameters of an initial BAD MoB model, experimental datasets of time-series of data samples in the same format as expected during runtime is required. These could be obtained in simulator studies using driving simulators available in HoliDes utilizing ProSIVIC for simulating sensors, or in real-life driving studies using the CRF test vehicle.
The actual DIR machine agent will be developed in WP9 using the RTMaps SDK provided by INT for MS Visual Studio. The output is a RTMaps component package (.pck) which can then be used within RTMaps for the overall development and simulation of the CRF AdCoS.

4.1.8 Covered Requirements

The list of requirements is the following:

- WP9_CRF_AUT_REQ3_v1.0 ⇒ Classification of driver's cognitive state.
- WP9_CRF_AUT_REQ6_v1.0 ⇒ Risk for collision in straight roads (Forward Collision Warning - FCW), based on a minimum of 2 levels.
- WP9_CRF_AUT_REQ7_v1.0 ⇒ Risk for collision in curves (FCW), based on a minimum of 2 levels.
- WP9_CRF_AUT_REQ9_v1.0 ⇒ Lane Change Inhibit.
- WP9_CRF_AUT_REQ10_v1.0 ⇒ Lane Change Assistant (LCA).
- WP9_CRF_AUT_REQ15_v1.0 ⇒ Braking with Front Obstacle.
- WP9_OFF_AUT_REQ1_v1.0 ⇒ Offline parameter and structure learning
- WP9_OFF_AUT_REQ2_v1.0 ⇒ Online parameter learning and adaptation
- WP9_OFF_AUT_REQ4_v1.0 ⇒ Guaranteed maximal computation time
- WP9_OFF_AUT_REQ6_v1.0 ⇒ Interface between Bayesian Driver Model an CRF AdCoS.
- WP9_OFF_AUT_REQ7_v1.0 ⇒ Manoeuvre Classification.
- WP9_OFF_AUT_REQ8_v1.0 ⇒ Manoeuvre Intention Classification.
- WP9_OFF_AUT_REQ9_v1.0 ⇒ Driving Style Classification.
- WP9_OFF_AUT_REQ10_v1.0 ⇒ Likelihood of current driving behaviour.
- WP9_OFF_AUT_REQ11_v1.0 ⇒ Confidence in Manoeuvre Classification.
- WP9_OFF_AUT_REQ12_v1.0 ⇒ Confidence in Intention Classification.

- WP9_OFF_AUT_REQ13_v1.0 ⇒ Confidence in Driving Style Classification.
- WP9_TWT_AUT_REQ03_v0.1 ⇒ Car simulation with operating vehicle speed range and different route options.
- WP9_TWT_AUT_REQ04_v0.1 ⇒ Algorithm for analyses of distraction.
- WP9_TWT_AUT_REQ12_v0.1 ⇒ Automatic external context cues, like lane following/ changing behaviour.
- WP9_TWT_AUT_REQ13_v0.1 ⇒ Synchronization of various measurement sources.
- WP9_TWT_AUT_REQ14_v0.1 ⇒ Feedback rendering modality of estimated distraction level.

In addition, there are also the requirements dedicated to the Human Machine Interface (HMI), under the responsibility of REL partner, for the interaction and communication between human-agent and machine-agent.

4.2 The Ibeo test-vehicle - Driver model for partial automated driving

4.2.1 List of Partners

Partner	Role
IAS	• Implement highly automated driving functionality into passenger vehicle. • Provide interface for adaptation and cooperation with the human driver.
DLR	• Driver Model: analysing the human driver to enable adaptation of the machine agent to the obtained characteristics of the human operator.
ERG	• Eye gaze detection and tracking. • Providing input to the Driver Model to determine distraction with respect to single objects in the environment.
TWT	• Implementing an audio-based distraction estimation for the human driver in order to adapt warning thresholds.

4.2.2 AdCoS description

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. This encompasses cooperation and adaptation 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 NHTSA 2013.

The novelty of the automated driving approach applied to the ibeo test vehicle (a partial automated vehicle) is the advanced cooperation with a human driver and adaptation to his or her capabilities, needs and preferences, to other road users and the environmental conditions. It is characterized by a decentralized decision making between the artificial and the human intelligence. The machine agent can control the lateral and longitudinal movements of the vehicle. The AdCoS implemented in the ibeo test vehicle adapts the default behaviour of the machine agent towards the human driver. The adaptation considers two levels. In the first level the AdCoS adapts the driver's preference of manoeuvres. In second level the AdCoS adapts the shapes of the trajectories connected to the manoeuvres.

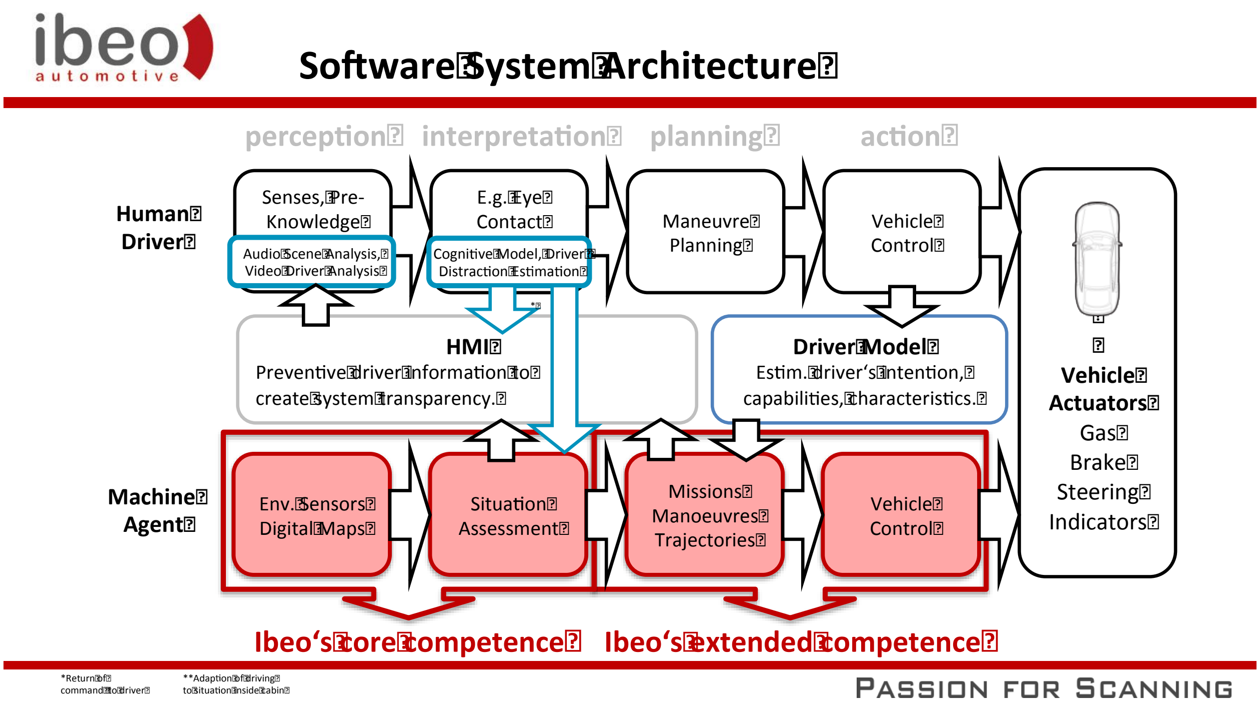


Figure 16 - ibeo test vehicle architecture

Figure 16 visualises the interaction between the human driver and the machine agent. There are two modules connecting the communication flow between the human driver and the machine agent, and which close the loop of interaction:

1. **HMI**
   The HMI is unidirectional and provides information from the machine agent to the human driver.
2. **Driver Model**
   The Driver Model analyses the behaviour of the human driver and provides information about the driver to the machine agent. The machine agent can then use this information to adapt the driving style to the individual human driver.

These two modules close the interaction loop between the human driver and the machine agent.
The highly automated driving (HAD) system is characterised by four main features, as depicted in Figure 17:

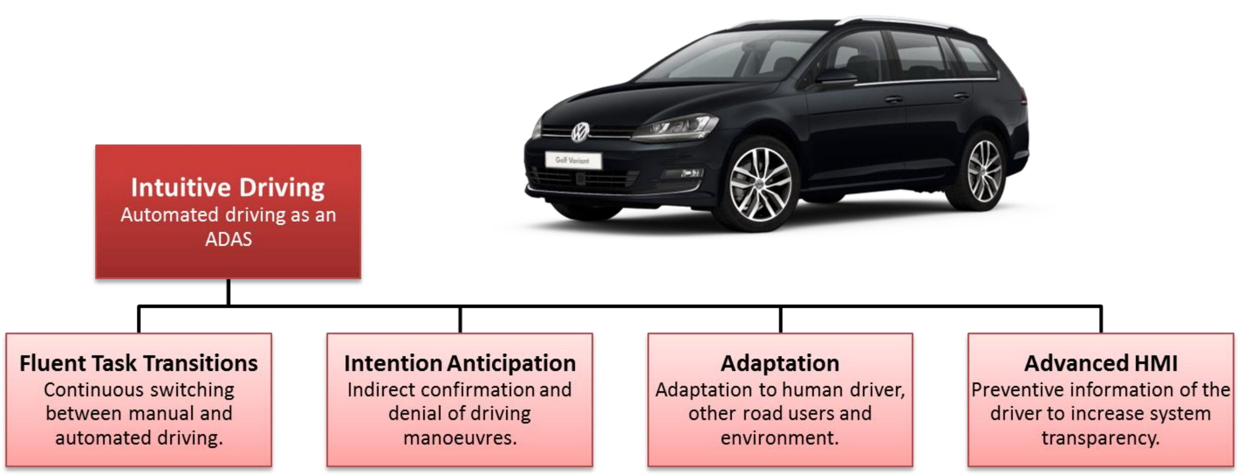


Figure 17 - Features of the HAD system

1. **Fluent Task Transition**
   The switching between manual and automated driving shall be fluent. This means that the driver can give control to the automated system at any time, only if the function is available. Also the driver can interact with the automated system by operating the standard control inputs (gas, brake, steering wheel, indicators). In case the system detects that it is unable to handle an upcoming traffic situation it will warn the driver with sufficient time to take over control.

2. **Intention Anticipation**
   In case the human driver operates the pedals, the steering wheel or the indicators during automated driving, the system will automatically anticipate the driver's intention, e.g. if the vehicle is following a truck in the outer lane of a highway and the driver sets the indicator to the left, the automated system could anticipate that the driver wants to overtake and go faster.
3. **Adaptation**
   The automated vehicle will be able to determine a range of safe driving manoeuvres at any time. Within this range the system offers room to adapt the driving style according to the driver's needs, characteristics and capabilities.
4. **Advanced HMI**
   To keep the driver informed about the detected traffic situation and planned manoeuvres the system will include and HMI to communicate these information to the driver. The HMI is an important part of the overall system to create transparency for the human driver.

4.2.3 Tools

MTTs (already used)	**Use**
ibeo test vehicle	Test environment
Simulators	Test environment
Matlab	Validation
MS Visual Studio 2010	Developer kit for CONFORM
Vires VTD	Visualization and Traffic Simulation Software
Dominion	Middleware and Simulation Software
SoSci Survey	Web interface for questionnaires
Standardized questionnaires: attrakDIFF, semantic differentials	Evaluation of acceptance/usability

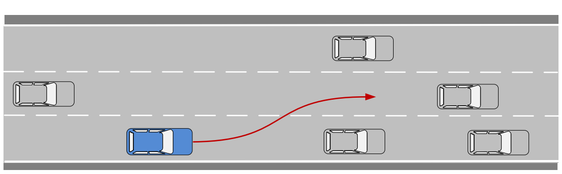
MTTs adopted from WP2-5	**Use**
CONFORM (WP3 tool under development)	Driver Model and data visualization

4.2.4 Description of Strategy

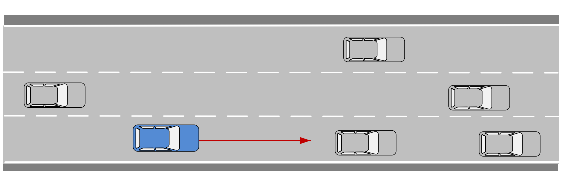
The ibeo test vehicle as a partial automated vehicle allocates a new role to the driver. The driver only has to monitor the system. This new role requires the willingness of the driver to hand over control to the vehicle. A central point to enhance this willingness is to ensure a high acceptance of the system behavior by the driver. The ibeo test vehicle realizes this by adjusting its behavior on an average driver. Nevertheless studies unveil a large variance in natural driving behavior between drivers. Therefore a situation based adaptation to the individual driving behavior is an opportunity to further improve the acceptance. In the ibeo test vehicle this is done by an additional component: The driver model.
The driver model is called CONFORM. CONFORM will be part of the ibeo test vehicle. The basic idea is that CONFORM learns the natural driving behavior during manual driving. This happens before the test vehicle activates its partly automated driving mode. CONFORM compares the learned natural driving behavior with the default behavior. Only in case the behavior differs, we call that a conflict, CONFORM forwards relevant parameters to the machine agent to avoid the conflict later on when partly automated driving mode is active. CONFORM distinguishes between maneuver and trajectory conflicts. For both types of conflicts CONFORM exchanges the parameters with the ibeo test vehicle as defined in the next section. The following figures wrap up the different possibilities of adaptation.

<u>Different driver but same situation</u>:
Different maneuver preference, but same trajectory style

Driver A

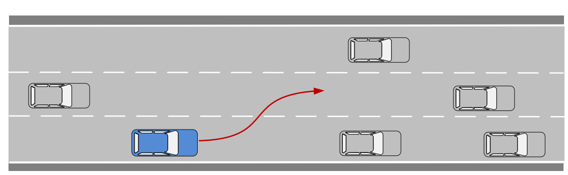


Driver B

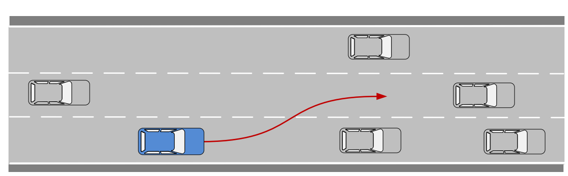


<u>Different driver but same situation:</u>
Same maneuver preference, but different trajectory style

Driver C

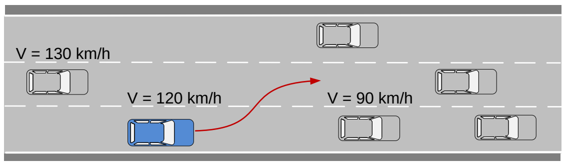


Driver D

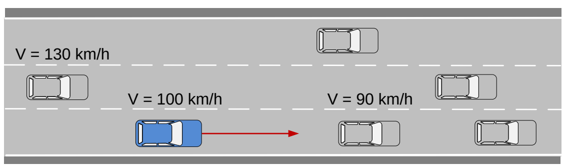


<u>Same driver but different situation:</u>
Different maneuver preference

Situation A

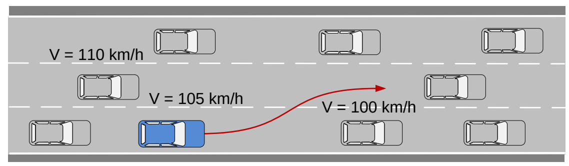


Situation B

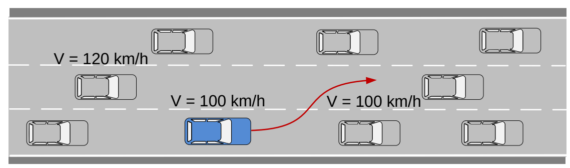


<u>Same driver but different situation:</u>
Different trajectory

Situation C



Situation D



4.2.5 Solution Description

In general CONFORM (**Conf**lict rec**o**gnition by image p**r**ocessing **m**ethods) is a visualization tool and method to analysis the conflict potential between the human agent behavior and machine agent behavior in a given situation/context. The basic idea of CONFORM is to adapt the behavior of the machine agent if only if the conflict potential is above a certain threshold. If the conflict potential is below the machine agent behavior remains the default behavior designed by the system designer. In the context of the HoliDes CONFORM is used as a driver model and visualization tool which enables an adaptation of the machine agent behavior towards the natural behavior of the driver. More precisely the adaptation considers the preference of maneuvers and the shape of the corresponding trajectories in dependency of the position and velocity of the surrounding vehicles.
To realize this kind of driver and situation adaptation, CONFORM obvious has to consist of three modules: a situation classifier to adapt to the situation, a memory to adapt to the natural driver behavior and a conflict analyzer to recognize the urgency for an adaptation. Figure 18 illustrates this general structure.

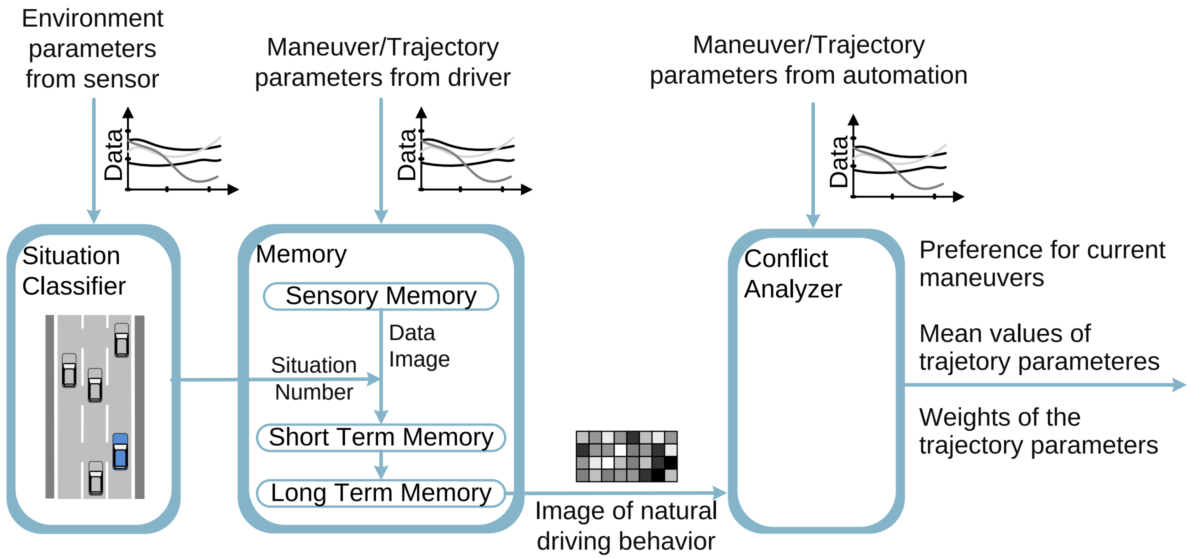


Figure 18 - Structure of CONFORM in the HoliDes project.

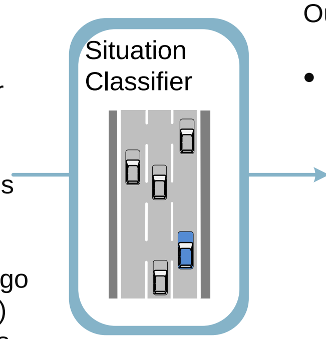
The main ideas of CONFORM, like the organization of the memory or the recognition of conflicts are already explained in Griesche, Dziennus 2013. They are use case independent and therefore we refer to Griesche, Dziennus 2013 to get a better understanding. We will rather focus on the use case dependent parts of CONFORM. First, these are the inputs to the situation classifier which define the situation state vector and situation number respectively. Second, these are the inputs to the memory, which define the data image. Third, these are the outputs of the conflict analyzer which describe what to adapt and how.

Situation Classifier

In the first implemented version the situation classifier will transform the input variables into a state variable and connect the different states to state vector as already done in Griesche, Dziennus 2013. In HoliDes the situation or context is a two or three lane highway with a varying traffic density. Since the basic decisions while driving happen on the maneuver level inputs have to be somehow related to the possible maneuvers on a highway. Possible maneuvers are: change lane left/right, follow vehicle/lane, approach vehicle, stop vehicle and an emergency brake. Figure 19 shows the considered inputs based on the mentioned maneuvers and the capability of the sensors.

Inputs:

- Ego velocity
- Distance/position of other obstacles relative to own position
- Velocity of other obstacles
- Acceleration of other obstacles (if available)
- Current lane number of ego vehcile (left, middle, right)
- Classification of obstacles



Output:

- Situation State Vector

Figure 19 - Inputs/Outputs definition for the Situation Classifier

<u>Memory</u>

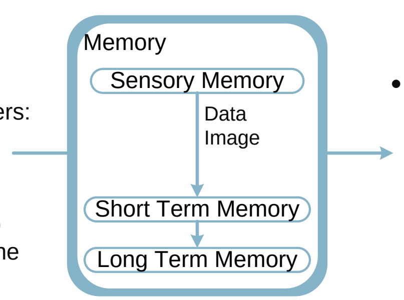
As mentioned above the memory is responsible for the adaptation to the natural driving behavior. By adaptation to the natural driving behavior we mean in HoliDes the preference of maneuvers and the shape of the corresponding trajectories. Therefore the inputs to the memory can be distinguished in maneuver related inputs and in trajectory related inputs. Figure 20 illustrates the considered inputs for the memory in HoliDes.

Inputs:

From Situation Classifier:
- Situation State Vector
- Ego Velocity
- Velocity/rel. distance of other obstacles

Maneuver related parameters:
- Steering angle
- Brake pedal position
- Accelerator position
- Indicator position (on/off)
- Lateral position within lane

Trajectory related parameters:
- Long./Lat. acceleration ego vehicle



Output:

- Natural driving behavior encoded in a data image for each situation state vector
- Distribution of data images for each situation state vector

Figure 20 - Inputs/Outputs definition for the memory module.

Conflict analyzer

In context of Holides the outputs of the conflict analyzer are directly used as inputs for trajectory planning algorithm of the vehicle automation (see Figure 21). We will distinguish between a maneuver conflict and a trajectory conflict. To resolve the maneuver conflict the conflict analyzer forwards the weight/preference of the each possible maneuver to the vehicle automation. The vehicle automation than can decide if the maneuver is technical possible and safe. To resolve the trajectory conflict the conflict analyzer sends the mean values of the parameters which are used to describe the trajectory of the corresponding maneuver. Moreover we attach the uncertainty index as well (see Griesche, Dziennus 2013 for detailed explanation) to adapt the costs of the cost function in the optimization routine of the trajectory planning algorithm. Hence we are able to focus on parameters which seem to be relevant for the driver or at least characteristic for the driver.

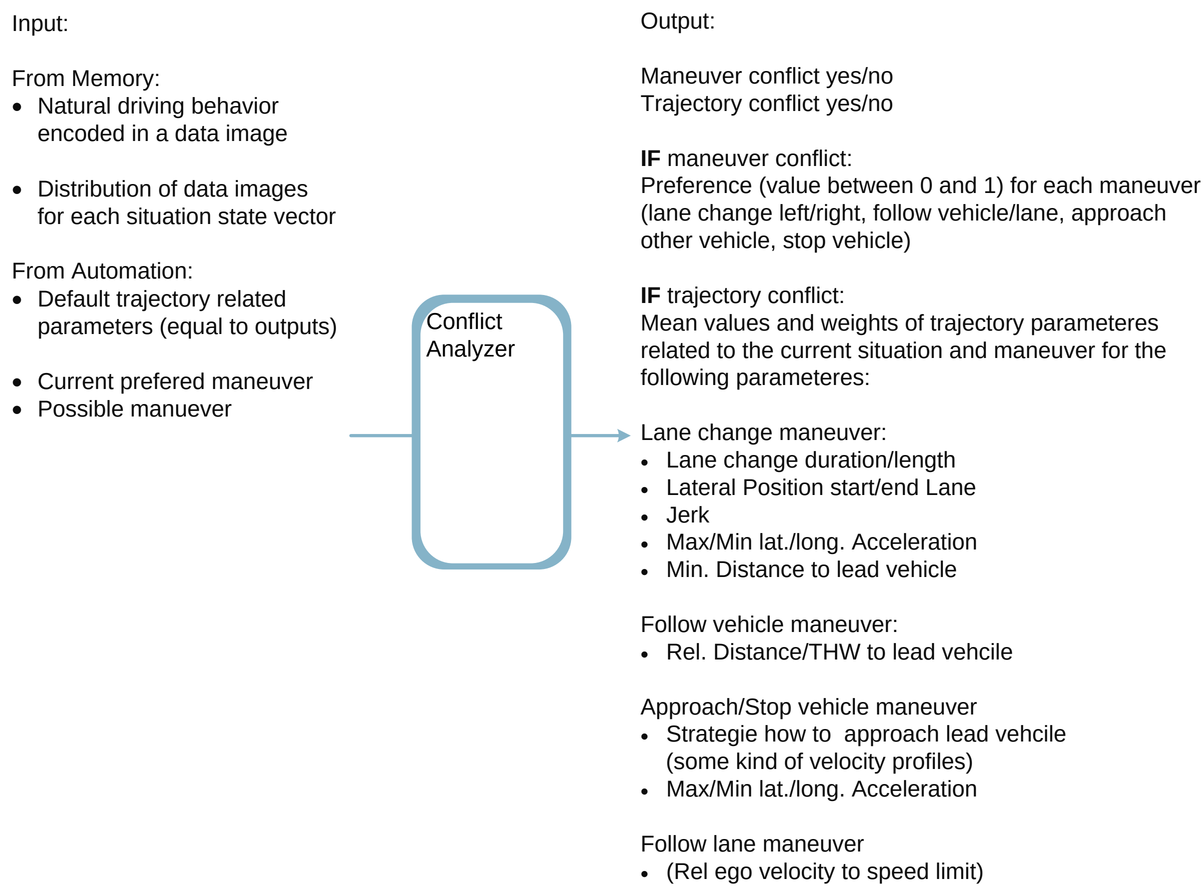


Figure 21 - Inputs/Output definition for the Conflict Analyzer

4.2.6 Solution Workflow: Linear

Figure 22 illustrates the linear workflow for the ibeo-DLR-AdCoS. The linear workflow and the non-linear workflow respectively consist of three larger development phases. In each phase CONFORM requirements and specifications are refined and updated. Additional we increase the portion of reality and decrease the portion of virtuality per phase.
More precisely in the first phase we use the wizard of Oz technique to realize a rapid prototype in the simulator by emulating the behaviour of CONFORM through a human being. The rapid prototype enables a first refinement of the initial requirements and specification based on the feedback of first "users" without any coding in advance. In the second phase we start the actual implementation of CONFORM considering the refined requirements and specifications from the first phase. The testing and evaluation of the implemented model is first done in the DLR simulator to be sure that the model works at least in the simulation. If this is the case we integrate the model in the ibeo test vehicle and perform a final evaluation of the AdCoS.

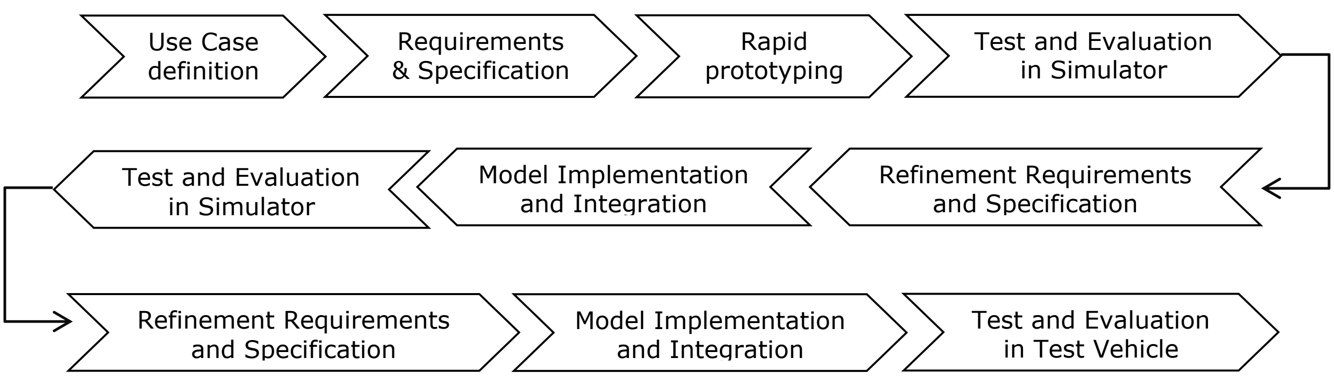
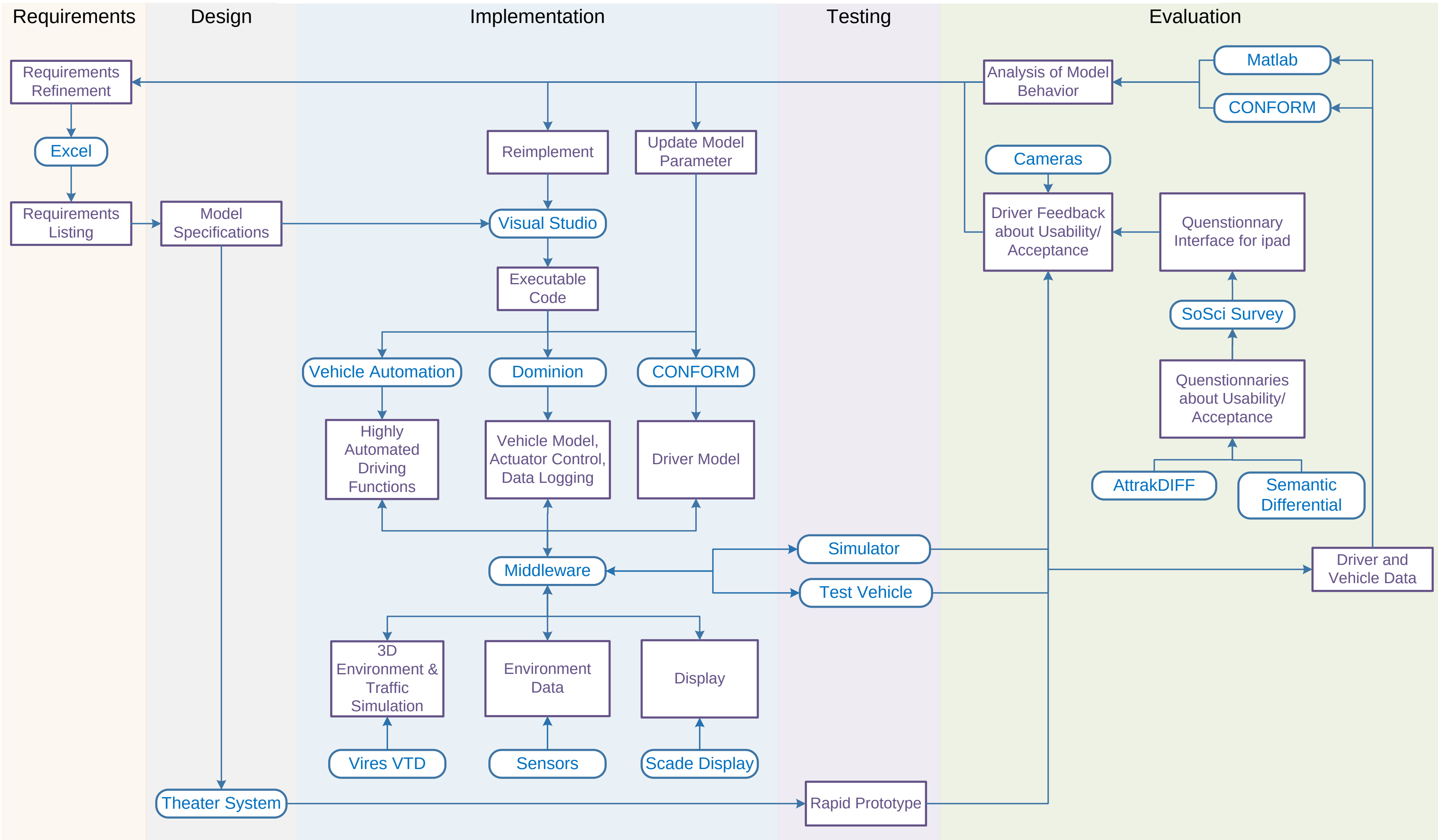


Figure 22 - Linear workflow of the ibeo DLR AdCoS

4.2.7 Solution Workflow: non linear

Figure 23 shows the non linear workflow for the ibeo-DLR-AdCoS from a tool and technology viewpoint. Especially software tools involved in the evaluation and implementation of CONFORM are added and highlighted.

Figure 23 - Non-linear workflow of the ibeo DLR AdCoS development.



4.2.8 Covered Requirements

WP9_OFF_AUT_REQ1_v1.0	Offline parameter and structure learning
WP9_OFF_AUT_REQ2_v1.0	Online parameter learning and adaptation
WP9_OFF_AUT_REQ9_v1.0	Driving Style Classification
WP9_OFF_AUT_REQ10_v1.0	Likelihood of current driving behavior
WP9_OFF_AUT_REQ12_v1.0	Confidence in Intention Classification
WP9_OFF_AUT_REQ13_v1.0	Confidence in Driving Style Classification
WP9_IAS_4.1	Driver Model: Maneuver Interface
WP9_IAS_4.2	Driver Model: Maneuvers
WP9_DLR_AUT_REQ1_v1.0	Learning of individual driving behavior
WP9_DLR_AUT_REQ2_v1.0	Online learning
WP9_DLR_AUT_REQ3_v1.0	Offering safe maneuvers

4.3 The IFS Simulator Demonstrator

In the frame of HoliDes project, IFSTTAR will develop both an AdCoS based on *MOVIDA* functions (for Monitoring of Visual Distraction and risks Assessment) and a *Virtual Human Centred Design (V-HCD) platform* (named *COSMO-SIVIC*), to be used in WP9 as a tailored HF-RTP in order to virtually design and test the MOVIDA-AdCoS.

4.3.1 MOVIDA-AdCoS Description

The AdCoS based on MOVIDA functions (Monitoring of Visual Distraction and risks Assessment) to be designed by IFS will be an integrative system combining several simulated Driving Aid sub-Systems (DAS) to be managed in and Adaptive and Cooperative way by MOVIDA, according to the drivers' visual distraction status and traffic situational risks assessment (Figure 24):

- The core DAS sub-systems to be combined in this AdCoS are a Collision Avoidance Systems (CAS; Forward, Rear and Lateral), Adaptive Cruise Control (ACC), Lane Keeping Assistance (LKAS), and Lane Change Assistant (LCA).
- Regarding their Human-Machine Interaction modalities, all this DAS should be based on (1) warning and/or (2) vehicle automation functions.
- Adaptive and Cooperative abilities in the AdCoS will be supported by a set of monitoring functions (i.e. MOVIDA module), in charge to monitor the drivers, to assess their visual distraction state, and to evaluate the risk of accident in the current traffic situation.
- From MOVIDA monitoring, Risk-based analysis algorithms and a Centralized Management of DAS sub-systems 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 drivers needs in accordance with their visual distraction state and the situational risks.

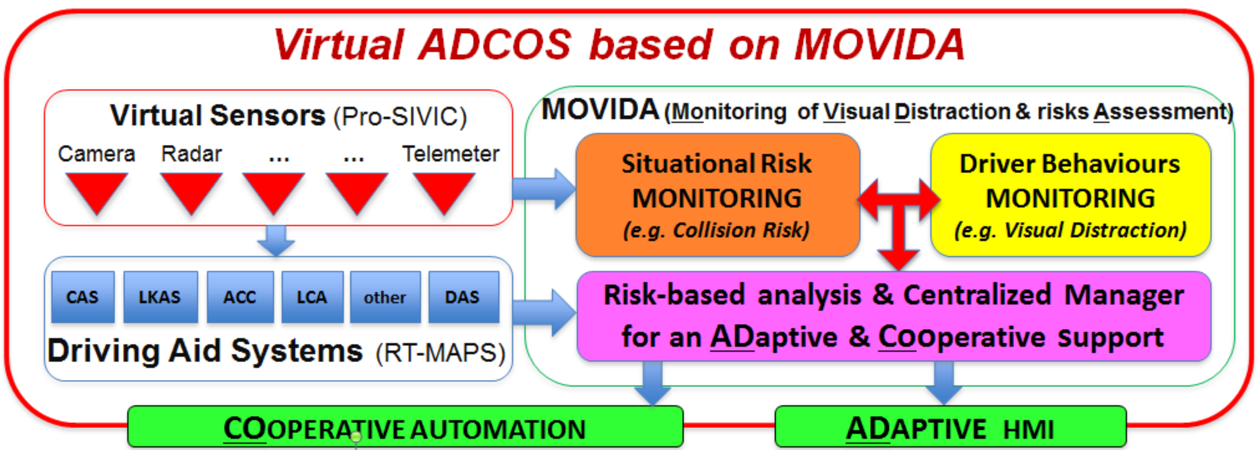


Figure 24 – Functional architecture of AdCoS based on MOVIDA

4.3.1.1 Driving Scenarios and Use Cases for MOVIDA-AdCoS

The Driving Scenarios and MOVIDA-AdCoS Use Cases to be investigated by IFSTTAR in WP9 will concern driving situation occurring on a two-lanes Inter-Urban Highway limited to 90 km/h (Figure 25).

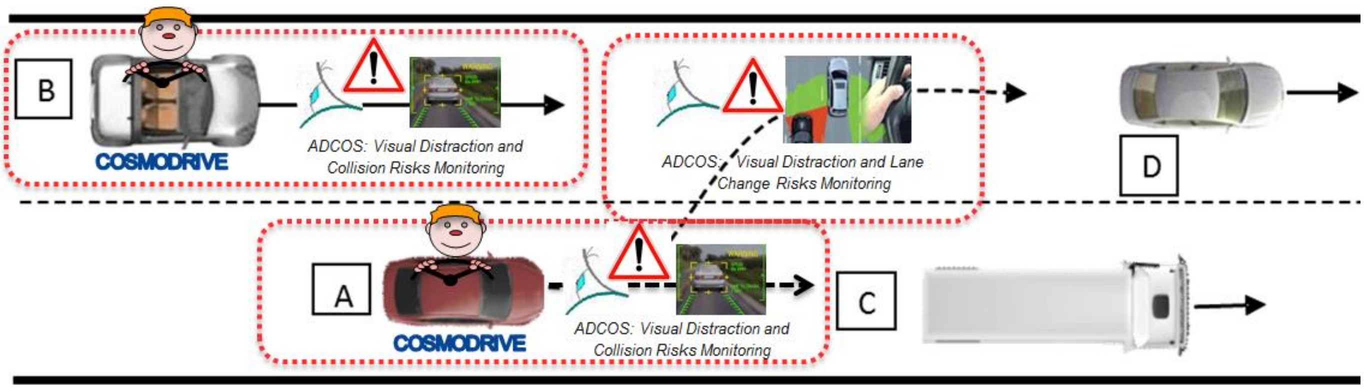


Figure 25 - Driving Scenarios and Use cases for the MOVIDA-AdCoS

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

4.3.1.2 Tools used for MOVIDA-AdCoS design and support

Tool (Partner)	Use
RT-Maps (INT)	Integrative software to support DAS and AdCoS simulation
Pro-SIVIC (CVT)	Road Environment & Virtual Sensors for AdCoS simulation
Tobii/Dikablis (ERG)	Eye tracking systems (analysis of drivers' visual scanning)
MoViDA module (IFS) (MatLab, StateFlow, and RTMaps modules)	Monitoring of divers' visual distraction and situational risks, associated with Risk-based analysis algorithms and a Centralized Manager of DAS systems

4.3.1.3 Specific Design and Analysis Problems

In order to avoid accident risk due to driver's visual distraction and to adapt

aids in accordance with the traffic conditions, IFSTTAR will design and develop a set of Monitoring Functions and Decision Support Functions, to be integrated in the AdCoS for supporting Adaptive and Cooperative abilities of Driving Aid Systems, according to human drivers' errors and to the Situational Risk (e.g. collision risk with a car ahead or with a car in rear/lateral position in case of lane change manoeuvre).
As inputs, these Monitoring Functions will take into account, from one side, driving behaviours (i.e. actions on vehicle pedals and steering wheel, and visual scanning assessed through eye tacking measures) collected among the driver (real or simulated with COSMODRIVE) and, from the other side, situational parameters collected through car sensors (virtually simulated with Pro-SIVIC software).
By combining these two types of Monitoring Functions, risk-based analysis algorithms will be developed in the AdCoS in order to assess in real time the criticality of the traffic situation. From these diagnoses, a Centralized Manager will be in charge to monitor the Driving Aid sub-Systems integrated in the AdCoS and to adapt their Human-Machine Interaction modalities in accordance with the drivers' status (i.e. visually distracted or not), their behavioural errors (e.g. inadequate or risky manoeuvre implemented), and with the external collision risks with the other vehicles.

4.3.1.4 Solution Description

As the V-HCD platform will be supported by RTMAPS and Pro-SIVIC, a short description of these 2 tools should be provided in this section by INTEMPORA and CIVITEC.
Then, specific interfaces between these tools and COSMODRIVE & MOVIDA will be introduced.

4.3.2 Virtual HCD Platform description

To support the virtual design of future AdCoS based on MOVIDA functions, a tailored HF-RTP will be developed by IFSTTAR, in partnership with INTEMPORA and CIVITEC. This *Virtual Human Centred Design (V-HCD) platform* of AdCoS will be based on a COgnitive Simulation MOdel of the car DRIVEr (developed in WP2) to be interfaced (in WP4) with Pro-SIVIC (CIVITEC) and RT-MAPS (INTEMPORA) software. At last, the final objective is to have a V-HCD platform integrating (1) a human driver model using a "virtual eye" for road scene scanning, and able to drive (2) a virtual car (3) equipped with virtual AdCoS, for progressing in a virtual 3-Dimensional environment.

4.3.2.1 Partnerships for the Virtual HCD Platform

List of involved partners	
Partner	**Role**
IFSTTAR (IFS)	Research Partner, Driver Model, AdCoS-MOVIDA and COSMO-SIVIC main Developer
INTEMPORA (INT)	SME Partner WP4 & WP9, RT-MAPS support for V-HCD
CIVITEC (CVT)	SME Partner WP4 & WP9, Pro-SIVIC support for V-HCD
ERGONEERS (ERG)	SME Partner WP3 and WP9, Eye Tracking System (To be confirmed)
EADS-FR (EAD-FR)	Industrial Partner WP3, partnership on Monitoring Functions design (To be confirmed)
ENAC (ENA)	Research Partner WP4, partnership on Validation Methods (To be confirmed)

As the V-HCD COSMO-SIVIC platform is primarily focused on the virtual design and evaluation of AdCoS, collaborations with other WP9 partners, and more particularly regarding the two car Demonstrator developers (CRF and IBEO), are also open.

4.3.2.2 Tools

The COSMO-SIVIC V-HCD platform to be developed by IFSTTAR will be an integrative tool supported by RT-MAPS software functionalities. It will more precisely include:

- A Cognitive Driver Model with (COSMODRIVE), to be developed in WP2, including a virtual Eye (liable to be monitored by MOVIDA), and able to drive a virtual SIVIC car in a Pro-SIVIC environment
- A Virtual AdCoS (DAS + MOVIDA functions), to be designed in WP3 and WP4 (previously described in section 3), developed with Matlab/SateFlow tools and through RT-MAPS modules
- An Eye tracking system (for drivers visual distraction assessment)
- Virtual Sensors and a 3-Dimensionnal Road Environment, to be simulated with Pro-SIVIC software

- RT-MAPS as interfacing tool liable to be interfaced with tools and/or platforms developed by other partners. Could also support transfer towards real car demonstrators

Tool (Partner)	Use
COSMODRIVE (IFS)	COgnitive Simulation MOdel of the DRIVEr
RT-Maps (INT)	Integrative software to support DAS and AdCoS
Pro-SIVIC (CVT)	Road Environment & Virtual Sensors for AdCoS simulation
Tobii/Dikablis (ERG)	Eye tracking systems (analysis of drivers' visual scanning)
MOVIDA-AdCoS (IFS) based on MatLab, StateFlow, RTMaps modules and Pro-SIVIC	Virtual AdCoS simulation, including Monitoring functions of diver Visual Distraction and risks Assessment

4.3.2.3 Use of COSMO-SIVIC for AdCoS virtual design: Linear workflow

Figure 26 provides an overview of the virtual Human Centred Design process of AdCoS, and the linear workflow to be supported by COSMO-SIVIC, as an example of tailored HF-RTP platform in WP9.

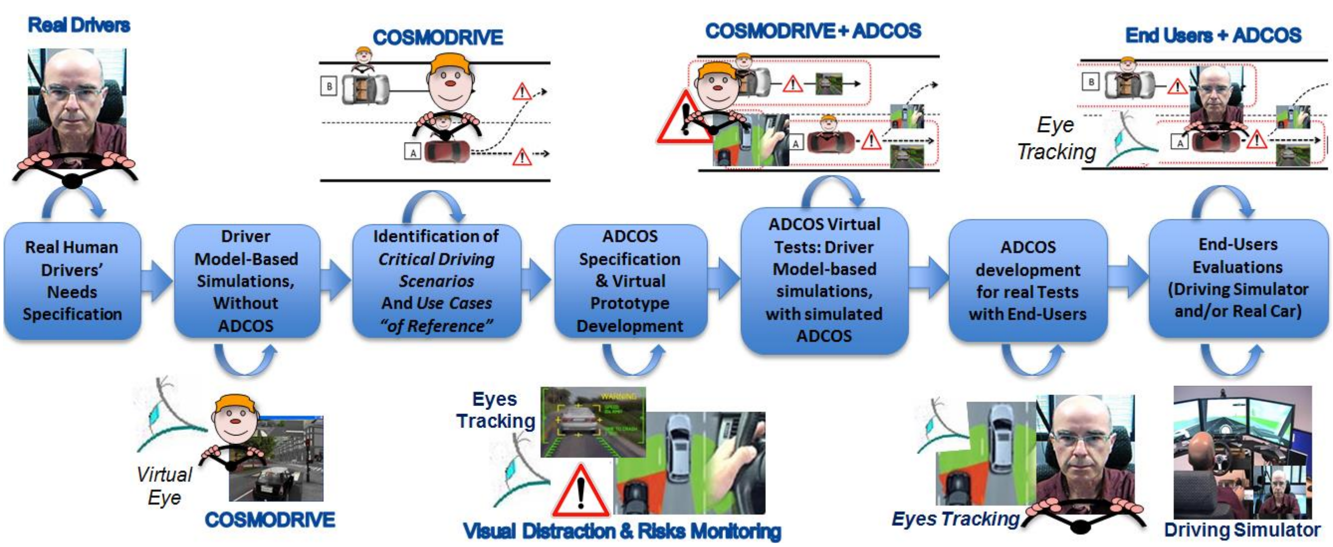


Figure 26 - Virtual design method of AdCoS with the COSMO-SIVIC platform

From real needs collected among human drivers (or from a literature review), simulations based on COSMODRIVE model will be implemented at two mains levels (identification of critical scenarios and use cases of reference + virtual ADCIOS tests), in order to virtually design and progressively evaluate the AdCoS supported by Monitoring functions of driver's visual distraction risks (MOVIDA). Advantages of using a Human operator model in this design process are to consider end-users' needs since the earliest stages (even if not any AdCoS prototype is available), and also to investigate driving scenarios and AdCoS functioning in a systematic way (that was not possible or highly expensive to do among real drivers).

4.3.2.4 Use of COSMO-SIVIC : "V design process" of AdCoS

Figure 27 presents the expected use of the COSMO-SIVIC in WP9, in a "V design process" of AdCoS based on MOVIDA.

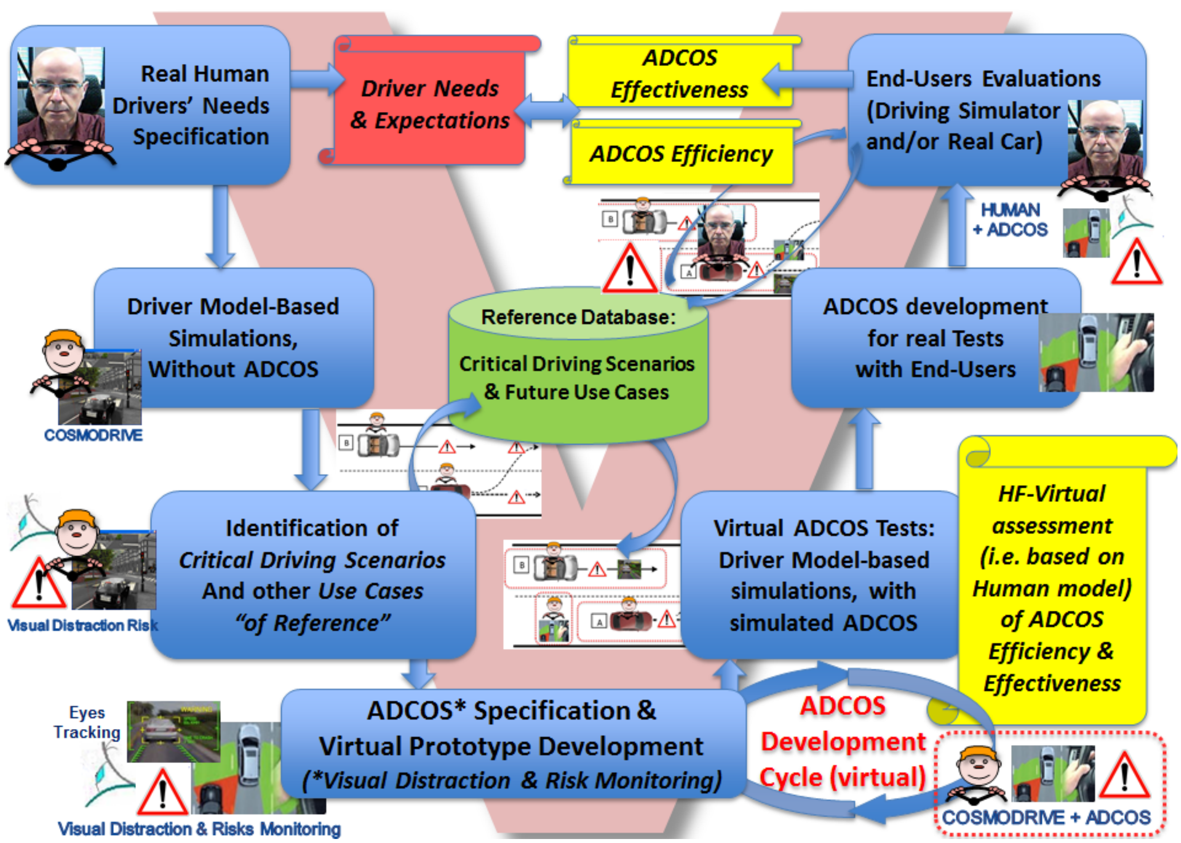


Figure 27 - V-Design process of AdCoS with the COSMO-SIVIC platform

In the frame of WP9, this integrative COSMO-SIVIC “V-HCD platform” based on a Human Driver model will be used to both simulate driving performances of a human drivers With and Without driving aids (from normal behaviours to critical behaviours due to visual distraction) in order to virtually support the AdCoS and MOVIDA functions design process, at 2 main levels.
At the earliest stages of the design process, COSMODRIVE-based simulations will be used to estimate human drivers’ performances and risks in case of unassisted driving, in order to identify critical driving scenarios due to visual distraction for which a given AdCoS based on MOVIDA could support them, in an adaptive and cooperative way. These critical scenarios will correspond to traffic situations for which the visual distraction could critically impact the human drivers’ reliability, and then increasing the risk of accident. Through these simulations, it will be possible to provide ergonomics specifications of human driver needs, as a set of “Critical Scenarios” and “Use Cases” of reference, liable to be stored in a “reference database”.
During the virtual design process of the MOVIDA-AdCoS, this reference database associated with visual distraction simulations based on COSMODRIVE, will be used to progressively increase its *efficiency* (i.e. AdCoS developments & virtual tests Cycle on the figure) in accordance with the different variations of the critical scenarios previously identified. Such COSMODRIVE + AdCoS based simulations will also allow the designer to assess the potential *effectiveness* of MOVIDA, before developing a real prototype and then testing its effectiveness among real human drivers, through full scale tests with end-users implemented on driving simulators and/or with real cars (final stage of the design process).

4.3.2.5 Requirements

To support the V-HCD COSMOSIVIC platform, several requirements were done by IFSTTAR among HoliDes partners / tool developers (like WP9_IFS_AUT_REQ03: “Data synchronization coming from different simulation tools”; WP9_IFS_AUT_REQ04: “Having a virtual car able to be dynamically piloted by the driver model", WP9_IFS_AUT_REQ10 (REQ12): Virtual simulation of car sensors (radar, camera, telemeter), as components of AdCoS, to be simulated and tested in WP9). To support AdCoS based on MOVIDA functions, a collaboration with ERGONEERS is under discussion, in order to use the virtual eye of COSMODRIVE for generating Eye Tracking data liable to be processed in Ergoneers software; cf. WP9_IFS_AUT_REQ07: "Recording/using of eye-tracking data to assess driver' visual distraction").

4.4 The TAK Simulator Demonstrator - Development and model-based evaluation of an adaptive HMI

4.4.1 Partners

List of involved partners	
Partner	**Role**
TAK	Industrial Partner WP9, ADCOS Developer
OFF	Research Partner WP2, Driver Model Developer
ERG	SME Partner WP3 and WP9
INT	SME Partner WP9
TWT	SME Partner WP9

4.4.2 AdCoS Description

The system under investigation is a combination of Adaptive Cruise Control (ACC) and Lane Keeping Assistance (LKAS).

- The targeted Use-cases are WP9 TAK1-6 (as described in Deliverable D9.1), which deal with approaching slower lead vehicles and overtaking situations on a two-lane German Autobahn scenario.
- The system can be driven in different modes of automation according to BASt (Gasser, Westhoff 2012), i.e. assisted, partially automated, highly automated. TAK is going to develop an HMI for an adaptive, cooperative driver assistance system based on the two basic components ACC and LKAS.
- The adaptation of the HMI will be based on driver state and driving situation.

4.4.3 Tools

Tool	Use
SILAB	Driving Simulator Software
CASCaS	Driver Model
RTMaps	Synchronising different data streams
Dikablis	Eye tracking system to assess driver state
Distraction Model	Distraction Model

4.4.4 Specific Design and Analysis Problems

Early testing of system designs (in this case an HMI) is important to reduce the risk of undesirable development. A lot of different techniques exist for HMI development to incorporate end-user feedback as early as possible in the system design process, e.g. questionnaires, pen and paper sketches of an HMI or function exploration using theater systems (a human instructor plays the role of the assistance function in a simple driving simulator environment).

Empirical studies to evaluate concrete HMI prototypes within a driving simulator require software and hardware implementations. Dedicated prototyping hardware (e.g. dSPACE) offers the possibility to test software functions, e.g. how the system interacts with a CAN Bus system typically deployed in a real car. For HMI designs such automated tests cannot be achieved easily, because the counterpart of the system (the user) has to be simulated as well. Automated test scripts can be used to simulate some interaction, for example button presses. However, this technology is not sufficient to simulate actual interaction with the user.

TAK seeks for a pure software simulation of the HMI in a driving simulator software environment to evaluate alternative configurations based on defined measures.

Over the last five years, OFF has developed a driver model which is already able to basically deal with the intended scenario of WP9. The model can simulate free-flow, car-following and lane changes right and left on a two-lane German Autobahn with medium traffic density. The model simulates gaze behavior including a mirror view which will cover blind spot problems. Output of the model contains for example task interaction time and gaze behavior traces. Additionally, the impact of secondary tasks on for example distance keeping or lane changing could be analysed. The existing OFF driver model can be used as basis within HoliDes and will be extended in “WP2 Task 4 Human Operator models” to fit the needs of TAK.

In order to validate the results derived from applying the CASCaS-model to the alternative HMI-solutions, user tests will be conducted. The validity of the model will be assessed by comparing the results of the user tests and the model runs. A prerequisite of this task is the availability of real human data and simulated – or modelled – human data. For the user tests such data will be made available by using Dikablis and a distraction model. The data produced by these sources will be integrated and combined together with driving data by using RTMaps.

The CASCaS model will be used to simulated driving behavior in the given scenarios of Use-cases WP9 TAK1-6

- Specific evaluation questions
    - Simulate gaze duration and reaction time necessary to gather information of the alternative display solutions
    - Take into account the interaction between traffic situation and driver state
    - Simulate impact on driving performance in given traffic situations
    - Optional: configurable driving styles
    - Pre-defined driver behavior: this can be compared to a "manual failure injection" technique which forces specific errors to evaluate their safety impact. The existing driver model behavior can be modified to specifically test HMI problems, e.g. removing certain actions, like mirror glance.

4.4.5 Solution Description

To tackle the problems described above, all partners will have to work on a common workflow which integrates driving simulator software, system development tools, hardware components and the driver model into an integrated simulation platform. An intended workflow diagram can be seen in the next section. The solution needs to tackle a number of different integration issues

- Integration of the assistance system into the integrated platform. One solution could compile the source code of the system into a dynamic link library (.dll) which can be loaded directly by SILAB. OFF has experience in generating those so called Digital Processing Units (DPU). Typical toolchains including model based tools like SCADE or Matlab Simulink could be used for system development to generate source code. This code can be embedded into a c-code wrapper to compile a DPU.
- The integration of the driver model into the SILAB software is already done. The specification of the exchanged data has to be adapted according to the new HMI.
- The combination and integration of the different tools into a single database will needs to be assured by using RTMaps.
- For system validation a test case suite has to be specified which is based on the scenario specification and the system requirements. For each test case a link to the simulation results has to be established.

- The design for the validation study has to be specified
- The validation study must be conducted, data must be analysed and the results must be discussed.

4.4.6 Solution Workflow: Linear

A simplified version of the workflow is shown in Figure 28.

Figure 28 – Linear workflow for TAK simulator demonstrator

4.4.7 Solution Workflow: non linear

The non-linear workflow is shown in Figure 29.

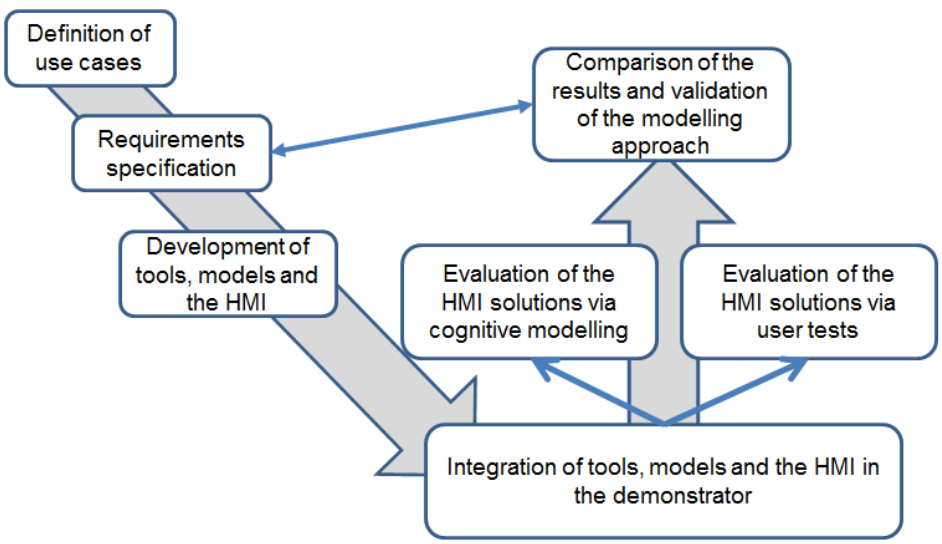


Figure 29 – Non linear workflow for TAK simulator demonstrator

The view for the cognitive modelling branch is further specified in Figure 30:

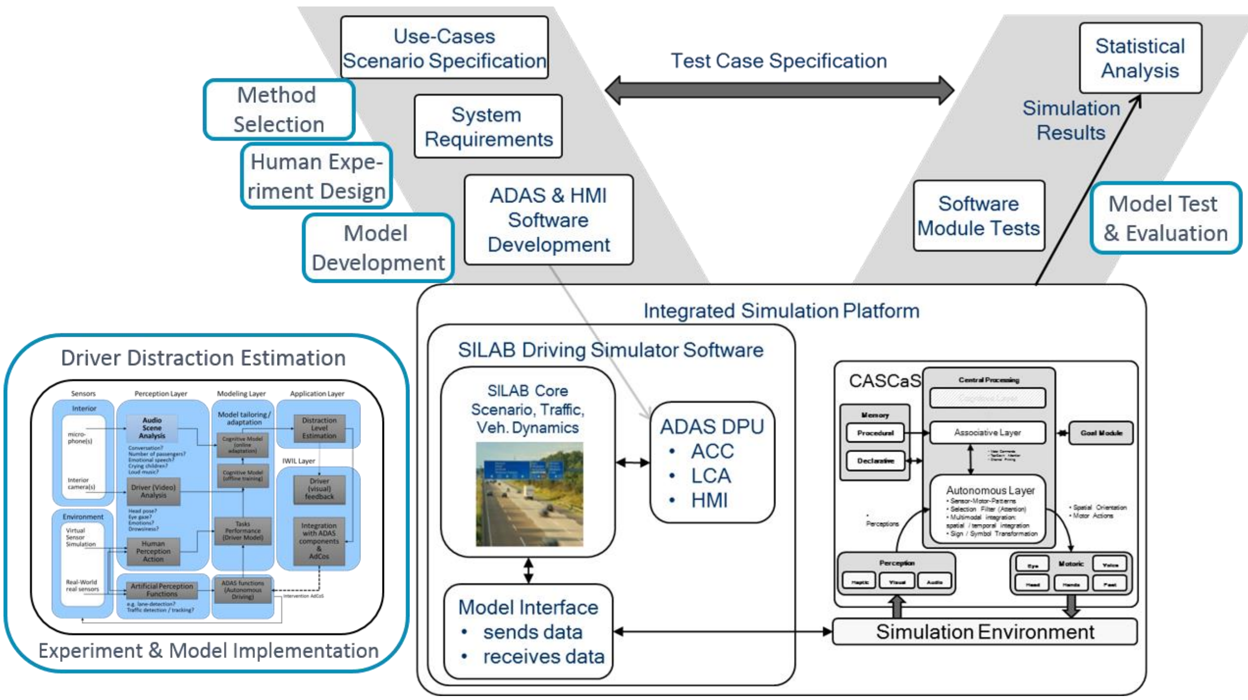


Figure 30. Adapted AdCoS design from TAK/OFF including the driver distraction estimation from TWT.

4.4.8 Covered Requirements

It is intended to cover all TAKATA requirements (WP9_[TAK]_AUT_REQ01 to WP9_[TAK]_AUT_REQ68).
Regarding CASCaS, special emphasis will be given to the requirements dealing with the blind spot indicator; here, especially WP9_[TAK]_AUT: REQ01 / 03 / 05-10 will be covered.

4.5 Further AdCoS components for the Automotive domain

The following components can be integrated into above AdCoSs but are not bound to one specific application.

4.5.1 Monitoring System Tool

4.5.1.1 Monitoring System description

A monitoring system has been developed to give a visualization in real time to end users, administrator users and different actors about specific uses cases that are given in the document "D9.1- Requirements Definition for the HF-RTP, Methodology and Techniques and Tools from an Automotive Perspective".
The Monitoring System (MS) will be connect to the HoliDes platform to send/receive data information from/to devices, actuators and the servers provided by different sources (vehicles, users, devices). In the use cases proposed for monitoring overtaking and cross-traffic scenarios, the MS allows monitoring the webcams installed in the vehicles, follow the routes by maps in 2D and 3D and a lists of the devices status provided by vehicles.
Pre-requisites:

- The system is developed to interoperate with AdCos (HoliDes communication module).
- A real overtaking or prototype simulation should exist. These sources will provide data to monitoring System.
- Data information should be homogeneous previously.
- Formatting messages for generic and specific domains (vertical and horizontal).

4.5.1.2 Components in Monitoring System

4.5.1.2.1 Software involved in Monitoring System

Software contains the logical part of the system to generate a web application, the logical layers involved in Monitoring System, and the components to communicate with others systems.
The software most important features are:

- Web services: Developing a logical Web Service Layer and their services, SendAlerts(), geoLocation(), SubscribeMS(), PublishMS().

- Publish Subscribe Layer: Develop a logical layer to communicate with a Publish Subscribe Server and publishers / subscribers via channels. All the vehicles included in the demo should added in a channel of Publish Subscribe Server.
- Database: Develop a logical layer to communicate the database with Web Service Layer. All the data providing by vehicles, devices will be stored in a database. Providing a historical data for the administrator users.
- Web application: source, frontend and frameworks used to build Web app. Maps, Locations, routes, driving simulators from routes given. Possibility to follow scenarios from laptops, tablets and other devices.
- Test tools: to test the application web and check that all is working fine.
- Software Tools: Google maps API, Google earth API, Rest services, Java JDK, HTML5, Install Publish Subscribe Server Rabbit MQ, Apache Tomcat Server, MySql Database server.

4.5.1.2.2 Hardware involved in Monitoring System

The Hardware involved in MS can be internal (provided by the own system) or external (provided by others systems).
Internal hardware:

- Web Servers
- Database Servers
- Servers providing Virtual machines
- Mobile, laptops, tablets

External hardware:

- Car devices
- Gateways
- Web cams

4.5.1.3 Functionalities in Monitoring System

Aimed functionalities in Monitoring System are:

Monitoring an overtaking use case: saving data from "overtaking use case", devices, use case actors, saving historical data and showing real time data

Monitoring a cross-traffic use case: saving data from "cross-traffic use case", devices, use case actors, saving historical data and showing real time data.

Real time video streaming: recording real time video streaming to follow an overtaking, cross-traffic, in a real case or prototype scenario.

Historical data by publish/subscribe channels: use case actors to follow publish/subscribe channels in the use cases. View data and filtered by data time, by periods, vehicles.

Common Web Services to give common functionalities to monitoring systems or external systems.

4.5.1.4 Users profiles in Monitoring System

Monitoring System could be managed and view for different users. A role for each user type is given, the main goals are:

- User driver, administrator monitoring system.
- User driver: has access to application overtaking and their specific channels to check an overtaking or a cross-traffic. Can view their data and update personal profile.
- Administrator Monitoring System: has access all control over all applications and database administration.

4.5.1.5 Application description

To explain this chapter first look at the diagram in Figure 31.

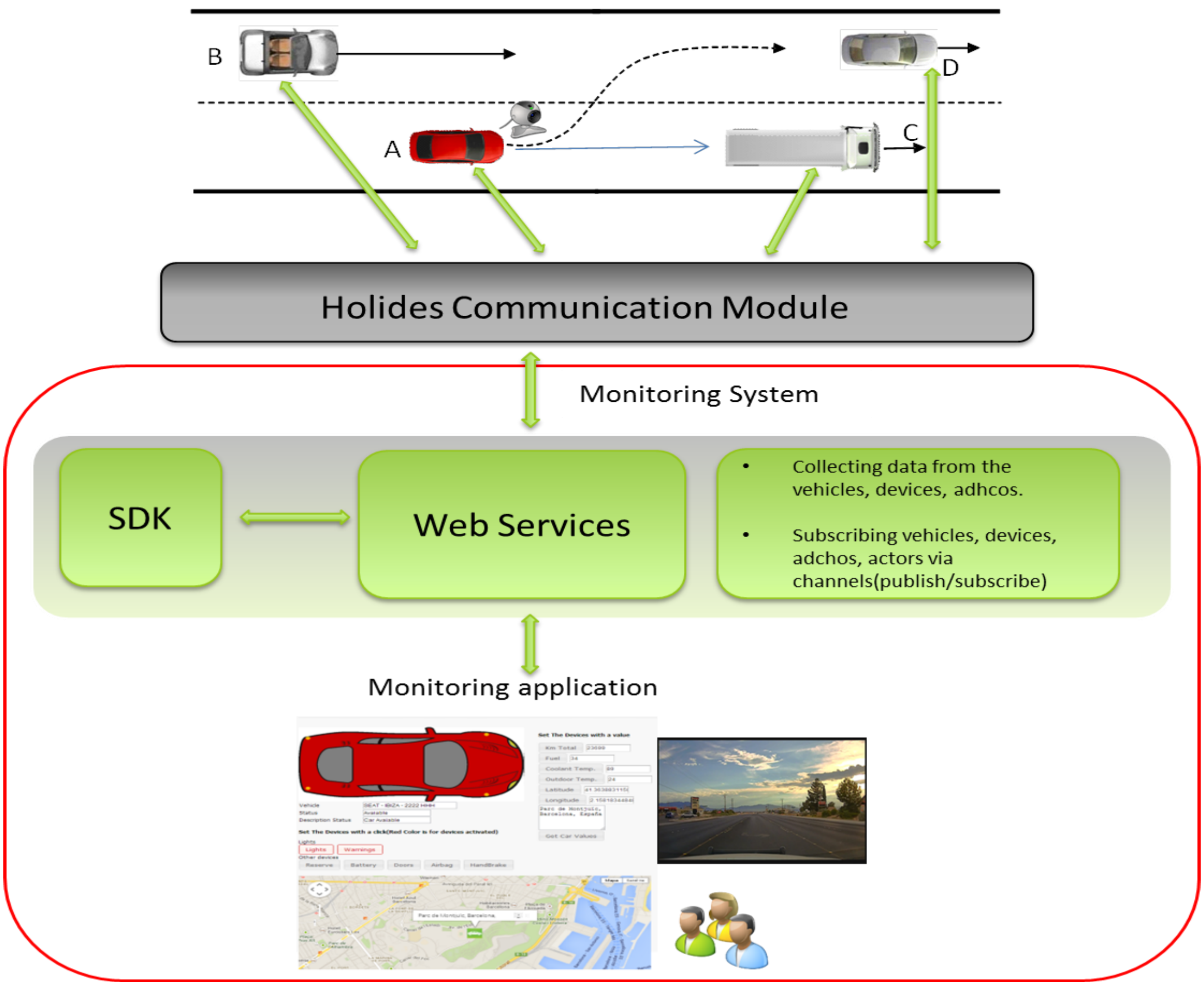


Figure 31 - Diagram describing the Monitoring System Application

The Application provides the web pages and the interaction with users to monitoring a specific overtaking, a map route given, status devices, from one or various vehicles.
It's created to view and manage the data provided for an eventual overtaking, giving a visual situation of a real scenario:

- Graphical diagrams from scenarios.
- Map locations showing the real location of the vehicles.
- Real time video and backup videos to follow a specific situation in a data time given.

- Different user roles have access to different webs.
    - E.g: administrator user can manage all data and a vehicle user and can follow it data during an overtaking manoeuver.

4.5.1.6 System components and overview

System components are the keys of the architecture Monitoring System:

WSL - Web services layer
PSL - Publish /subscribe layer
SL – Security layer
DBL – Data base layer
APP – Web application(front end)

4.5.1.6.1 WSL - Web services layer

A services layer is an abstract layer containing common services used by different uses cases called from HoliDes communication Module or other layers (SL, DBL, PSL) and providing data information to APP monitoring system.
Built in CRUD (Create, read, update, delete), using HTTP protocol, providing GET or SET services for generic scopes.

4.5.1.6.2 PSL - Publish /subscribe layer

The publish/subscribe layer allows to send information between subscribers of a channel via message–oriented middleware system.
Publisher: Senders of messages
Subscribers: Receivers of messages
Channel: The topic to which the receivers of messages have subscribed.

4.5.1.6.3 SL - Security Layer

This Layer contains:

- User Roles: services for users with rights for applications.
- Functionalities for Authentication/authorization, Login and register functions for user to access to applications.
- HTTP Protocol security: communications protocol secure

4.5.1.6.4 DBL – Database Layer

This layer contains functionalities to save data.

- Each data sent or received from the actors should be saved in a structure for storing historical data.

- Different kinds of data to store: common data structures, audio visual content.

4.5.1.6.5 APP –Monitoring System Applications

Web Applications developed:

- Responsive designed for different kinds of devices: laptops, mobile phones, tablets.
- It doesn't need installation previously, web connection and user/password register you can start to use.
- Graphics environment, interactive maps, historical data, webcams in real time.

Figure 32 shows the architecture diagram: shows the connection between monitoring system and HoliDes Communication Module detailing each layer in monitoring system and the connections between them:

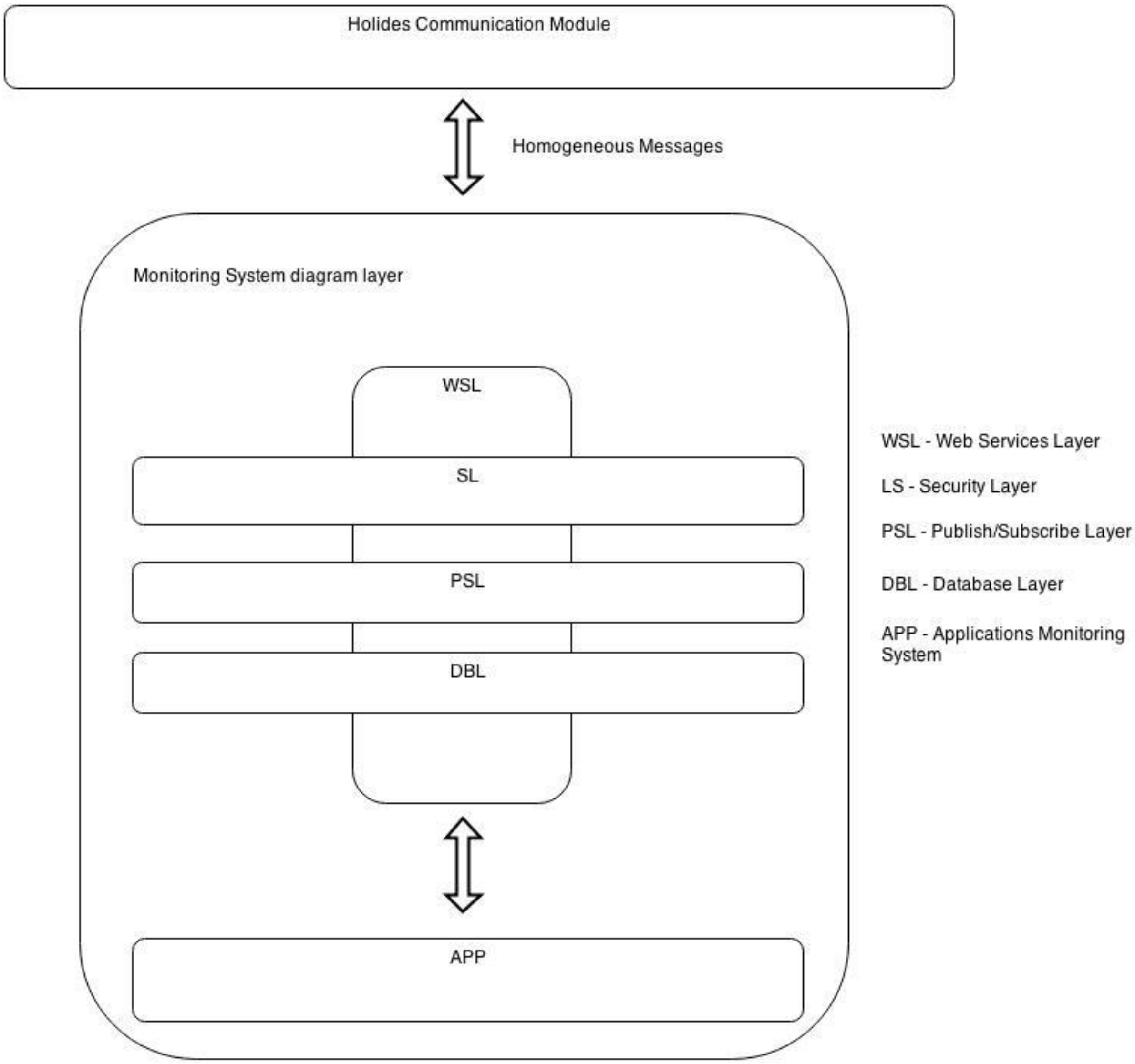


Figure 32 - Monitoring System Architecture

4.5.1.7 Modules - Input and outputs

WSL – the main goal of this layer is to communicate Monitoring System with HoliDes communication module, also has communication with the rest of layers SL, PSL, DBL and APP to give services to them. All the connections are I/O.

For example: write and read in a database (communication with DBL)

SL – Security Layer – communicates with WSL. All the communications are I/O.

PSL – Publish and subscribe layer have connected with a PB Server, this one manage the queues and the messages synchronism. WSL and SL have communication I/O with PSL.

DBL – Database layer has connections I/O with WSL.

APP – The main goal is to show and manage all data and for it, its Applications are connected to WSL.

4.5.1.8 Functional requirements

Taking the list of functional requirements gives in WP9:

Functional requirements should accomplish these features:

- Communicate with vehicles devices (ex: Blind-spot visual indicator, visual noise from irrelevant devices, ACC indicator, ACC warnings indicators, Haptic indicator).
- Different actions by user roles.
- Possibility to work in different domains: automotive or others.

Functional requirements:

- Portability: Possibility to see some messages in different devices.
- Security:
    - Authentication and authorization to access to WEB pages.
      Login for user registered and authorization to give specifics roles.
    - Web services authorization: Include authorization in the web services headers.
    - Data Message encryption: encrypt data in the body of the messages.
    - Framework security: Include framework security.

- Interoperability:
    - Same format messages for different devices.

- Management of historical data:
    - Management of historical information collects from different sources: AdCoS, devices, vehicles, users and others.

4.5.2 Detection of Driver Distraction based on In-Car Measures

4.5.2.1 AdCoS Introduction

Name: Detection of driver distraction based on in-car measures (Sub-AdCoS / partial AdCoS usable by all WP9 AdCoS Use Cases)

Partners: TWT, and others

Used HoliDes MTTs (subject to change):

- *Detection of operators' head orientation (BUT)*
- *Detection of driver distraction based on data on vehicle dynamics (UTO)*
- *RTMaps*
- *Pro-SIVIC*

Other MTTs:

- *Cognitive Model (detailed technique still open)*
- *OpenDS*
- *Visual Studio*

Distraction during driving leads to a delay in recognition of information that is necessary to safely perform the driving task (Regan und Young 2003). Thus, distraction is one of the most frequent causes for car accidents (Artho et al., Horberry et al. 2006). Four different forms of distraction are distinguished, although not mutually exclusive: visual, auditory, bio-mechanical (physical), and cognitive. Human attention is selective and not all sensory information is processed (consciously). When people perform two complex tasks simultaneously, such as driving and having a demanding conversation, the brain shifts its focus. This kind of attention shifting might also occur unconsciously. Driving performance can thus be impaired when filtered information is not encoded into working memory and so critical warnings and safety hazards can be missed (Trick et al. 2004). Sources for distraction of the driver can be located within and outside of the car.
The goal of this AdCoS system is to make optimal use of a rich set of redundant and complementary cues for driver distraction.

4.5.2.2 Description of Strategy / Problem Statement

Deriving knowledge about the human operator can be very valuable already in the system design and testing phase. During interaction with a system,

e.g. while driving a car, or even, while being driven or supported by an autonomous system, the operator's degree of distraction can be estimated.
Such a tool bears the potential to be used online to estimate the driver's distraction not only during testing of a prototype, but also during everyday interaction with the AdCoS. This online measure of distraction could in turn be used to adapt the degree of automation of the AdCoS to the driver's state.
A combination with the tools developed by BUT (Detection of operators' head orientation) and UTO (Detection of driver distraction based on data on vehicle dynamics) is possible to increase the tool's predictive power.
This method can be applied to the frontal collision use case and/or to the overtaking use case of WP9.

4.5.2.3 Solution Description

Acoustic scene analysis comprised of the detection of the number of speakers, the degree of emotional content, information about the driver's involvement in the conversation (e.g., whether the driver himself is speaking), is to be employed for the prediction of the driver's degree of distraction. In addition, eye-tracking signals, such as temporal measures of eye movements, and face movement information, such as mouth movements, can be exploited to increase the reliability of distraction prediction. A computational and empirical cognitive distraction model is used for analysing the different signals, with the aim of computing a 'distraction degree' of the driver. The effect of cognitive distraction, based on different audio scenarios, on driving performance will be empirically tested in a parallel task in order to assess the impact of auditory stimuli on distraction.
For this, in-vehicle information is needed. This includes, but it is not limited to, in-car audio recordings and behavioural data from the driver. These data need to be stored in a way that enables linking them to certain system states, e.g., inputs from the user to the system. Thus multimodal data integration and synchronization is mandatory for the tool to produce meaningful results.
The tool provides a temporally evolving estimate of driver distraction. The metrics used to quantify the driver's distraction based on in-car information are developed in T5.2. The audio-component is developed in WP2. The different measurements will be integrated in RTMaps provided by INTEMPORA. Virtual Environment simulations could be provided by, e.g. CIVITEC (providing also virtual sensors, e.g. Ibeo/CRF Use-Cases

(Autonomous Driving), besides a virtual simulation environment for human-driving).

Personal components of the cognitive model and computations are intended to be mobile, e.g., via a Smartphone App. The core of the App, the personal model, should be exchangeable between the mobile device and an on-board system.

Figure 33 gives an overview of an AdCoS containing the described tool for driver distraction estimation:

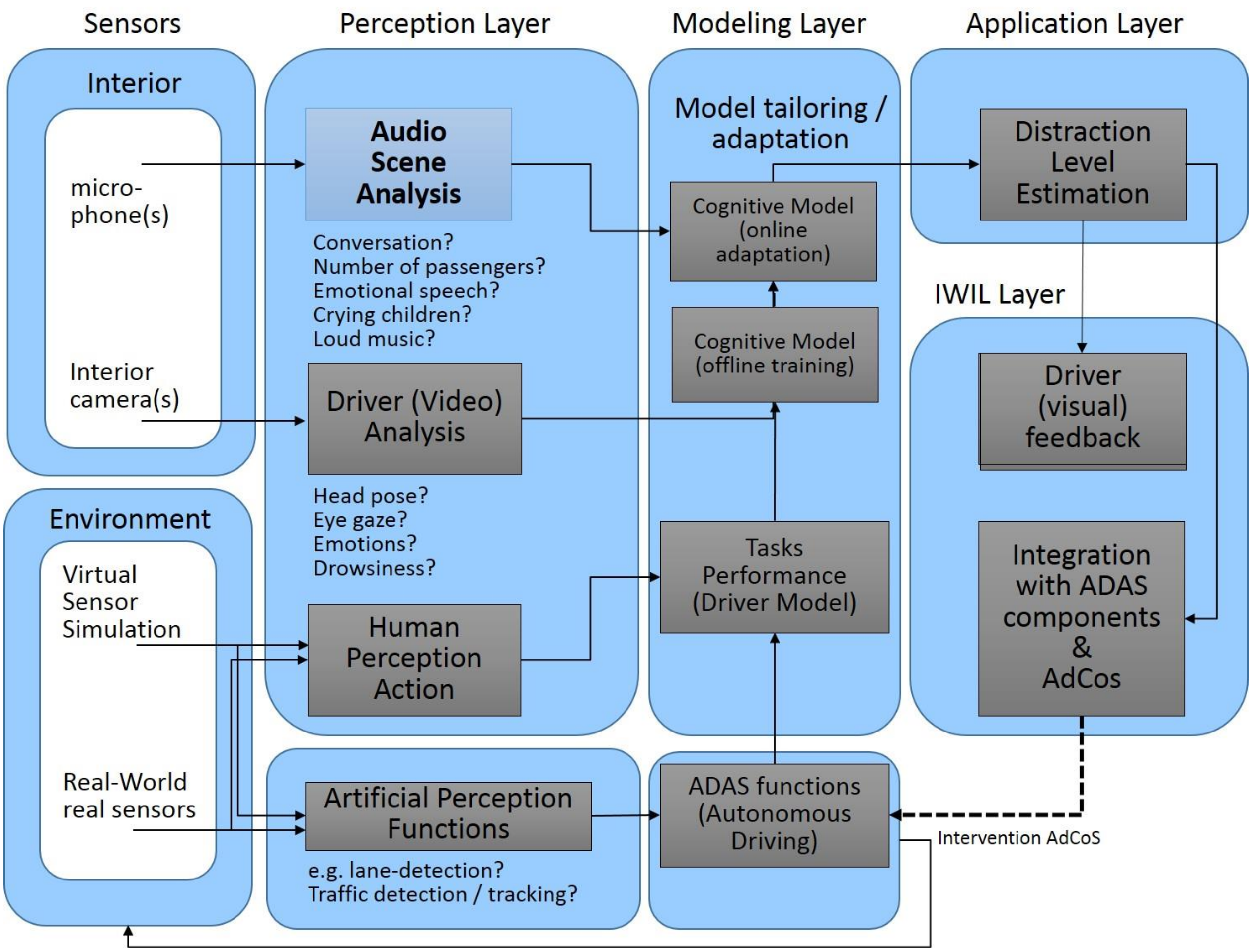


Figure 33 - TWT AdCoS design for audio-based driver distraction estimation integrated with other AdCos system components under development.

4.5.2.4 Solution Workflow: Linear

The following design workflow will be used for the integration of the distraction model into the HF-AdCoS (Figure 34).

Figure 34 - Linear solution workflow for the integration of the distraction model into the AdCoS.

4.5.2.5 Solution Workflow: non linear

The following process activities with specified tools show how the different applications are expected to be used in the development of the AdCoS (Figure 35).

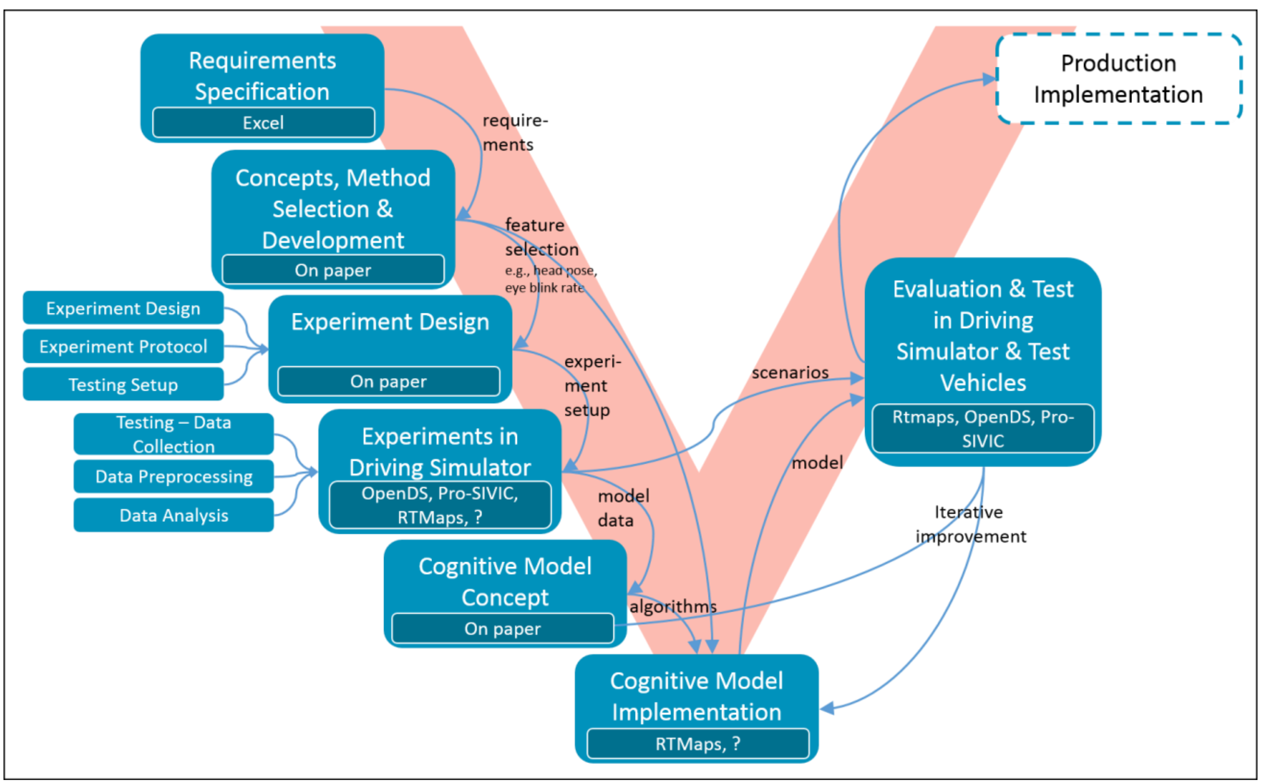


Figure 35 - Non-linear solution workflow with potential tools to be used for the development of the AdCoS containing the driver distraction estimation.

The tools we expect to use are mostly HoliDes technologies, as described in the figure, but might also be combined with other technologies if necessary or advantageous.
Activities specific for the HF-domain are those related to the experimental design, the testing procedure, data analysis as well as the identification of data predicting the degree of the driver's distraction. On the basis of this data the cognitive model will be implemented and evaluated and validated using simulator experiments. In case the autonomous system functions of other AdCos systems will be implemented with the virtual reality simulators (virtual sensors, like in a combination of RTMaps and CIVITEC) the transition from simulated to real-world autonomous driving would be facilitated. Working with the same simulation environment offers the chance to develop human-centered system functions around autonomous functionalities.

4.5.2.6 Covered Requirements

Table 2 describes the list of covered requirements from TWT. When integrating the driver distraction estimation module into the AdCoS planned by TAK/OFF, CRF, or IBEO, the list will need to be adapted accordingly.

REQ-ID	Name
WP9_TWT_AUT_REQ01_v0.1	Recording instruments/microphones to be installed inside the car/ simulator
WP9_TWT_AUT_REQ02_v0.1	Algorithm for analyses of sound signals
WP9_TWT_AUT_REQ03_v0.1	Car simulation with operating vehicle speed range and different route options
WP9_TWT_AUT_REQ04_v0.1	Algorithm for analyses of distraction
WP9_TWT_AUT_REQ05_v0.1	Recording instrument
WP9_TWT_AUT_REQ06_v0.1	Actors for simulation of real background conversation/sounds
WP9_TWT_AUT_REQ07_v0.1	Eye-tracking instrument
WP9_TWT_AUT_REQ08_v0.1	Algorithm for analyses of eye movements
WP9_TWT_AUT_REQ09_v0.1	Behavioral pre-studies of planned testing conditions
WP9_TWT_AUT_REQ10_v0.1	Head Pose Detection/Tracking
WP9_TWT_AUT_REQ11_v0.1	Facial mouth movement detection
WP9_TWT_AUT_REQ12_v0.1	Automatic external context cues, like lane following/ changing behavior
WP9_TWT_AUT_REQ13_v0.1	Synchronization of various measurement

	sources
WP9_TWT_AUT_REQ14_v0.1	Feedback rendering modality of estimated distraction level

Table 2 - List of covered requirements for the AdCoS containing the driver distraction estimation.

5 Conclusions

The present document has provided a full and updated picture of the work done so far in WP9, HoliDes applicative work package on the Automotive domain.
In particular, after the definition of requirements and use cases in the Automotive domain (as detailed in D9.1), the involved partners have started the tailoring process of the HF-RTP (v0.5) defined in WP1. More specifically, the AdCoS to be developed have been defined in details, as well as the test-vehicles and driving simulators to be used for their qualification. For each AdCoS, a specific tool chain has been defined, following a design process that takes into account human factors issues and safety improvement.
Finally, it has to be pointed out that D9.2 is just the first step in tailoring the HF-RTP to the Automotive domain, where a first attempt to harmonise and linking all the aspects has been made. Nevertheless, we realise that the link with the first release of HF-RTP (v0.5) is still to be improved. This point will be more specifically covered during the cycles 2 and 3 of the project, for version 1.0, 1.5 and 2.0 of HF-RTP, i.e. in the next deliverables concerning the tailoring of the HF-RTP (D9.4, D9.6 and D9.8 for WP9).

6 References

Gasser, T.M., Westhoff, D. (2012) BASt-study: Definitions of Automation and Legal Issues in Germany. Road Vehicle Automation Workshop (TRB), 24-26. July 2012, Irvine, California.

Griesche, S., Dziennus, M. (2013): Images in mind – Design metaphor and method to classify driver distraction in critical situations. Contribution to 7. VDI-Tagung "Der Fahrer im 21.Jahrhundert", 5.-6. Nov. 2013 Braunschweig, Germany, In: VDI-Berichte 2205, pp. 85 – 99, VDI Verlag, ISBN 978-3-18-092205-8.

National Highway Traffic Safety Administration (NHTSA), 2013, "Preliminary Statement of Policy Concerning Automated Vehicles", Washington, USA.