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

D9.3 - Requirements & Specification & first Modelling for the Automotive AdCoS and HF-RTP Requirements Definition Update (Feedback)

Project Number:	332933
Classification:	Public
Work Package(s):	WP9
Milestone:	M2
Document Version:	V1.0
Issue Date:	16.02.2015
Document Timescale:	Project Start Date: October 1, 2013
Start of the Document:	Month 14
Final version due:	Month 16
Deliverable Overview:	**Main document:** Requirements,Specification,Modeling Automotive.pdf
Compiled by:	Fabio Tango, CRF
Authors:	Fabio Tango (CRF); Elisa Landini (REL); T. Bellet, J.C. Bornard, D. Gruyer, B. Richard (IFS); Jordi Fonoll Masalias (ATO); Mark Eilers (OFF); Gert Weller (TAK); M. Kraehling, S. Kaufmann (IAS); Stefan Griesche (DLR); M. Botta (UTO).
Reviewers:	Zdenek Moravek, HON Jeremy Sproston, UTO
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
05.12.2014	First structure presentation (version 0)	Zdenek Moravek, Fabio Tango, Miriam Quintero.
19.12.2014	Draft version: structure presentation with WP9 contents	Fabio Tango
09.01.2015	Version with first partners' contributions and comments	Elisa Landini
14.01.2015	Version with other contributions (ATOS)	Jordi Fonoll Masalias
14.01.2015	IFSTAR contribution	T. Bellet, J.C. Bornard, D. Gruyer, B. Richard
15.01.2015	Added chapter 3.2.2 and references	Mark Eilers (OFF)
20.01.2015	Addition of TAK contributions	Gert Weller (TAK)
21.01.2015	Updated IAS and DLR Contributions	M. Kraehling, S. Kaufmann (IAS); S. Griesche (DLR)
26.01.2015	Contributions from UTO and Conclusions	M. Botta (UTO); F. Tango (CRF)
13.02.2015	Final version Comments from internal reviewers integrated.	F. Tango (CRF)

List of content

1 Introduction

The main objective of WP9 is the development and qualification of AdCoS in Automotive (AUT) domain using the tailored HF-RTP and methodology from WP1, to demonstrate the added value for industrial engineering processes, in terms of reduced cost, fewer necessary development cycles and *better functional performances*.
This report describes the requirements, specifications and the first modelling for the AdCoS applications in the Automotive (AUT) domain, with reference to the target-scenarios (TSs) and the Use-cases (UCs) described in the deliverable D9.1 "Requirements Definition for the HF-RTP, Methodology and Techniques and Tools from an Automotive Perspective". In particular, we mainly refer to the two AdCoS applications implemented on the real test-vehicles (TVs):

- **Adapted Assistance**, that is a Lane-Change Assistant (LCA) system, led by the CRF partner.
- **Adapted Automation**, that is an automatic Intuitive Driving (ID) system, led by the IAS partner.

In addition, this report includes the results of a first attempt to model the AdCoS using the HF-RTP and methodology utilising either pre-existing tools or new tools to be developed in the frame of the HoliDes project.

Section §2 contains a list of tools definitely applied from WP1-5. Section §3 describes each AdCoS use case including AdCoS operational definitions, HMI for the AdCoS, tools applied from the HF-RTP, requirements and specifications, and the system architecture. Section §4 reports on feedback from WP 1-5. Section §5 presents some conclusions and the next steps.

2 Tools and Services applied from the HF-RTP

The table below shows a list of all tools applied within Automotive Domain and it is taken from the Annual Project report.

Tool name	Tool type	Tool provider	AdCoS
Driver Distraction Classifier	HF Modeling Techniques and Tools (Machine learning prototype)	UTO, CRF	LCA
Probabilistic Driver Intention Recognition	HF Modeling Techniques and Tools (Probabilistic model)	OFF	LCA
Audio-Distraction	Algorithms / Tool	TWT	ID TAK
MOVIDA (MOnitoring of VIsual Distraction and risks Assessment)	HF Modeling Techniques and Tools & Algorithms for driver monitoring	IFS	LCA, AA (Adapted Automation)
Great SPN for Co-pilot MDP development	Techniques and Tools for Adaptation (SW framework and probabilistic model)	UTO	LCA
ProSIVIC	Model based HF techniques & tools (SW Car sensors simulator)	CIV	LCA ID
RT-MAPS	Model based HF techniques & tools (Software framework and ADAS simulation support tool)	INT	LCA TAK ID
V-HCD (Virtual-Human Centred Design) simulation platform (*COSMO-SIVIC)*	Model based HF techniques & tools (SW simulator of Driver + ADAS)	IFS	LCA, AA (Adapted Automation)
SURT	Empirical based HF techniques & tools	DLR	LCA
CASCaS	System/HMI evaluation	OFF	TAK
Dikablis	Eye-Tracker and analysis software	ERG	TAK

Table 1: tools and services from HF-RTP of WP1 and used in AdCoS applications of WP9.

3 Detailed AdCoS Description

In this section, a detailed description of the Automotive AdCoS applications is illustrated, considering the operational definition (which problem is addressed, which is the goal and the situations to be avoided, etc.), the HMI and the tools used for the applications development, as well as their requirements and specifications.

3.1 Operational Definition of the AdCoS

This section describes which problem(s) is intended to be solved by a specific AdCoS, what is the controlled entity of the related Use-case (UC) and which kind of interaction can be envisaged with the operator when performing a task (see also D9.1 for details) [NHT13].
In order to increase safety and traffic efficiency, an intelligent and adaptive support to the human is needed. In order to achieve this, three types of adaptive harmonized systems are developed:

- **Adapted Assistance**, which includes the generation of warnings/advice/information to the driver, depending on the external situations and the internal conditions (driver's intention and distraction).
- **Adapted Automation**, which investigates the shift of the longitudinal and lateral controls of the car from human to machine (and back) according to the capacity, load and intentions of driver (different Levels of Automation).
- **Adaptive HMI**, provided by TAKATA partner, which is a system that helps the driver to re-direct its attention to the road.

These AdCoS applications will be developed and qualified on demonstrators which can be test-vehicles (TVs) or driving simulators (DSs). The first ones are implemented by CRF and IAS partners, with the support of a dedicated team; the second ones are implemented by IFS, DLR, REL, TWT, ATO and TAK, in order to support the demonstrator vehicles or to prepare a specific and independent prototype (e.g. simulation of AdCoS behavior in critical and hence dangerous scenarios; definition and preparation of the HMI design; and so on).

For what concerns the target-scenario (TS) addressed, that is the problem statement for the AdCoS in AUT domain, it is sketched in the following figure. An agent (let's call it the Ego-Vehicle (EV), namely the RED car in the figure) is preparing to overtake 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.

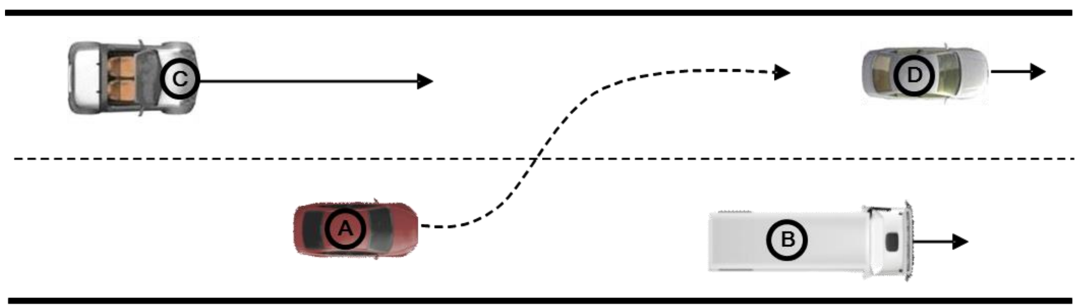


Figure 1: representation of the target-scenario (the problem that the AdCoS intends to solve) in AUT domain.

The precondition is that agent A is driving faster and approaching a slower vehicle (B) on a straight road.
The successful end-condition is that the Lane-Change (LC) manoeuvre – and then the overtaking (OV) – is performed without risks and without stop/strong speed reduction of EV (minimum change in traffic flow, namely the function has not to disturb traffic "too much").
The trigger event is that the vehicle with lower speed is driving in the same lane as the agent A.
We can guess the following conditions:

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

Considering the Agents, Tasks and Resources (ATR) framework, the first – the agents – are represented by the red vehicle, both the grey vehicles in the adjacent lane and the grey truck (machine agents), all with related drivers (human agents). For the Tasks, red vehicle aims at changing lane (i.e. to overcome the grey truck, which is the slower vehicle). The Resources are represented by the room and time availability in the adjacent lane to let the red vehicle act the lane-change manoeuvre (i.e. lane capacity); in addition, also vehicles braking/accelerating capacity are resources for the Multi-agent Systems (MAS); finally, the sharing of vehicle control between human and machine agents can be considered in this category as well. The following figure shows this situation, with reference to the adaptation concept (see D3.4 for details):

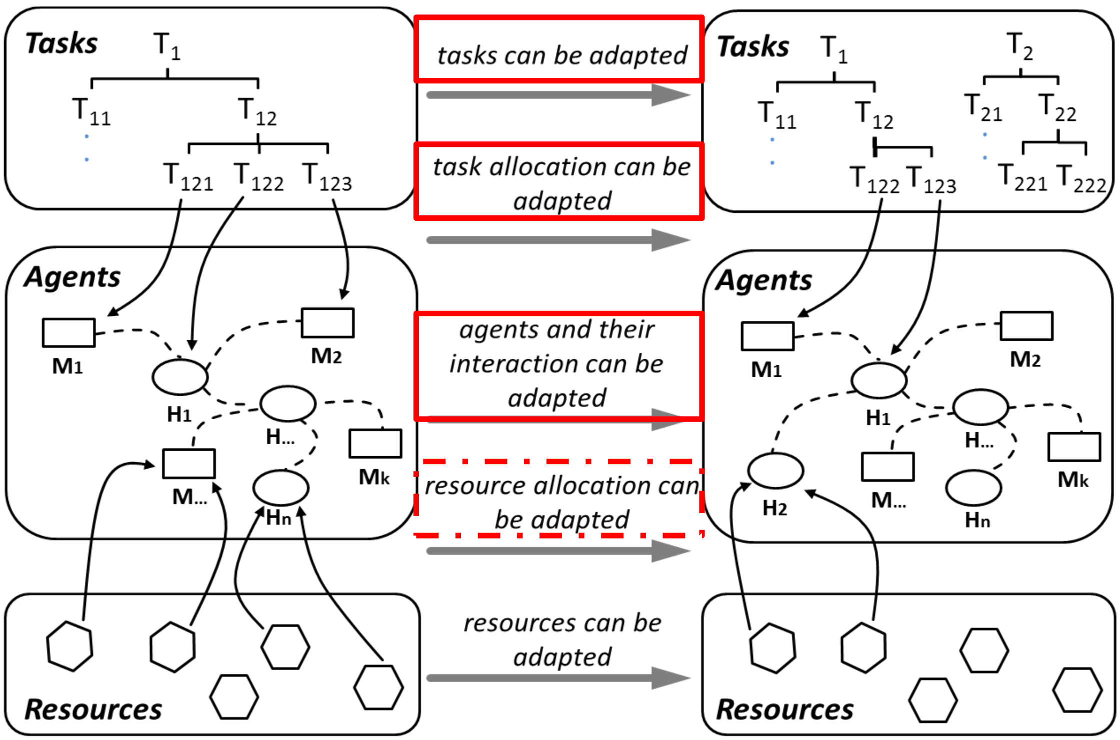


Figure 2: tasks, agents and resources in relation to adaptation for AUT AdCoS.

As shown in the figure, the adaptation can regard:

- tasks to be performed (their execution can be different depending on the state of the human-agent or of the machine-agent),
- the interaction among agents (e.g. different HMI channels depending on the state and on the content),
- the resource(s) allocation, based on different level of automation.

The first two points are common to all demonstrators, while the last one is specific for the Adapted Automation of IAS (this is the reason of the different line symbol in the figure).
With reference to figure 1, two main AdCoS have been selected.

In addition, there is also the Adaptive HMI AdCoS, provided by TAKATA partner, which is based on the description provided in Deliverable 1.3 (Development and evaluation of an adaptive HMI).

3.1.1 Adapted Assistance Use-cases

As previously mentioned, the Lane-Change Assistant (LCA) function can be regarded as a unique supporting system, adapting to the behaviour of the different agents, namely, it is able to adapt to the internal and external scenarios. This means that the "optimal" manoeuvre is suggested from machine-agent to human-agent, by means of specific warnings, advice and information, according to the visual or cognitive state and intentions of driver, as well as external environment. In the CRF Test-Vehicle (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).

For the real AdCoS of CRF, the basic idea is to adopt a statistical approach: the principle is to model our system as an MDP (Markov Decision Process), in order to construct optimal warning and intervention strategies (WISs).

In the IFS Driving Simulation platform (DS), the following functionalities (or ADAS) are simulated:

- Lane-Change Assistant (LCA) and Overtaking Assistant (OA)
- Forward Collision Warning (FCW), including Adapted Automation (AA) functions based-on automatic emergency braking and/or emergency lane change manoeuvre
- Rear / Lateral Collision Warning (LCW), including Adapted Automation (AA) functions based-on automatic emergency braking and/or Lane Change inhibition.
- Fully Automated Car driving (based on ACC, Lane Keeping Assistant and Adapted Automation functions).

For the IFS Simulated AdCoS (supported by RT-MAPS and ProSivic software), the aim is to design monitoring functions of Drivers' visual distraction, Risk-based analysis algorithms and a Centralized Manager of ADAS, in order to provide an adaptive and cooperative support system (based on warning or on vehicle control taking), specifically adapted to the current drivers state and the situational risk.

For these AdCoS applications, several use-cases can be envisaged. The following sequence-diagrams illustrate the situation, where the involved actors are the environment (that is the scenarios surrounding the EV), the vehicle itself (the red one, with reference to figure 1), the machine-agent (that is the Adapted Assistance system represented by the LCA in this case), the HMI (to guarantee the interaction between the human-agent and the machine-agent) and the human-agent (namely, the driver).

UC1 – Normal situation: driver intends to perform the lane change and initiates the manoeuvre.

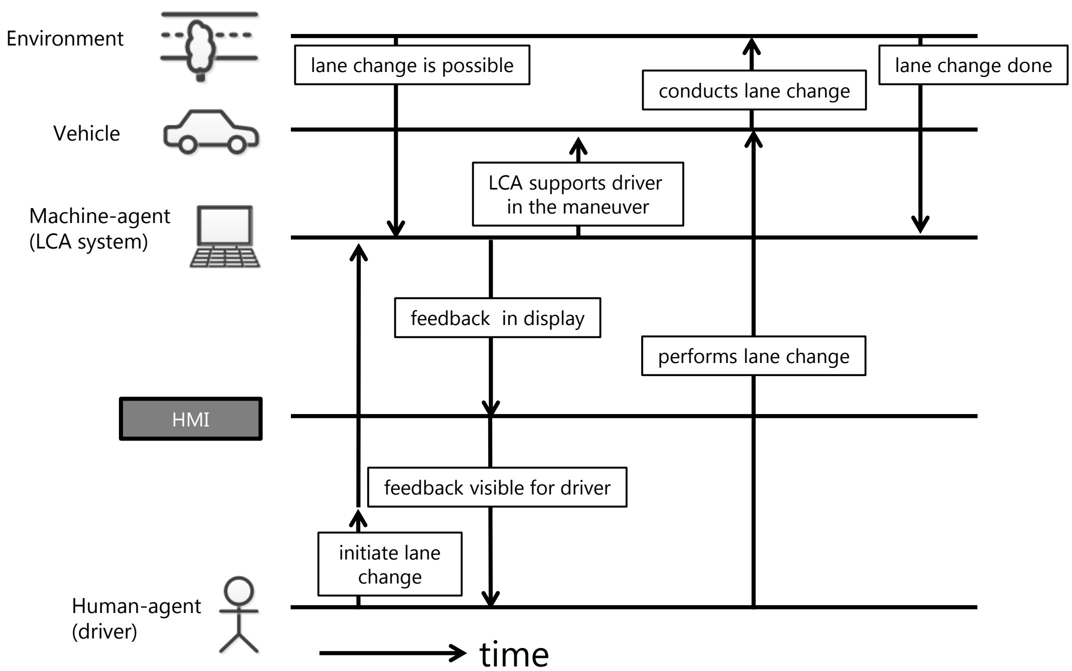


Figure 3: sequence diagram of LCA AdCoS for the use-case "Normal situation – LC possible".

This is the "standard situation": the human-agent is completely attentive (the system is monitoring the cognitive state), the lane-change is possible and thus the system can simply "pay attention" that the manoeuvre is correctly executed.

UC2 – Normal situation: driver intends to perform the lane change and initiates the manoeuvre, which is now not possible due to lane obstruction.

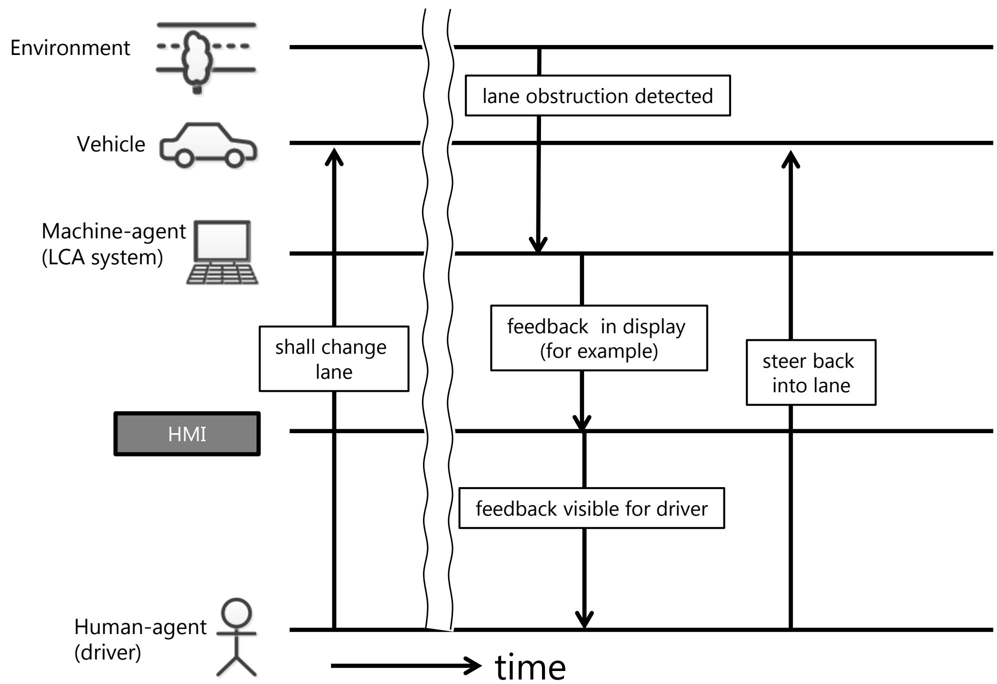


Figure 4: sequence diagram of LCA AdCoS for the use-case "Normal situation – LC not possible".

This situation is very similar to the previous one, but now the manoeuvre is not possible anymore, due to – for example – another vehicle approaching from the rear in the adjacent lane. The machine-agent (LCA system) detects the situation and supports properly the human-agent (the driver) who aborts the lane-change manoeuvre. Also in this case, the driver is attentive (monitored by the system).

The most interesting case is when the human-agent in the RED vehicle is distracted and thus the lane-change is performed without considering the vehicle approaching from the back. Of course, if the driver state is OK, the situations are the same, as described by figure 3 and 4.

UC3 – Impaired driver: driver intends to perform the lane change and initiates the manoeuvre; the driver is impaired, but responding to the HMI.

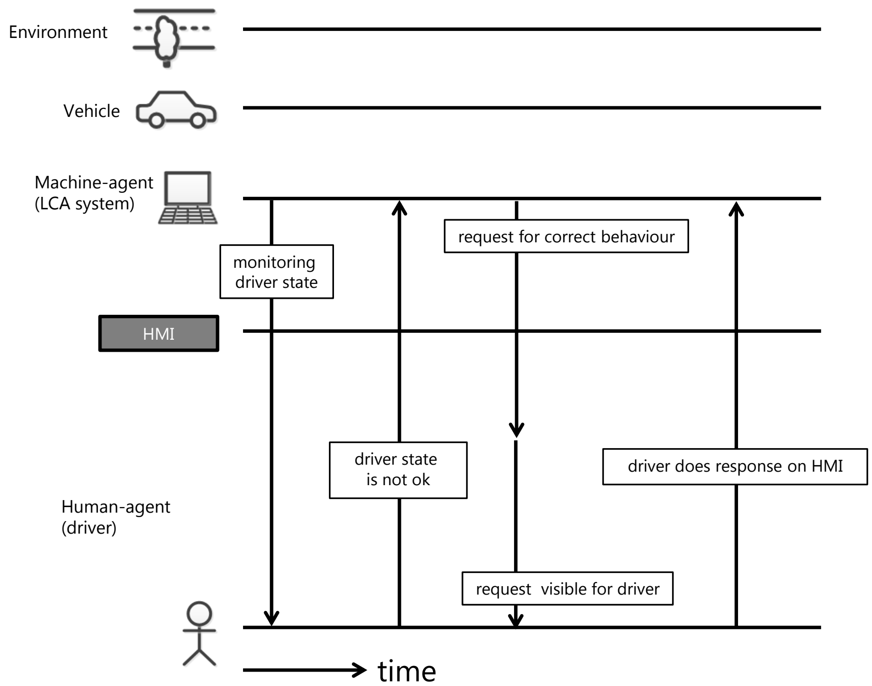


Figure 5: sequence diagram of LCA AdCoS for the use-case "Driver Status – Distracted, but responding"

In this situation, the machine-agent is monitoring the driver, who is distracted; therefore, when he/she intends to perform a lane-change (for overtaking), the system requests for a correct behaviour[2] before supporting this manoeuvre. After that, UC1 or UC2 can occur (see figures 3 and 4).

UC4 – Impaired driver: driver intends to perform the lane change and initiates the manoeuvre; the driver is impaired and does not respond. A minimum risk manoeuvre is necessary.

[2] At the moment, with "correct behaviour", we mean a behaviour without risky situation. For example, for the longitudinal aspect, this can be related to the Time-To-Collision (TTC) values, which have to be under a given threshold.

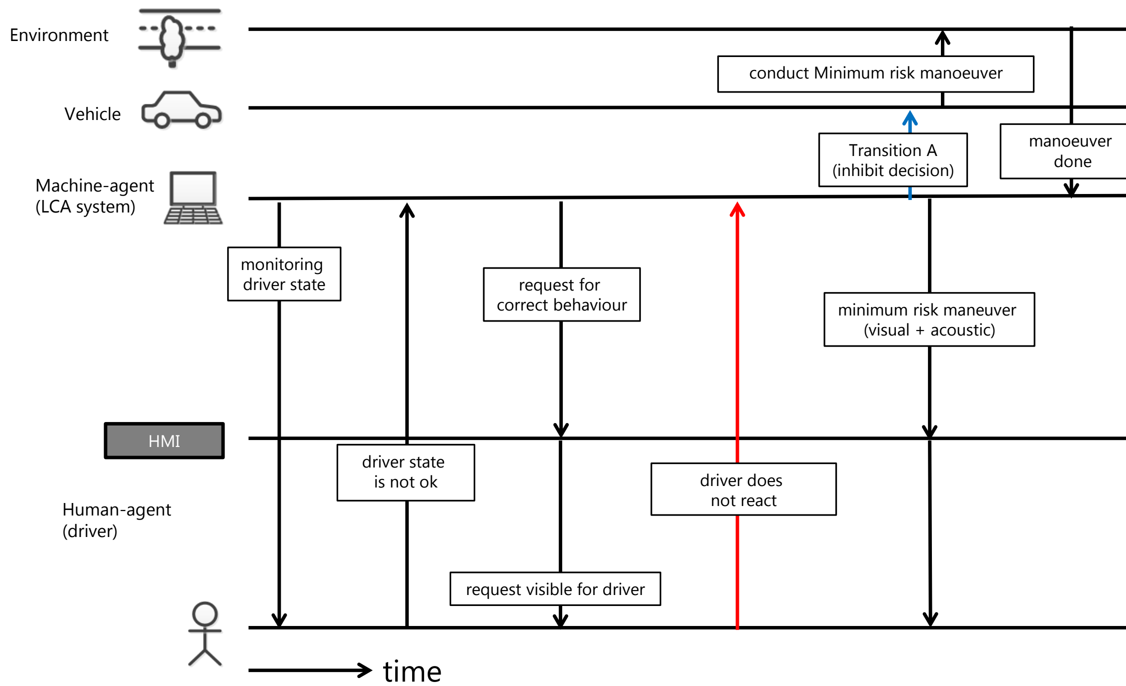


Figure 6: sequence diagram of LCA AdCoS for the use-case "Driver Status – Distracted and not responding".

Also in this case the machine-agent is monitoring the driver, who is distracted; therefore, when he/she intends to perform a lane-change (for overtaking), the system requests for a correct behaviour before supporting this manoeuvre, but the driver is not responding (the driver is too distracted). If the driver is not reacting, a transition can be performed, from the human control of the vehicle to the automated control of it. In particular, a minimum risk manoeuvre is performed by the machine-agent, until the driver remains in the impaired state. The definition of such a manoeuvre will be done during the second project year; for the moment, we propose the inhibition of the lane-change (and thus of overtaking), while simultaneously avoiding the risk of frontal collision with the slower vehicles ahead.

With reference to the Levels of Automation (LoA) from SAIE, the situation can be summarized by the following scheme:

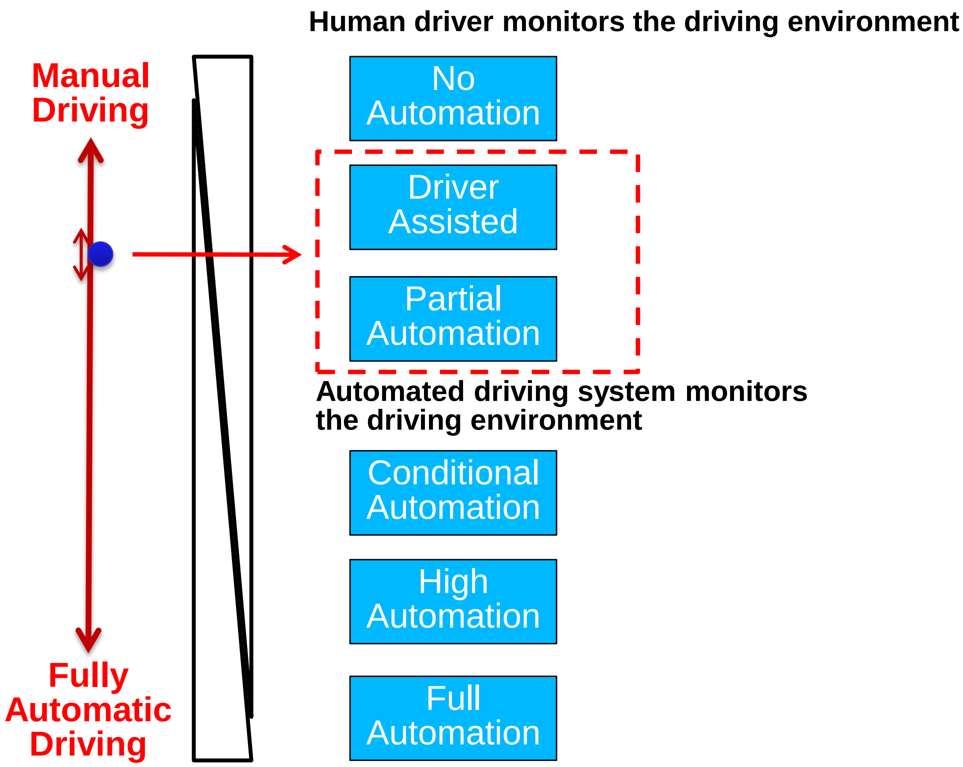


Figure 7: LOA based on SAIE categorization with reference to LCA AdCoS.

The LCA AdCoS is collocated between the levels of "Driver Assisted" and of "Partial Automation): the classification tool of distraction – described above, see the table in Section 2 – is the "trigger" for the adaptation. In fact, depending on the cognitive state of the driver (if he/she is distracted or not), the strategies of the AdCoS are modified, both for LCA and for FCW functions, based on the cases described the sequence diagrams above (figures 3-6).

In addition to these four main use cases investigated by the real car of CRF, similar four critical uses cases will be investigated through simulation with the IFS V-HCD platform. The critical event will be introduced here by an emergency braking of the truck B followed by the ego car A (cf. Figure 1), requiring an emergency reaction of the Driver to avoid collision, in terms of braking or lane change. These critical Use Cases will be investigated for both "Attentive" and "Impaired" (i.e. visually distracted) drivers. In these critical driving conditions, the machine agent (AdCoS) will have to make complex decisions to determine when and how braking and/or lane change manoeuvres are "relevant", "possible" (i.e. with an acceptable level of risk) or "dangerous", with the aim of assisting the driver in an adapted way (from warnings to taking control of the vehicle via Adapted Automation).

3.1.2 Adapted Automation Use-case

Today the development of highly automated driving is the research focus of many OEMs [RAAK13] and research institutes. A major need regarding automated vehicles is an increased usability. 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 [Bm13].

In the past decade it was proven that artificial intelligence can guide a vehicle safely through complex traffic situations. The research and development effort of car manufacturers today focuses on going from a technological showcase to an ergonomic driver assistance system. Main challenges in the development include functional safety, legal aspects and driver acceptance, among others.

Therefore, the AdCoS developed by IAS focuses on an improved interaction between the human and the machine agent. The following aspects are investigated:

1. Adaptation:
2. Human Machine Interaction:
3. System Transparency:

The use case that is investigated refers to the previously described target scenario.

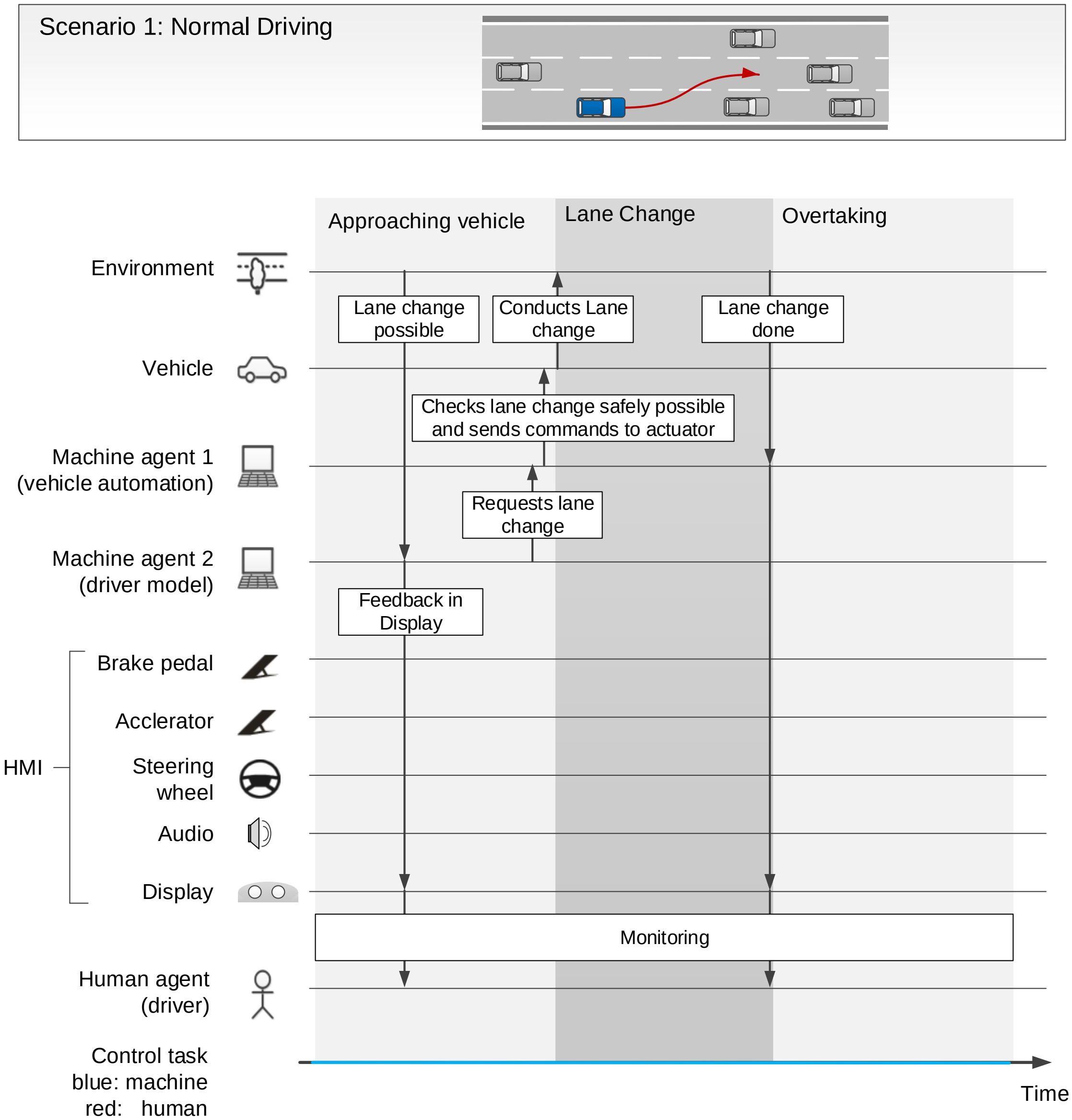
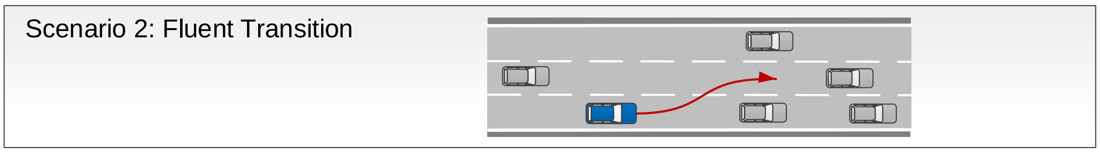


Figure 8: UC1 - Normal Driving

UC1 – Normal Driving [Figure 8**]:** For this scenario we assume the initial learning of the driver model is done and the adapted automated driving mode is active. Moreover, we assume the ego vehicle approaches a slower vehicle in the front, the vehicle in the rear is far enough and a lane change is possible. The possibility of a lane change is given as a feedback to the driver. The driver model (machine agent 2) then decides based on learned natural driving behaviour of the driver when a lane change is suitable for the driver.

The driver model sends its request to the vehicle automation (machine agent 1) and the vehicle automation verifies if a lane change is currently safely possible for the automation. If so, the vehicle automation sends the control commands to the actuators and the vehicle finally conducts the lane change.



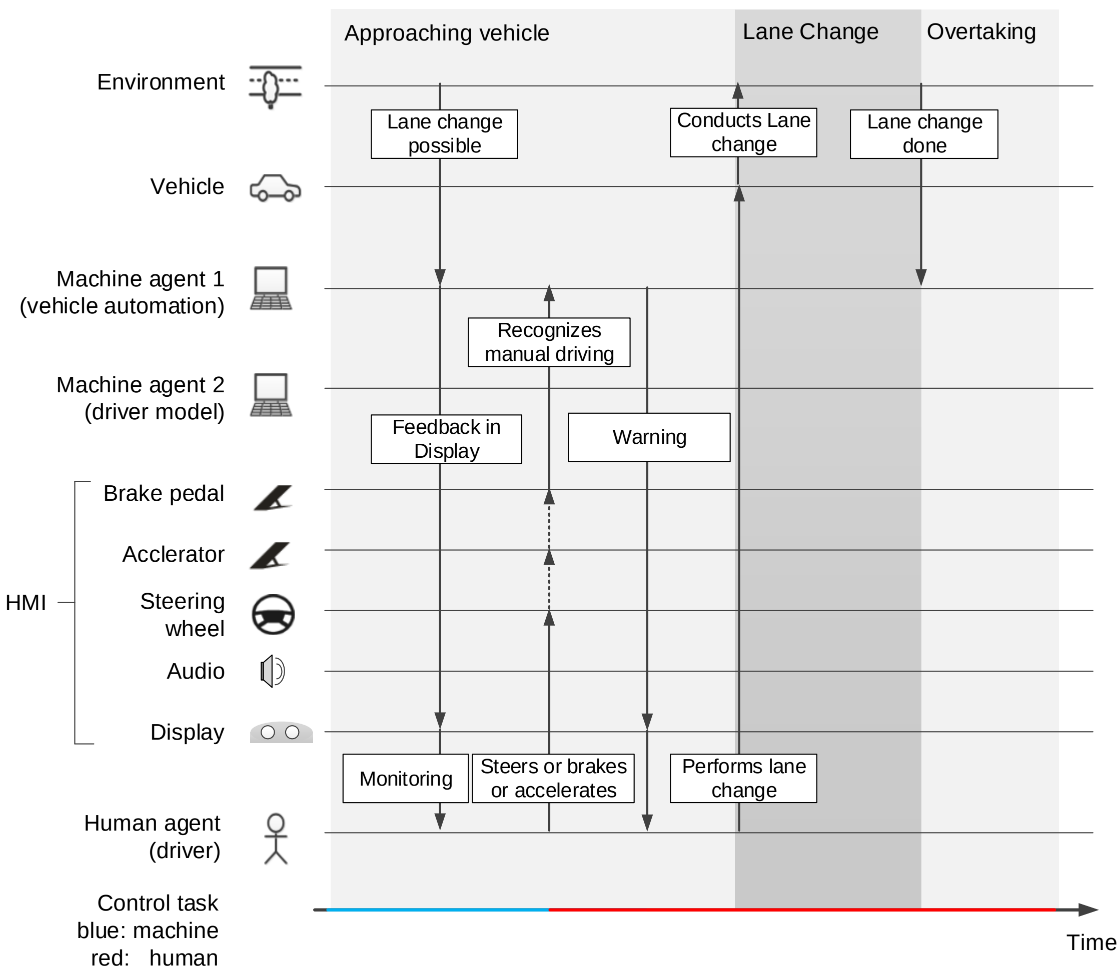
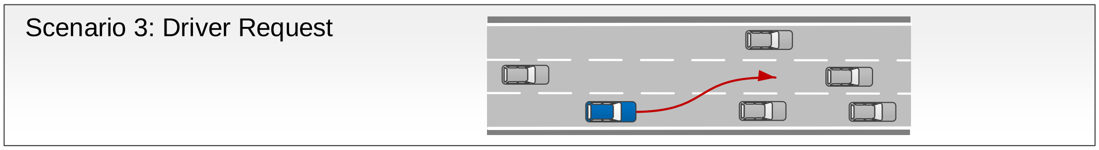


Figure 9: UC2 - Fluent Transition. Control task changed from machine to human.

UC2 – Fluent Transition [Figure 9**]:** Again we assume the initial learning of the driver model is done and the adapted automated driving mode is active. Moreover, we assume the ego vehicle approaches a slower vehicle in the front, the vehicle in the rear is far enough away and a lane change is possible. The possibility of a lane change is given as a feedback to the driver. Compared to scenario 1, the driver does not wait before the driver model initiates the lane change; instead, the driver is aware that a lane is possible and initiates a transition by either steering or braking/accelerating. The vehicle automation (machine agent 1) double checks this transition request and if the lane change is still safely possible the vehicle automation warns the driver, that the control task is now with the human driver.

UC3 – Driver Request [Figure 10**]:** The scenario is the same as in UC2, but when the human driver wants to perform the lane change, he just controls the steering or acceleration pedal for a short time range. This is recognized by the machine agent as a request for a lane change which will be acknowledged and performed autonomously. This can be regarded as a kind of system override.

UC4 – Refused Driver Request [Figure 11**]:** Again we assume the initial learning of the driver model is done and the adapted automated driving mode is active. Moreover we assume the ego vehicle approaches a slower vehicle in front. Compared to the previous scenarios a lane change is not possible since another vehicle is to the left, beside the ego vehicle. This information is given as a feedback to the driver. Nevertheless the driver ignores this information because the driver is for instance inattentive and initiates a transition by either steering or braking/accelerating. The vehicle automation (machine agent 1) double checks the transition request and verifies that a lane change is currently not possible. A warning is sent to the driver.



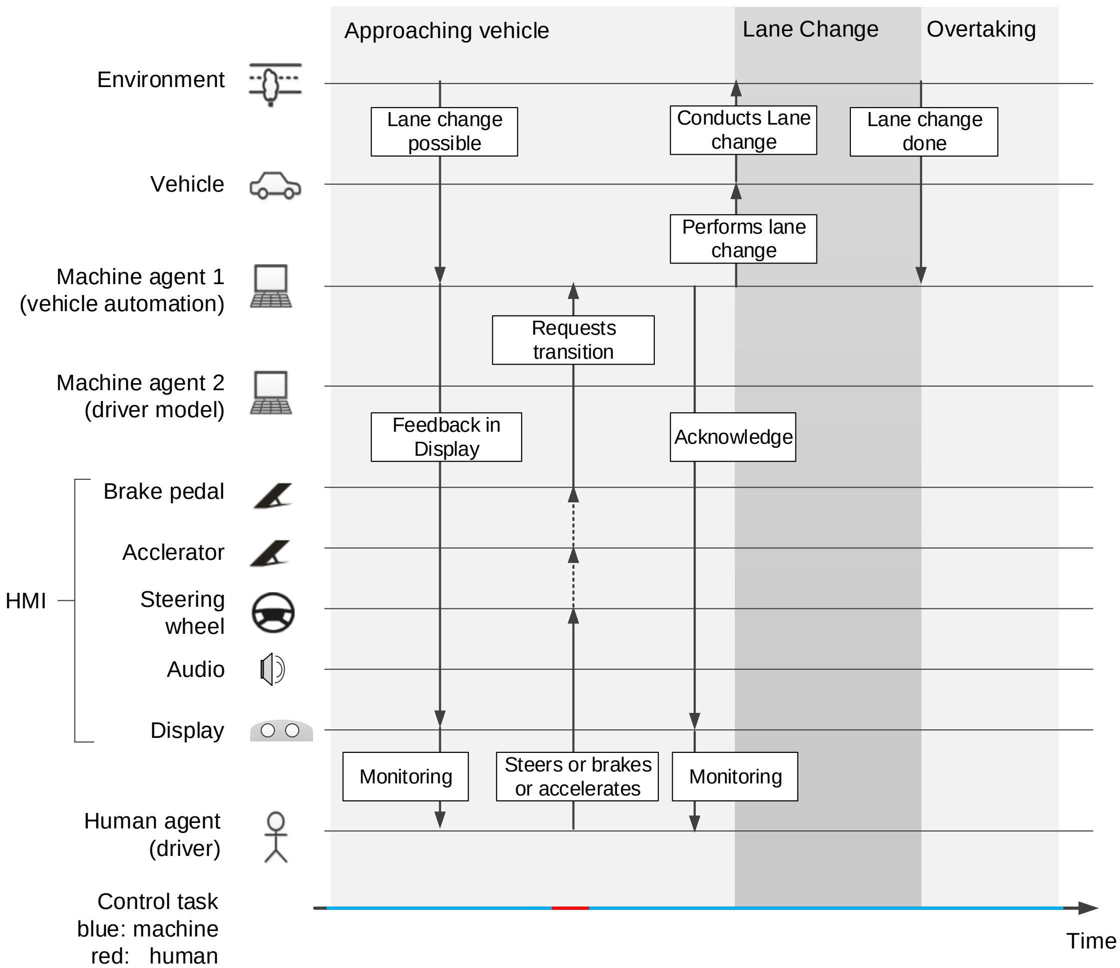


Figure 10: UC3 – Fluent Transition: Short interaction is interpreted as a request.

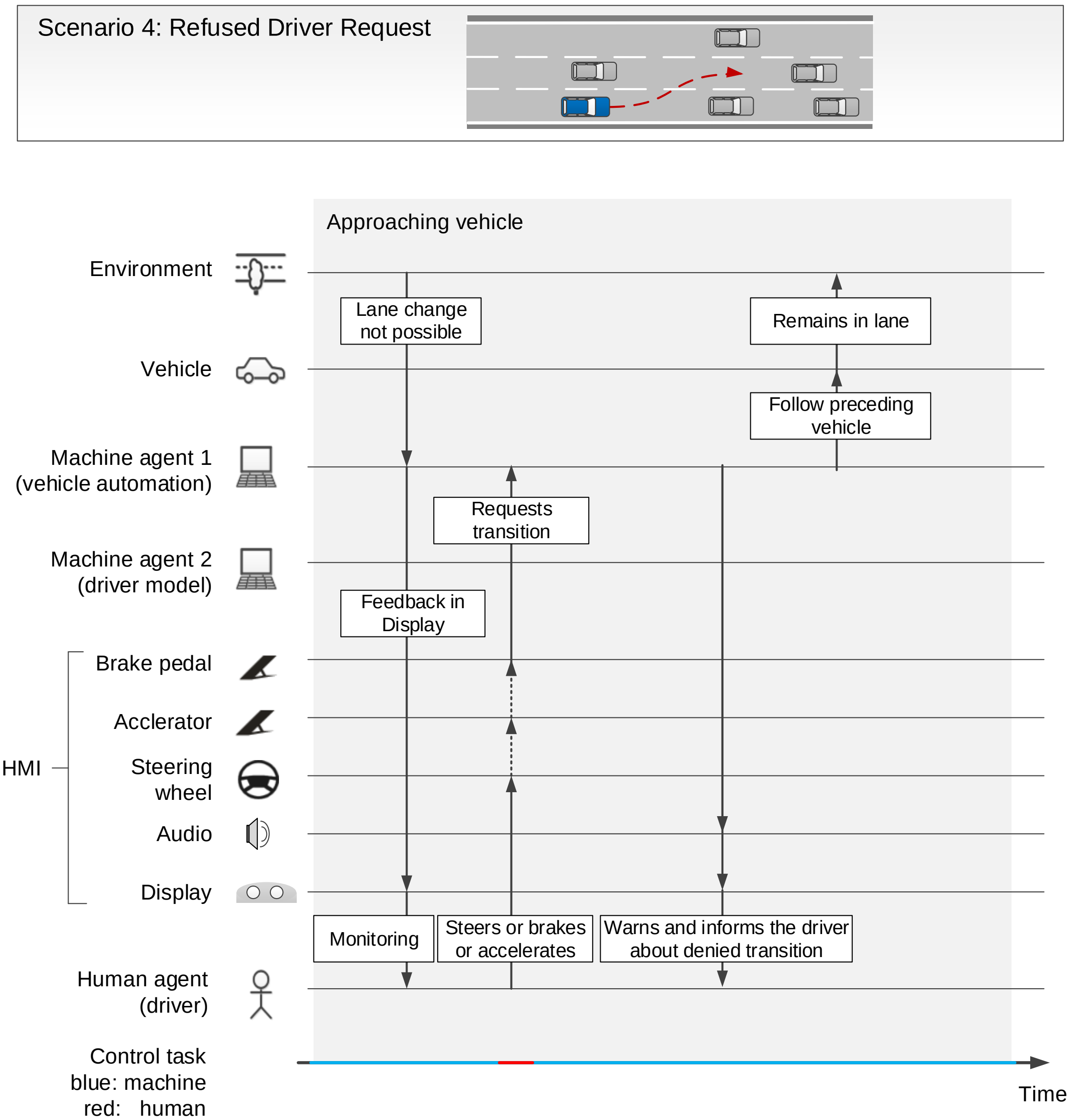


Figure 11: UC4 - Refused Transition.

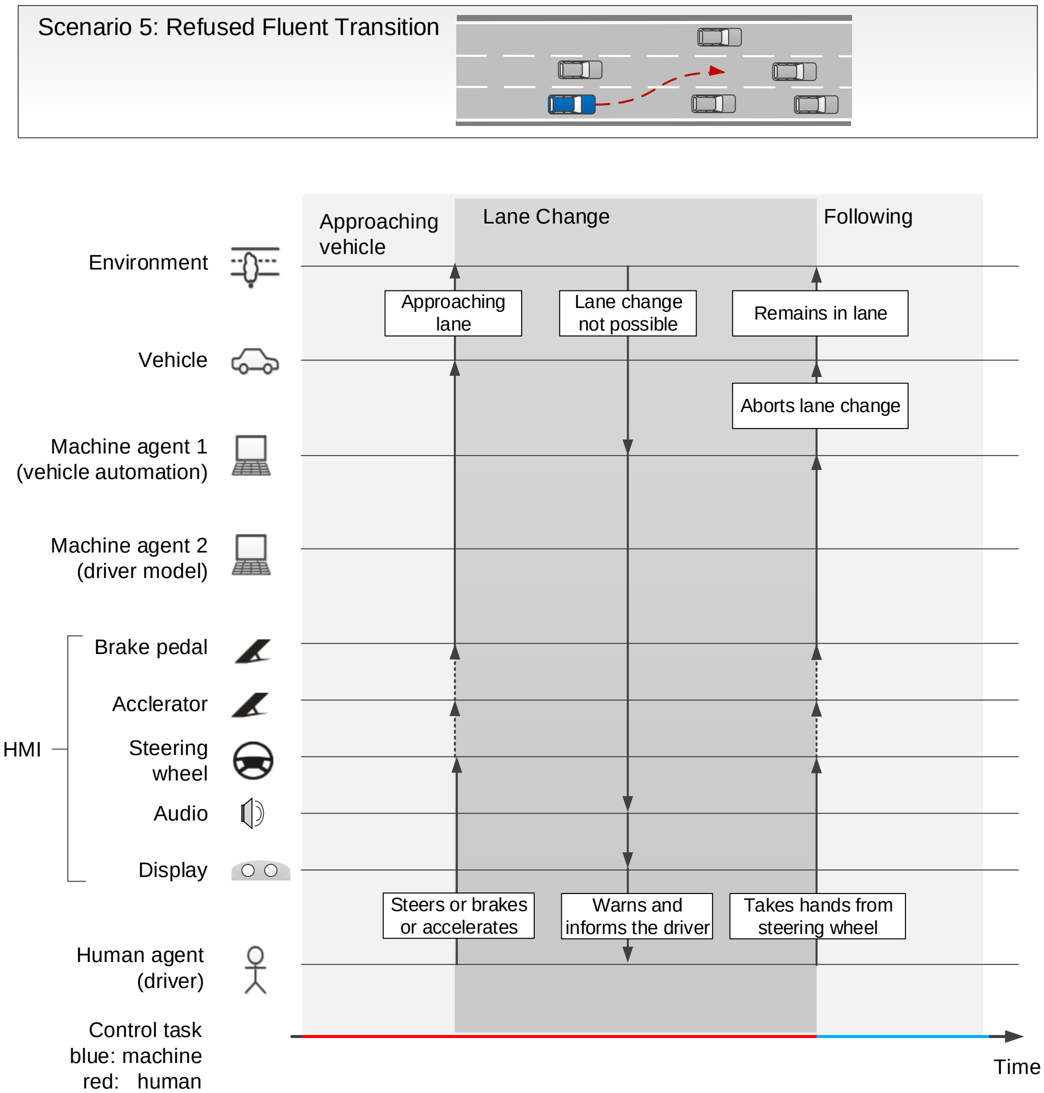


Figure 12: UC5 - Refused Transition. Control task changed from human to machine agent.

UC5 – Fluent Transition [Figure 12**]:** The scenario is as in UC4, but with the difference that the vehicle is driven by the human agent. Although the neighbouring lane is blocked, the driver is trying to perform a lane change, which is recognized by the machine agent and a warning is sent to the driver. The driver does not want to resolve the situation and takes its hands from the steering wheel to pass the driving task to the machine agent, which will solve the situation and keep the lane.

3.1.3 Adapted HMI Use-case

In addition, there is also the Adaptive HMI AdCoS, provided by the TAKATA partner, which is a system that helps the driver to re-direct its attention to the road. There are two preconditions: firstly, the driver has to be distracted, i.e. is not paying attention to the road and secondly, attention is required on the road. The relevance of this scenario is based on the lower levels of automation that still require the driver to take over control in a predefined amount of time. For experimental purposes distraction has to be implemented. This implementation will be done via the surrogate reference task (SURT) which will be presented to the driver on a centre stack display. In the case in which the driver is distracted - that is he or she pays too much attention to the SURT - and in the case in which attention is needed, attention will be re-directed to the road via the HMI. However, besides the adaptive HMI there will also be an adaptive ADAS. This ADAS permits that the limits regarding distance, speed, lane-keeping or other criteria are adapted to the driver state. The variables used here will likely be a combination of different time based variables such as time-to-collision or time-to-line-crossing (TLC).
The TAKATA AdCoS will consist of three parts that will be implemented in the TAKATA driving simulator (see below):

- The distraction detection hard-and software
- The adaptive assistance systems
- The adaptive HMI.

For this AdCoS the use-cases described in Deliverable D9.1 will be used:

- **UC1: Automatic mode** – the system informs the driver, waits until P2 passed, then overtakes.
- **UC2: Automatic mode** – driver wants to override the system and ignores warnings; the system prevents driver from taking over control.
- **UC4: Manual mode** –disattentive driver; the system warns the driver of a potential collision with P3.
- **UC5: Automatic mode** – the same as UC1 but the system accelerates P1 and overtakes before P2 reaches P1.
- **UC6: Manual mode** – the driver wants to overtake despite P2; blind spot warning, the driver steers back; does not slow down; the system warns of P3.

Depending on the technical feasibility, these use-cases might have to be adapted.

3.2 Modelling of the AdCoS

In this section, the modelling approach for the different AdCoS applications is presented and the preliminary results are pointed out.

3.2.1 Modelling the Adapted Assistance System

Preliminary task modelling and task analysis have been carried out by REL on the Lane Change (LC) Manoeuvre, for the design of the HMI of the mobile app that will implement the LC Assistant.

For the graphical representation of tasks, Microsoft Power Point has been used, i.e. the standard tool used by REL for the task analysis representation. This tool has been selected by REL and it has been included in its HMI development process because it presents several advantages:

- It is cross-domain
- It is flexible (it allows including text, notes, images, schemas, etc..)
- It can be shared with customers (as well as project partners) in order to speed up the definition of the task model

However, it has also relevant drawbacks, such as:

- It does not support the designer in the definition of the task model
- It does not provide any graphical support to create and modify the tasks and subtasks
- The representation of a non-trivial tasks (such as the LC Manoeuvre) requires several subtasks and sublevels, which can be represented in a single PowerPoint slide (or even a set of slides) only with difficulty.

Therefore, REL decided to apply the MagicPED tool (developed by OFF in WP2) to assess the potential improvements it can bring. This preliminary modelling activity thus represents the baseline for the future cycles, in order to understand the improvements that could be achieved by using the MagicPED instead of Microsoft Power Point.

The task analysis of lane change manoeuvre provides a general description of the tasks involved with making a Lane Change (LC) Manoeuvre. These tasks are preliminary cognitive, motor, visual or some combination thereof.

The preliminary task modelling and task analysis carried out by REL about the LC Manoeuvre was meant as a preparatory activity for the definition of the HMI of the LC Assistant.

In fact, the task analysis can highlight the cognitive, visual and motor loads of the driver in each task and subtask of the overall LC Manoeuvre, and this information is key for the design of an HMI that can actually adapt to the context (driving status, driving intention, etc.), and provide tailor-made information which the driver is capable of processing in continuously changing conditions.

This task overview has implications for HMI design and prioritization of information according to the specific task the driver is performing, in order to avoid presenting safety critical information or pressing for decision-making tasks where cognitive load is high or, preventively, when lane change is likely to occur (according to the forecast provided by the driver intention module).

The overall LC Manoeuvre task has been split into 3 sub-tasks, as shown in Figure 13.

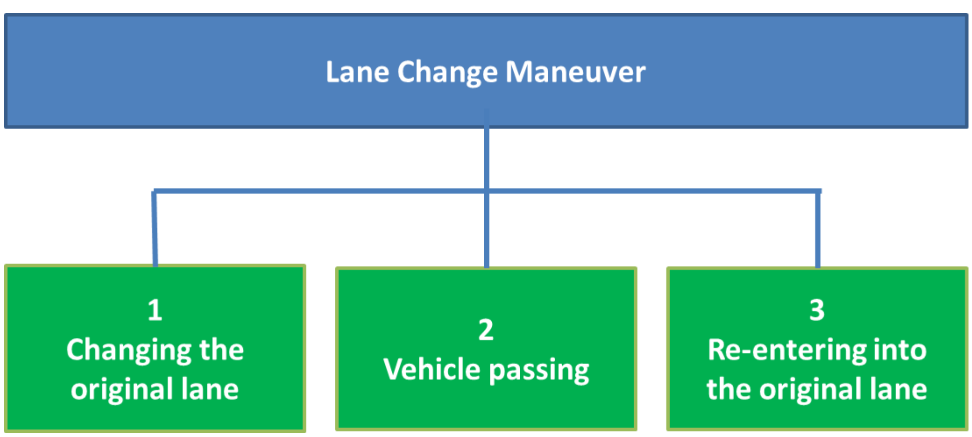


Figure 13: Preliminary macro task analysis of Lane Change Manoeuvre

By adapting the guidelines provided by Lee, Olsen and Wierwill for interchange design and sign placement [3], each sub-subtask can be categorized as:

1) Decision: decide when the manoeuvre is possible (sub-tasks in yellow)
2) Preparation: prepare for the manoeuvre (sub-tasks in orange)
3) Execution: perform the manoeuvre (sub-tasks in blue)

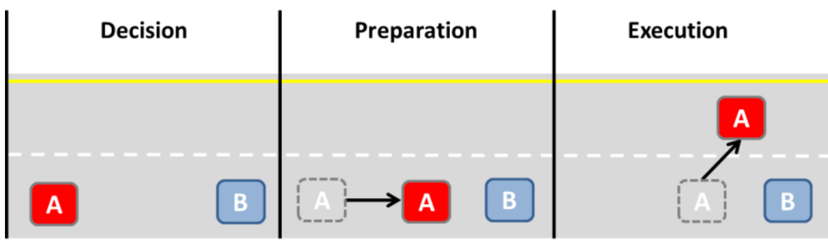


Figure 14: Representation of decision, preparation and execution tasks to change the original lane (task1)

Figure 14 represents the different phases (decision, preparation and execution) while changing the lane.

Figure 15 shows the task model for the first subtask (changing the original lane), where the different colours highlight how it includes decision, preparation and execution tasks.

3 Lee S. E., Olsen E. C. B., Wierwille W. W.(2004). *A Comprehensive Examination of Naturalistic Lane-Changes. Report No. DOT HS 809 702*. Washington, D.C.: National Highway Traffic Safety Administration.

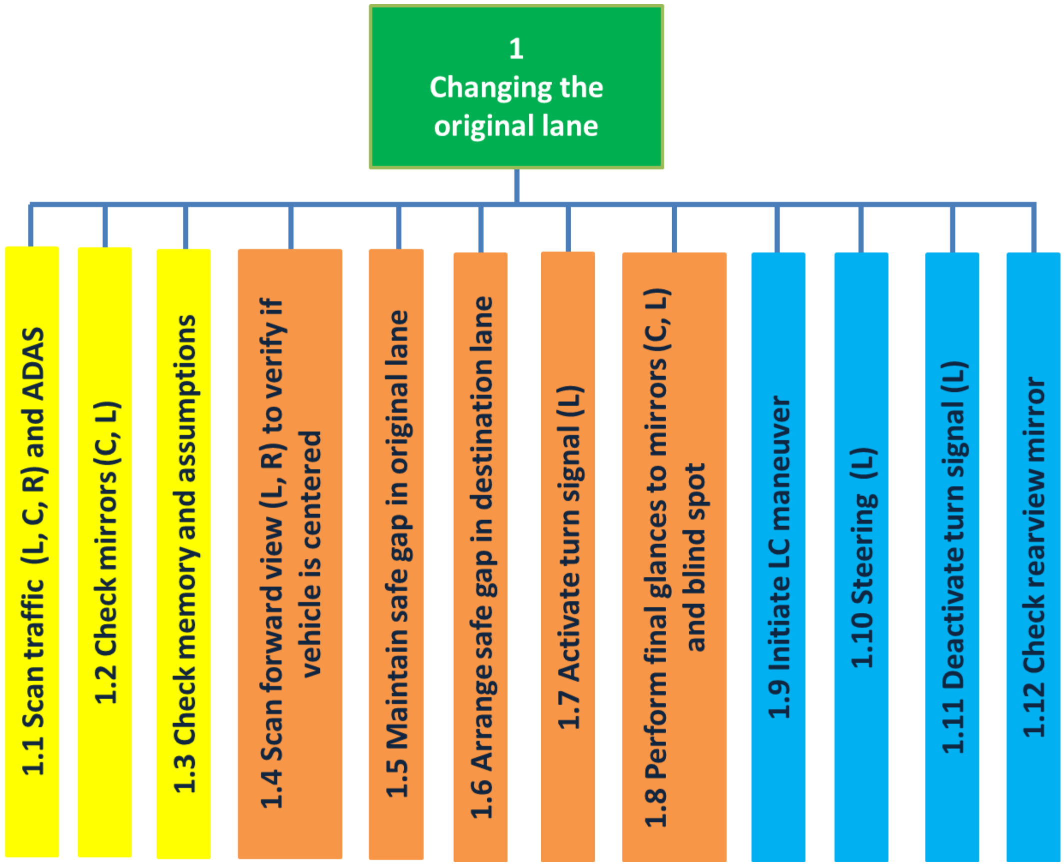


Figure 15: Sub-tasks involved in changing the original lane (task1)

The approach has been applied to the other sub-tasks ("vehicle passing" and "re-entering into the original line").

Figure 16 represents the different phases (decision, preparation and execution) while passing a vehicle (second sub-task), and Figure 17 the corresponding task model.

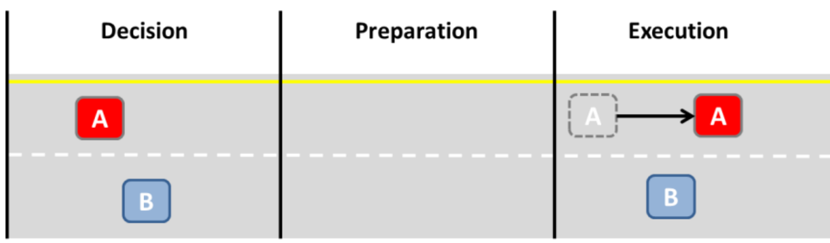


Figure 16: Representation of decision, preparation and execution tasks to pass a vehicle (task2)

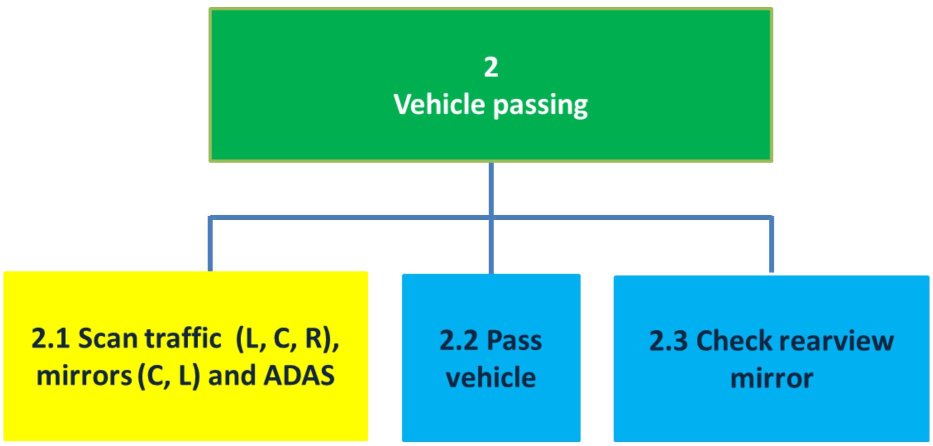


Figure 17: Sub-tasks involved in passing a vehicle (task2)

Figure 18 represents the different phases (decision, preparation and execution) while re-entering the original lane (third sub-task), and Figure 19 the corresponding task model.

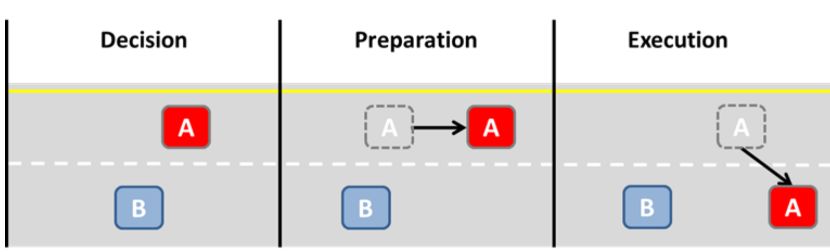


Figure 18: Representation of decision, preparation and execution tasks to re-enter the original lane (task3)

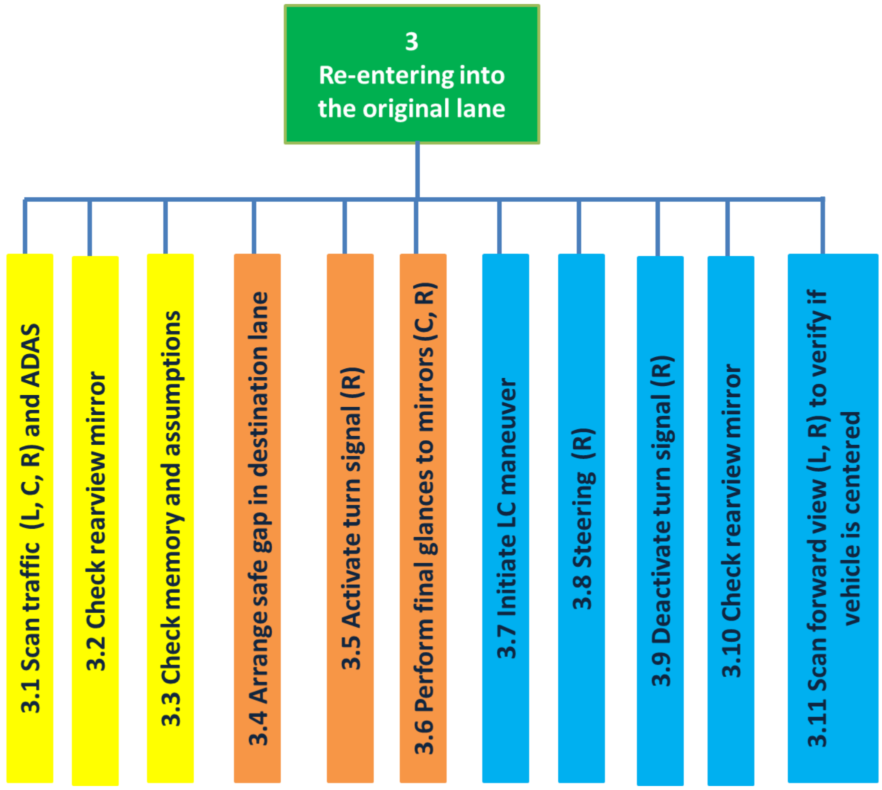


Figure 19: Sub-tasks involved in re-entering the original lane (task3)

According to the cognitive, motor and visual tasks that the driver must complete in each phase, he/she has different cognitive, motor and visual loads, summarized in Table 2.

Subtask		Decision	Preparation	Execution
1. Changing the original lane	**Cognitive load**	medium	medium	medium
	Visual load	high	high	medium
2. Vehicle passing	**Cognitive load**	low	low	low
	Visual load	medium	low	medium
3. Re-entering the original lane	**Cognitive load**	medium	medium	medium
	Visual load	high	high	medium

Table 2: cognitive, motor and visual loads in each subtask.

Table 2 provides a relevant support for the design of the HMI of the AdCoS. In fact, the AdCoS (LC assistant) can adapt to the status of the driver (distraction, intention, etc..) and the status of the environment (other cars approaching) and provide different information to the driver, by also exploiting different interaction modalities.

3.2.2 Probabilistic Driver Intention Recognition

Intention recognition is primarily concerned with the recognition of behaviour intentions, which are defined as "*a person's intentions to perform various behaviours*" [FA75, p. 12]. In the automotive domain, i.e., the case of driving intentions, behavioural intentions mainly refer to the intentions of a human driver to follow certain behavioural schemata or to perform certain driving manoeuvres like e.g., overtaking or lane changes (e.g., [LP97], [OP00]).

Under the assumption that a person has a sufficient degree of actual control over the intended behaviour (i.e., the corresponding task is not executed by another agent), the existence of an intention implies the readiness for execution and "*people are expected to carry out their intentions when the opportunity arises*" [Ajz02, p. 12]. Following these implications, an intention can be seen as an immediate antecedent or predictor of the human's behaviour in the near future [Ajz02]. Knowledge about the current driving intentions of the human driver would therefore allow an AdCoS to adapt in order to comply these intentions and therefore decrease the risk of decreased user-acceptance or to initiate appropriate countermeasures when these intentions do not comply with the assessed situation.

Within HoliDes, intention recognition will be performed by a Driver Intention Recognition (DIR) module. As the name suggests, the DIR module solely focusses on the automotive use-cases addressed in WP9. However, many of the techniques used are domain-independent and are applicable for other domains and use-cases. The main objective of the DIR module is the assessment of the current intentions and behaviours of a single human agent, i.e., the driver of vehicle A depicted in **Fehler! Verweisquelle konnte nicht gefunden werden.**, within the use-cases for adapted assistance (Section **Fehler! Verweisquelle konnte nicht gefunden werden.**).

3.2.2.1 Selected behaviours and intentions

In order to select which intentions the DIR module shall be able to recognize, the selection of a set of behaviours/manoeuvres we expect the driver to perform is required. Based on the use-cases for adapted assistance, we selected the following set of behaviours that would allow the human driver to travel on a highway in the absence of emergencies:

- To perform a lane change to the left lane
- To perform a lane change to the right lane
- To perform lane-following
- To perform car-following

Not all of these behavioural schemata or manoeuvres are necessarily triggered intentionally, e.g., under the assumption of normative driving, a transition from lane-following to car-following should occur naturally given the current situation if no countermeasure (like e.g., performing a lane-change in order to overtake) is initiated by the driver. Furthermore, given the highly dynamic environment in the automotive use-cases and the limited knowledge about the environment due to limited sensor capabilities (e.g., surrounding traffic participants may be outside of the detection range, or a leading vehicle may occlude a second leading vehicle) we limit intention recognition to "short-term" intentions, which in the context of the selected use-cases can be narrowed down to lane change intentions. By now, we therefore selected two distinct behaviour intentions and an additional absence of intention:

- The intention to change to the left lane
- The intention to change to the right lane
- The absence of the above intentions

Additional or more fine-grained behaviours (following the task analysis in Section **Fehler! Verweisquelle konnte nicht gefunden werden.**) and intentions may be added after an evaluation of experimental data expected early in 2015.

3.2.2.2 Modelling Approach

Intentions are theoretical constructs that cannot be measured or assessed directly [Kob11]. This is especially true in the case of driving, where the choice and execution of manoeuvres may be highly automated skills whose execution will not necessarily be considered by the driver as intentional

[Ajz02]. Accordingly, they have to be inferred from the available context. Many modelling techniques have been proposed for intention recognition (an overview of the state of the art can be found e.g., in [Kob11] and [Bör13]), including rule- and fuzzy-based, and several discriminative (e.g., logistic regression) and generative probabilistic models (e.g., (dynamic) Bayesian networks). Due to the variability of human cognition and behaviour, the irreducible lack of knowledge about underlying cognitive mechanisms, and the irreducible incompleteness of knowledge about the environment, we will focus on the use of probabilistic graphical models, esp. dynamic Bayesian networks (DBNs).

As such, we will be concerned with probability distributions over sets of discrete and continuous random variables. Variables will be denoted by capital letters, such as X, Y, Z, while specific values taken by those variables will be denoted by corresponding lowercase letters x, y, z. We use boldface type capital letters $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$ to denote sets of random variables, e.g. $\boldsymbol{X} = \{X_1, \dots, X_n\}$ and corresponding boldface lowercase letters $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$ to denote assignments of values to the variables in these sets. For time series, we assume that the timeline is discretized into time slices with a time granularity corresponding to the control loop of the AdCoS for adapted assistance (c.f., Section **Fehler! Verweisquelle konnte nicht gefunden werden.**), i.e., 100ms. We will index these time slices by non-negative integers and will use X_i^t to represent the instantiation of a variable X_i at time t. A sequence $X_i^j, X_i^{j+1}, \dots, X_i^k$ will be denoted by $X_i^{j:k}$ and we will use the notation $x_i^{j:k}$ for an assignment of values to these sequences. Probability distributions will be denoted by $P(\cdot)$, probability density functions (PDFs) by $p(\cdot)$. As the probability $P(X = x)$ of a single value x for a continuous variable X with a PDF $p(X)$ is always zero, we imply without further notion that each assignment $X = x$ of a continuous variable X is replaced by $x - \epsilon \leq X \leq x + \varepsilon$. When ε is sufficiently small, the probability $P(x - \epsilon \leq X \leq x + \varepsilon)$ can be approximated by

$$P(x - \epsilon \leq X \leq x + \varepsilon) = \int_{x-\epsilon}^{x+\varepsilon} p(x)dx \approx 2\epsilon p(x).$$

Using the same ε for all probabilistic density functions will result in a common prefactor 2ε in all corresponding expressions that will be canceled during the inference process.

A Bayesian Network (BN) is an annotated directed acyclic graph (DAG) that encodes a joint probability over a set of random variables $\boldsymbol{X} = \{X_1, \ldots, X_n\}$. 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 $\boldsymbol{Pa}(X_i)$ 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 $\boldsymbol{X}$ as:

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

DBNs extend BNs to model the stochastic evolution over a set of variables $\boldsymbol{X} = \{X_1, \ldots, X_n\}$ over time. Note that each variable X_i does not represent a random variable of the joint distribution that takes a value, but instead a template variable that is instantiated at different points in time t, and each X_i^t is a variable that takes a value in $Val(X_i)$. 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 $P(\boldsymbol{X}^1)$ 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) $P(\boldsymbol{X}^t | \boldsymbol{X}^{t-1})$ 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 $P\left(X_i^t | \boldsymbol{Pa}(X_i^t)\right)$, where a parent $X_j^t \in \boldsymbol{Pa}(X_i^t)$ can either be in time-slice t or $t-1$. The JPD over an arbitrary number of T time-slices is then given by:

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

Modelling (dynamic) Bayesian networks requires the selection and definition of the random variables included in the model, determining the graph-structure of the model as a factorization of the JPD of these variables, and specifying the parameters that define the probabilities of the (conditional) probability distributions induced by the selected structure. In general, these steps will be guided in respect to a set of experimental data, which in the case of HoliDes, will be given by multivariate time-series of human behaviour traces.

A fully specified (dynamic) BN can be used for performing inferences, i.e. answering probability queries about posterior probabilities of variables in the model. A probability query consists of two parts: A subset $\boldsymbol{E} \subseteq \boldsymbol{X}$ of random variables in the model, and an instantiation $\boldsymbol{e}$ to these variables, called the *evidence*, and a subset $\boldsymbol{Y} \subseteq \boldsymbol{X}$ of variables in the model called the query variables, with $\boldsymbol{E} \cap \boldsymbol{Y} = \emptyset$. Inference denotes the computation of the posterior probability distribution over the values $\boldsymbol{y}$ of $\boldsymbol{Y}$, conditioned on the fact that $\boldsymbol{E} = \boldsymbol{e}$: $P(\boldsymbol{Y} | \boldsymbol{E} = \boldsymbol{e})$ [KF09].

Specifically, in the project the Bayesian Autonomous Driver Mixture-of-Behaviour (BAD MoB) models have been used, which are probabilistic models of the human driver. They are primarily developed in WP2 and will be described in more detail in D2.4. In general, a BAD MoB model is a DBN that aims to model the complex sensorimotor system of a human driver by combining multiple DBNs in a modular and hierarchical probabilistic architecture. Prior to HoliDes, BAD MoB models were developed and used as probabilistic driver models for autonomous control in simulator environments. For HoliDes, we are working on extending BAD MoB models to make them a valuable tool for intention recognition.

Within the context of HoliDes, a BAD MoB model defines a JPD over sets of discrete and continuous random variables representing *intentions*, *behaviours*, (*human*) *actions*, and (*context*) *observations*. Based on exemplary datasets provided by CRF, the use-cases for adapted assistance and the system specification for the LCA AdCoS, we consider the following random variables:

- Intentions: The selected behavioural intentions of the driver are represented by a discrete random variable I, with the possible values $\mathrm{Val}(I) = \{\text{lane change left, lane change right, default}\}$.
- Behaviours: The addressed behaviours/manoeuvres are represented by a discrete random variable B, with the possible values $\mathrm{Val}(B) = \{\text{lane change left, lane change right, lane-following, car-following}\}$.
- Actions: The actions of the driver are represented by a set of discrete and continuous random variables $\boldsymbol{A} = \{A_1, \dots, A_n\}$. By now, we consider the following variables as actions:

Variable	Type	Description
Steering angle	Continuous	Steering angle value
Acceleration	Continuous	Acceleration of the driver's vehicle
Head position	Continuous	Position of the driver's head
Head position rate of change	Continuous	Rate of change of the driver's head position
Head orientation	Continuous	Orientation of the driver's head (yaw, roll, pitch)
Head orientation rate of change	Continuous	Rate of change of the driver's head orientation
Direction Indicator signal	Discrete	Status of the left and right indicators

- Observations: All available internal and external context information that does not correspond to the actions of a driver is represented by a set of discrete and continuous random variables denoted by $\boldsymbol{O} = \{O_1, \ldots, O_m\}$. It is expected that most input available for intention recognition is already pre-processed, i.e., not provided as raw sensor data but instead as filtered (point) estimates based on an internal world model inherited by the sensor itself (e.g., by the use of Kalman-Filters). As a consequence, the model does not utilize a hidden world-model that needs to be estimated from noisy sensor data, and can instead take the provided estimates as evidence. While many observation variables correspond to available sensor data (c.f., Section **Fehler! Verweisquelle konnte nicht gefunden werden.**), additional variables are defined as functions of these values, e.g., rates of changes, time headway, or time-to-contact values. By now, we focus on the following variables:

Input	Type	Description
Lane curvature	Continuous	Curvature of the road
Lateral derivation	Continuous	Lateral distance between the middle of the lane and the longitudinal axis of the driver's vehicle
Yaw angle	Continuous	Angle between the longitudinal axis of the driver's vehicle and lane direction, tangent to the lane (also called "lane yaw angle")
Yaw angle rate of change	Continuous	Rate of change of the yaw angle
Lead car lat. speed	Continuous	Lateral velocity of the lead vehicle
Lead car lat. acceleration	Continuous	Lateral acceleration of the lead vehicle
Lead car long. speed	Continuous	Longitudinal velocity of the lead vehicle
Lead car long. acceleration	Continuous	Longitudinal acceleration of the lead vehicle
Lead car lat. distance	Continuous	Lateral distance of the lead vehicle with respect

		to the driver's vehicle
Lead car long. distance	Continuous	Longitudinal distance of the lead vehicle with respect to the driver's vehicle
THW to lead car	Continuous	Time headway to the lead vehicle
THW to lead car change of rate	Continuous	Rate of change of the time headway to the lead vehicle
TTC to lead car	Continuous	Time-to-contact to the lead vehicle
TTC to lead car rate of change	Continuous	Rate of change of the time-to-contact to the lead vehicle
Velocity difference	Continuous	Difference between the velocities of the driver's and the lead vehicle
Velocity difference rate	Continuous	Rate of change of the difference between the velocities of the driver's and the lead vehicle
Velocity	Continuous	Velocity of the driver's vehicle
VDD	Discrete	Visual Distraction Detection
VTSD	Continuous	Visual Time Sharing Distraction

In general, it is assumed that the actions and observations are always observable (i.e., we will be provided with actual values that can be used as evidence during inference), while the intentions and behaviours are always hidden. The distinction between actions and observations is rather redundant, as both will be provided by dedicated sensors. However, by now, we only model temporary dependencies between actions, not between observations. These restrictions will be relaxed during the course of the project, which consequently may render variables representing rate-of-changes redundant. Additionally, not all available selected variables are necessarily valuable for intention recognition and accordingly not all of them will necessarily be included in later BAD MoB models used for intention recognition.

By now, we rely on simulator data obtained in previous projects and synthetic data to develop learning algorithms and evaluate different factorizations of the resulting JPD $P(I^{1:T}, B^{1:T}, \boldsymbol{A}^{1:T}, \boldsymbol{O}^{1:T})$. We therefore focus on developing general template structures, whose exact structure can be refined in the light of experimental data, expected in early 2015. By now, we mainly consider the use of two template factorizations, whose graph structures are shown in Figure 20. The first one is based on a model class known as Factorial Hidden Markov Models (FHMM) [GJ97], for which, for an arbitrary number of T time-slices, the JPD is defined as:

$$P(I^{1:T}, B^{1:T}, \boldsymbol{A}^{1:T}, \boldsymbol{O}^{1:T}) = P(I^1)P(\boldsymbol{O}_I^1|I^1)P(B^1|I^1)P(\boldsymbol{O}_B^1|B^1, I^1)P(\boldsymbol{A}^1|B^1, I^1)P(\boldsymbol{O}_A^1|\boldsymbol{A}^1, B^1, I^1) \prod_{t=2}^{T} P(I^t|I^{t-1})P(\boldsymbol{O}_I^t|I^t)P(B^t|B^{t-1}, I^t)P(\boldsymbol{O}_B^t|B^t, I^t)P(\boldsymbol{A}^t|\boldsymbol{A}^{t-1}, B^t, I^t)P(\boldsymbol{O}_A^t|\boldsymbol{A}^t, B^t, I^t)$$

The second template is based on a model class known as Hidden Markov Decision Trees (HMDT) [JG96], for which the JPD factorizes as:

$$P(I^{1:T}, B^{1:T}, \boldsymbol{A}^{1:T}, \boldsymbol{O}^{1:T}) = P(I^1|\boldsymbol{O}_I^1)P(I^1)P(B^1|I^1, \boldsymbol{O}_B^1)P(\boldsymbol{O}_B^1)P(\boldsymbol{A}^1|B^1, I^1, \boldsymbol{O}_A^1)P(\boldsymbol{O}_A^1) \prod_{t=2}^{T} P(I^t|I^{t-1}, \boldsymbol{O}_I^t)P(\boldsymbol{O}_I^t)P(B^t|B^{t-1}, I^t, \boldsymbol{O}_B^t)P(\boldsymbol{O}_B^t)P(\boldsymbol{A}^t|\boldsymbol{A}^{t-1}, B^t, I^t, \boldsymbol{O}_A^t)P(\boldsymbol{O}_A^t)$$

Both templates define a general factorization of the JPD that ensures efficient inference, while allowing many different finer factorizations of its CPDs. The final structure of the BAD MoB model and the parameters of the (conditional) probability distributions will be derived by machine-learning methods from multivariate time series of human behaviour traces once they are available (expected early in 2015).

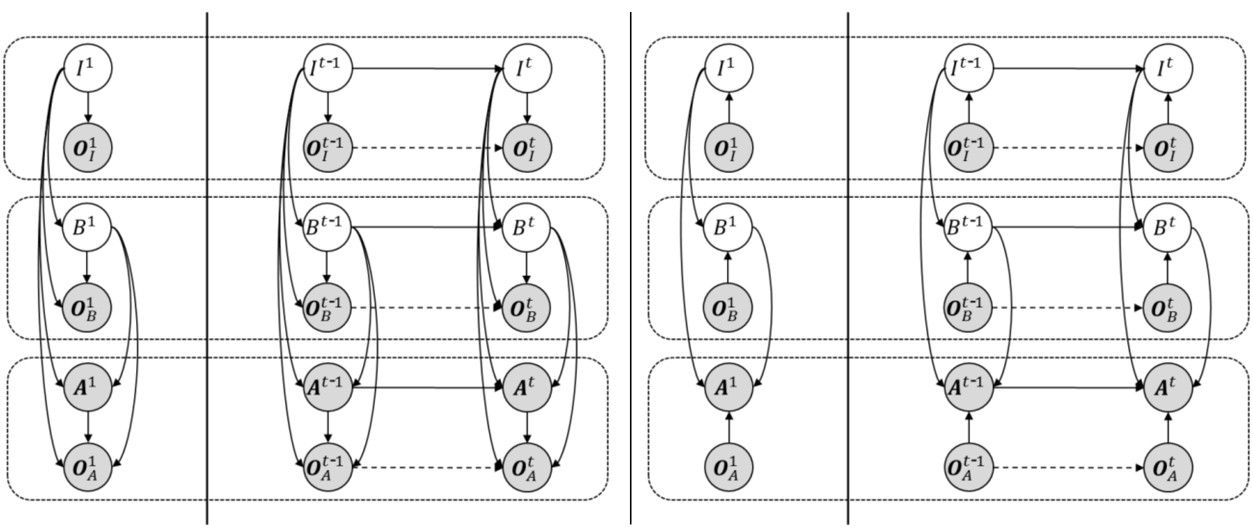


Figure 20: Template structures of BAD MoB models for intention recognition, (loosely) based on FHMMs (left) and HMDTs (right), both defined by a Bayesian network for the first time-slice $t = 1$ and a 2TBN for all $t > 1$. Blank nodes represent hidden variables, shaded nodes represent variables that are assumed to always be observed. Dotted lines imply optional temporal dependencies between

observations. Dotted boxes imply the scope of component-models with private observations.

Given a fully specified BAD MoB model, it will be used constantly, at each time step t, to infer the joint belief state of intentions and behaviours given all available evidence about actions and observations observed so far: $P(I^t, B^t | \boldsymbol{a}^{0:t}, \boldsymbol{o}^{0:t})$. Given this joint belief state, we can easily obtain the marginal belief states of intentions $P(I^t | \boldsymbol{a}^{0:t}, \boldsymbol{o}^{0:t})$ and behaviors $P(B^t | \boldsymbol{a}^{0:t}, \boldsymbol{o}^{0:t})$. The estimation of the belief state is known as filtering and can be solved by in constant time by recursively computing $P(I^t, B^t | \boldsymbol{a}^{0:t}, \boldsymbol{o}^{0:t})$ from the past belief state $P(I^{t-1}, B^{t-1} | \boldsymbol{a}^{0:t-1}, \boldsymbol{o}^{0:t-1})$.

3.2.2.3 Driver Intention Recognition Module

The Driver Intention Recognition (DIR) module aims at predicting the intention of the driver, that is to classify and recognize in advance the vehicle manoeuvre and is located in the application layer of the AdCoS for adapted assistance (c.f., Section **Fehler! Verweisquelle konnte nicht gefunden werden.**). The DIR module is developed and provided as an RTMaps macro-component that is composed of several smaller RTMaps components.

As depicted in Figure 21, the DIR module consists of two parts, a domain- and task-*dependent* part tailored to the actual system architecture and specification for the AdCoS for adapted assistance that deals with pre-processing the available input, and a domain- and task-*independent* part that enables the DIR module to answer probabilistic queries according to a probabilistic model defined in an xml-specification using an inference engine. Both parts are implemented as separate RTMaps components, so that the domain- and task-independent inference engine can potentially be used in different domains utilizing different probabilistic models.

As the development of the DIR module was guided and based on small exemplary datasets provided by CRF that were not sufficient for learning the actual structure and parameters of a probabilistic model, we focussed on the development and implementation of the general functionality of the DIR module regardless of the actual probabilistic model used for intention recognition. Once experimental data is available, we will shift the focus of development to learning the graph-structures and parameters of BAD MoB models.

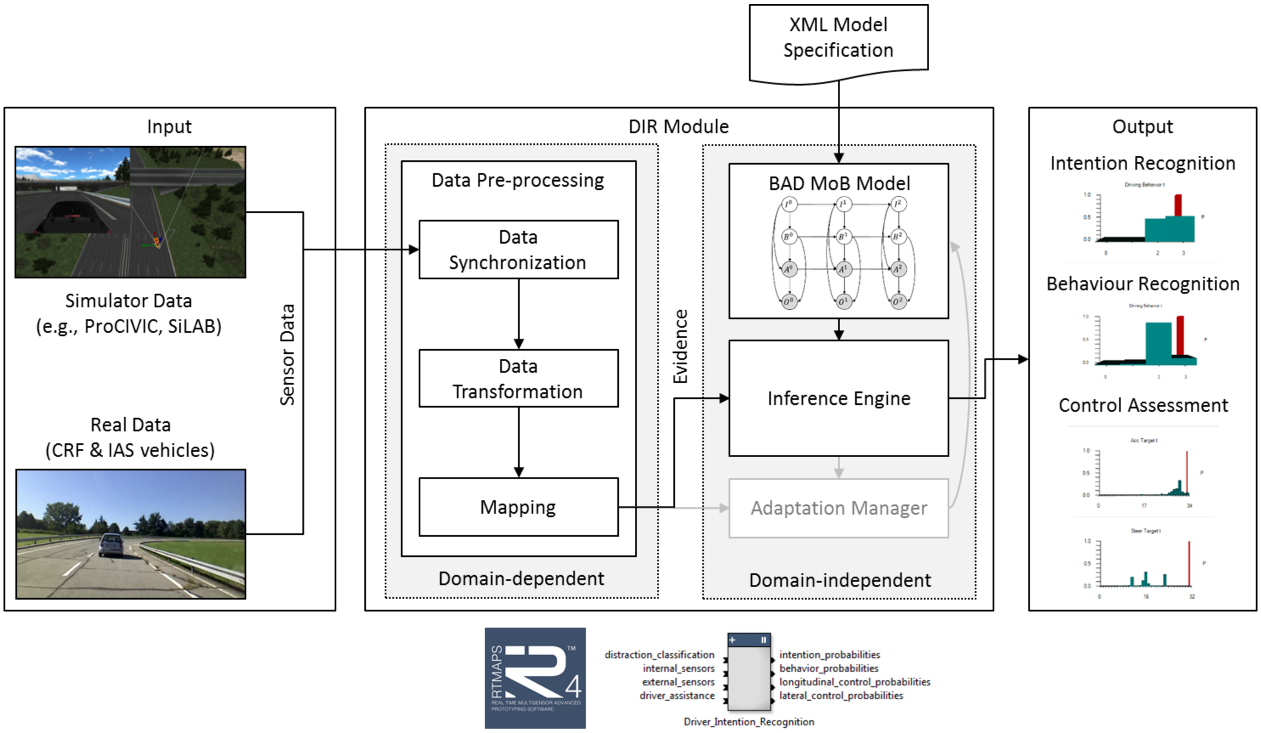


Figure 21: Overview of the DIR module.

3.2.2.3.1 Input

A context assessment function for intention recognition requires input in the form of information about the human agent, the system under control, and the environment. In the case of the DIR module, this input is provided by the Perception Layer of the CRF test-vehicle (c.f., Section **Fehler! Verweisquelle konnte nicht gefunden werden.**) which is composed of external sensors (e.g., cameras and radar) that perceive the environment and internal sensors (e.g., controller area network bus and cameras) that observe the driver and the state of the vehicle. By now, the following input is available and defined for the CRF test-vehicle:

Input	Description
Direction indicator left	Status of the left indicator
Direction indicator right	Status of the left indicator
Lane Lateral Distance[m]	Lateral distance between the middle of the lane and the longitudinal axis of the driver's vehicle
Lane Curvature[1/m]	Curvature of the road
Lane Width [m]	Width of the lane

Lane Detection distance [m]	Length of the lane segment detected
Yaw Angle [deg]	Angle between the longitudinal axis of the driver's vehicle and lane direction, tangent to the lane (also called "lane yaw angle")
Velocity [m/s]	Velocity of the driver's vehicle
Steering angle [deg]	Steering angle value
Yaw-rate [deg/s]	Value of the EV Yaw-rate (as from gyrometer)
VDD	Visual Distraction Detection
VTSD [s]	Visual Time Sharing Distraction
Head Orientation [deg]	Orientation of the driver's head (yaw, roll, pitch)
Head Position [deg]	Position of the driver's head
Object class	Classification of recognized object
Object id	Identifier of the recognized object as assigned by the radar
Coordinate X	Longitudinal position (distance) of the lead vehicle relative to the driver's vehicle
Coordinate Y	Lateral position of the lead vehicle relative to the driver's vehicle
Obstacle speed X	Longitudinal velocity of the lead vehicle
Obstacle speed Y	Lateral velocity of the lead vehicle

Table 3: inputs to the AdCoS model and their meaning.

Over the course of the project, it is expected that this input will be extended, once additional sensors and experimental data become available. In particular, it is expected that the following input will be added during the course of the project:

- Information about further surrounding traffic participants provided by an IBEO LIDAR sensor.
- Information about the current speed limit obtained by external cameras.
- Information about the current state and outputs of other machine agents.

3.2.2.3.2 Model specification

The inference engine requires an xml-specification of a probabilistic (or in the case of the DIR module, a BAD MoB) model satisfying the previously described FHMM or HDMT template structures that specifies the random variables $\boldsymbol{X} = \{X_1, \dots, X_n\}$ in the probabilistic model, the factorization of their JPD $P(X_1, \dots, X_n)$ as a product of (conditional) probability distributions, and the parameters specifying the (conditional) probability distributions defined by this factorization. As the exact xml-schema of the model specification is still subject to change, we will omit details of the xml-format and instead focus on what needs to and can be provided within the model specification.

The implemented inference engine allows the use of both discrete and continuous random variables, where each variable X needs to be specified by a unique name and a domain of possible values $Val(X)$. For a discrete random variable X, the domain $Val(X)$ is determined by the cardinality $|Val(X)|$, i.e., the number of possible values, as the domain of a variable X with cardinality n is implied as $Val(X) = \{1,2,..,n\}$. For a continuous variable X, the domain needs to be specified by the boundaries of an interval $[x_{min}, x_{max}]$. Furthermore, for each variable X, it has to be specified whether it is observable or hidden, i.e. whether during inference an assignment $X = x$ is known or unknown. By now, only discrete variables are allowed as hidden variables, i.e., continuous variables are assumed to be observable.

Sometimes, it can be of useful or necessary to map a continuous data signal to the domain of a discrete variable X with cardinality n. For this, discrete variables can be annotated by a set of $n-1$ values $b_1, \dots, b_{n-1}$ within the domain of the continuous signal that will serve as interval boundaries. Let $\mathbb{I}\{x \in A\}$ denote an indicator function defined as

$$\mathbb{I}\{x \in A\} := \begin{cases} 1 & \text{if } x \in A, \\ 0 & \text{otherwise,} \end{cases}$$

the discretization is realized by:

$$f(x,b) = \mathbb{I}\{x \in (-\infty, b_1]\} + (n)\mathbb{I}\{x \in (b_{n-1}, \infty)\} + \sum_{i=2}^{n-2} i\, \mathbb{I}\{x \in (b_i, b_{i+1}]\}.$$

The specified template variables $\boldsymbol{X} = \{X_1, \dots, X_n\}$ define a JPD $P(\boldsymbol{X}^{1:T})$, for which the model specification must provide a factorization in the form of

$$P(\boldsymbol{X}^{1:T}) = \prod_{i=1}^{n} P\left(X_i^1 | \boldsymbol{Pa}(X_i^1)\right) \prod_{t=2}^{T} \prod_{j=1}^{n} P\left(X_j^t | \boldsymbol{Pa}(X_j^t)\right).$$

Additionally, for each (conditional) probability distribution $P\left(X_i^1 | \boldsymbol{Pa}(X_i^1)\right)$ and $P\left(X_j^t | \boldsymbol{Pa}(X_j^t)\right)$, a set of parameters must be provided that can be used by the inference engine to calculate a probability for each combination of values $x \in X$ and $\boldsymbol{pa}(X) \in \boldsymbol{Pa}(X)$. Up until now, the inference engine supports discrete

CPDs in the form of table-CPDs and a limited set of continuous PDFs, namely Gaussians and mixtures of Gaussians.

When X represents a discrete variable, the set of parents $\boldsymbol{Pa}(X)$ is restricted to discrete variables. A CPD $P\big(X|\boldsymbol{Pa}(X)\big)$ can then be represented by a table that contains an entry $\theta_{x,\boldsymbol{pa}(x)} = P\big(x|\boldsymbol{pa}(X)\big)$ for each combination of $x \in X$ and $\boldsymbol{pa}(X) \in \boldsymbol{Pa}(X)$. Accordingly, a table-CPD is simply specified by a set of parameters $\theta_{x,\boldsymbol{pa}(x)}$ for all of these combinations.

When X represents a continuous variable, the set of parents may be composed of both discrete and random variables, however, conditional CPDs with continuous parents are currently restricted to linear Gaussian CPDs. Gaussian distributions are the most commonly used parametric form for continuous density functions [KF09]. The Gaussian probability density function for a variable X, specified by a mean μ and a variance σ^2, is given by:

$$p(x) = \mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right].$$

In the case of a more than a single continuous variable, the multivariate Gaussian is the most widely used joint probability density function. A multivariate Gaussian distribution over n variables $\boldsymbol{X} = \{X_1, \dots, X_n\}$ is commonly characterized by an n-dimensional mean vector $\boldsymbol{\mu} = (\mu_1, \dots, \mu_n)$ and a symmetric $n \times n$ covariance matrix Σ. Let Σ^{-1} denote the inverse covariance matrix and $|\Sigma|$ denote the determinant of Σ. The PDF is defined as:

$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}\,)\right].$$

As not every continuous PDF can be reasonable approximated by a Gaussian, the inference engine also supports the use of finite Gaussian mixtures, where a single density function is composed by a finite number of Gaussian components. Given a large enough number of component densities, each arbitrary PDF can be approximated by a mixture of Gaussians [KF09]. A mixture of Gaussians over a single continuous variable X, composed of a finite number of n Gaussians, defines a density

$$p(x|\Psi) = \sum_{i=1}^{n} \lambda_i \, \mathcal{N}(x|\mu_i, \sigma_i^2),$$

where $\Psi = (\mu_1, \dots \mu_n, \sigma_1^2, \dots, \sigma_n^2, \lambda_1, .. \lambda_{n-1})$ denotes a parameter vector consisting of n means, n variances and $n-1$ mixing proportions, with $\sum_{i=1}^{n} \lambda_i = 1$ and therefore $\lambda_n = 1 - \sum_{i=1}^{n-1} \lambda_i$. Correspondingly, a multivariate Gaussian mixture is defined as

$$p(\boldsymbol{x}|\Psi) = \sum_{i=1}^{n} \lambda_i \, \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_i, \Sigma_i),$$

where $\Psi = (\boldsymbol{\mu}_1, \dots \boldsymbol{\mu}_n, \Sigma_1, \dots, \Sigma_n, \lambda_1, .. \lambda_{n-1})$ denotes a parameter vector consisting of the n mean vectors, n covariance matrices and $n-1$ mixing proportions.

For each of the continuous PDFs presented, when conditioned by a set of discrete parent variables $\boldsymbol{U} = \{U_1, \dots, U_m\}$, we need to specify a different set of parameters for every value $\boldsymbol{u} \in Val(\boldsymbol{U})$. For example, for a single variable X this results in a number of m means and m variances with the PDFs:

$$p(x|\boldsymbol{u}) = \mathcal{N}(x|\mu_{\boldsymbol{u}}, \sigma_{\boldsymbol{u}}^2) = \frac{1}{\sigma_{\boldsymbol{u}\sqrt{2\pi}}} \exp\left[-\frac{(x-\mu_{\boldsymbol{u}})^2}{2\sigma_{\boldsymbol{u}}^2}\right].$$

In the case of continuous parents, the inference engine is restricted to linear Gaussian CPDs. Let X be a continuous variable with continuous parents $Y_1, \dots, Y_n$, a linear Gaussian CPD is defined by a set of parameters $\beta_0, \dots, \beta_n$ and a variance σ^2 with the PDF:

$$p(X|y_1, \dots, y_n) = \mathcal{N}(\beta_0 + \beta_1 y_1 + \dots + \beta_k y_n | \sigma^2).$$

3.2.2.3.3 Inference Engine

The inference engine is an RTMaps component that provides an interface to select an xml model-specification. In general, any model specification that satisfies the restriction above can be to answer probability queries. The actual query to be answered during inference is derived from the specified variables. Let $\boldsymbol{X}$ denote the set of hidden (and therefore discrete random variables) and $\boldsymbol{E}$ the set of non-hidden and therefore observable variables. At each time step t, the inference engine will answer probability query $P(\boldsymbol{X}^t|\boldsymbol{E}^{0:t} = \boldsymbol{e}^{0:t})$.

For the DIR module, the inference engine will be used to constantly, at each time step t, infer the joint belief state of intentions and behaviours given all the available evidence about actions and context observations observed up to this point: $P(I^t, B^t|\boldsymbol{a}^{0:t}, \boldsymbol{o}^{0:t})$. Given this joint belief state, the DIR module can provide the marginal belief states for intentions $P(I^t|\boldsymbol{a}^{0:t}, \boldsymbol{o}^{0:t})$ and behaviours $P(B^t|\boldsymbol{a}^{0:t}, \boldsymbol{o}^{0:t})$.

The inference engine implements an algorithm for *exact* inference based on algorithms for variable-elimination and conditioning [KF09]. In contrast to approximate inference based on sampling techniques, the use of exact inference ensures that the resulting belief state is deterministic with respect to the available evidence and therefore more easily certifiable.

3.2.2.3.4 Data Pre-processing

During runtime, the inference engine requires information or evidence about the current values of all observable random variables (i.e., actions and observations), which must be derived from the input provided to the DIR module by the perception layer of the AdCoS for adapted assistance. To obtain this evidence, the input from the different modules in the perception layer needs to be synchronized and summarized to discrete time steps, potentially transformed and annotated, and provided in an order that enables the inference engine to assign the obtained values as evidence for the observable variables defined in the corresponding probabilistic model. This is done by the data pre-processing component, which is accordingly composed of three minor components for data synchronization, transformation and mapping. A screenshot of the component in RTMaps is shown in Figure 22.

The data pre-processing component is highly domain- and task-dependent, as all available processing is tailored to the specific set and properties of the available input. It consists of the following steps:

- Data Synchronization

As the different inputs are provided asynchronously and in potential different frequencies, it is necessary to gather the different sensor values and combine them into synchronized time slices that can be processed by the inference engine in its entirety. For this, an arbitrary input with high and consistent frequency (e.g., the radar) is chosen as a central clock. Whenever new data for this input arrives, the most up-to-date and complete data (e.g., a full set of radar object data) is passed to the data transformation component. In future versions, we will investigate the use of sequential data for high-frequency data, i.e., instead of discarding outdated data, we extend the BAD MoB models to use sequences of data as evidence.

- Data Transformation

While many variables of the specified model should directly correspond to the available input, additional information may be defined as functions of these inputs, e.g., rates of changes, time headway, or time-to-contact values. This component is therefore used to enrich the available data by transforming certain data, or by calculating new, unavailable sensor values.

- Mapping

Finally, the synchronized and processed evidence needs to be mapped to corresponding random variables in the BAD MoB model. The mapping component collects the synchronized evidence provided by the data transformation component and arranges them in a vector according to an order implied by the model specification.

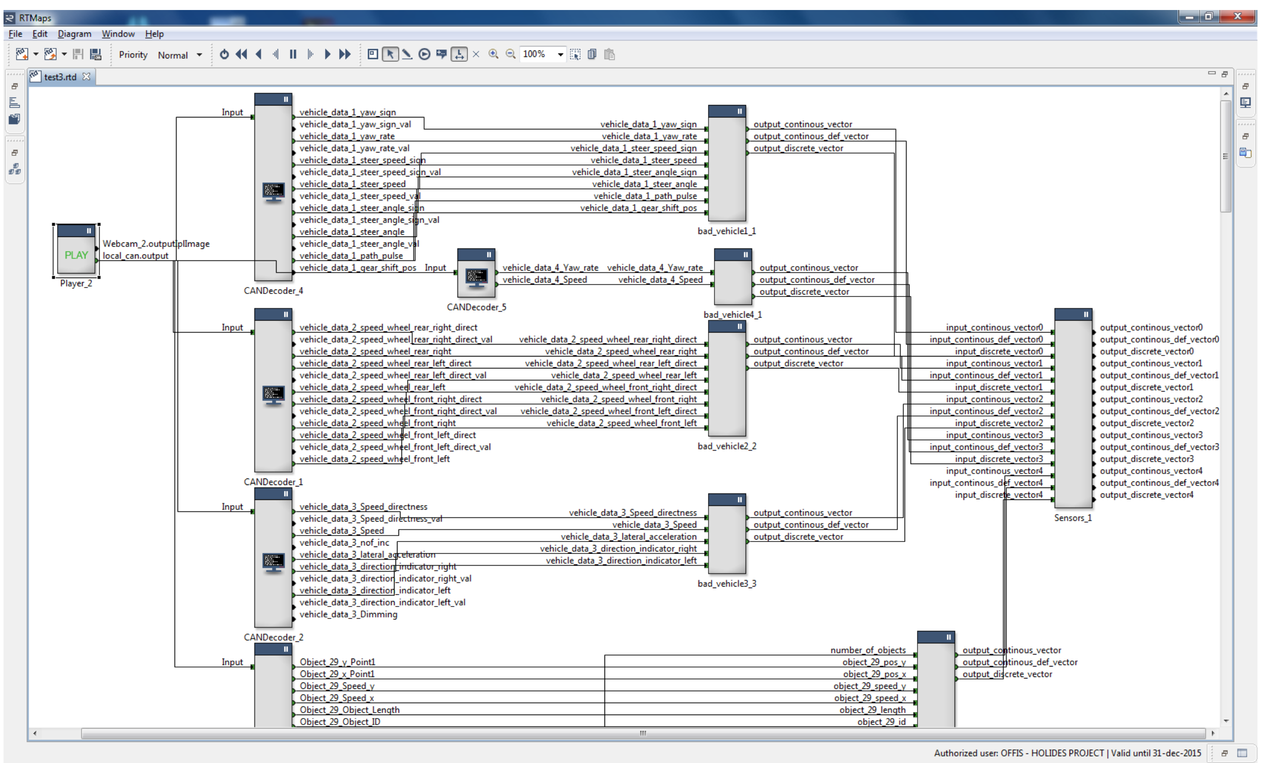


Figure 22: Screenshot of the data pre-processing of the DIR module in RTMaps.

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

3.2.2.3.6 Output
The primary output of the inference engine are marginalized belief states over all hidden variables, defined in the model specification in the form of conditional probability distributions. Under the assumption that the DIR module is used as intended (i.e., intention recognition), this relates to intentions and behaviours. An initial classification of the current intentions and behaviours are provided by the mode of the belief state, i.e., the most probable intention and behaviour. If requested, additional output can be provided in the form of behavioural assessments that measures the likelihood of the actions of the human driver with respect to the underlying BAD MoB model. This assessment can be used as an indicator on whether the driver shows an unlikely behaviour for the given evidence (once again with respect to the underlying model).

Output	Description
Intention belief state	Marginalized belief state of the driver's intentions given all available information $P(I^t \| \boldsymbol{a}^{0:t}, \boldsymbol{o}^{0:t})$
Current Intention	The most probable intention, mode of $P(I^t \| \boldsymbol{a}^{0:t}, \boldsymbol{o}^{0:t})$
Behaviour belief state	Marginalized belief state of the driver's performed behaviour given all available information $P(B^t \| \boldsymbol{a}^{0:t}, \boldsymbol{o}^{0:t})$
Current Behaviour	The most probable behaviour, mode of $P(B^t \| \boldsymbol{a}^{0:t}, \boldsymbol{o}^{0:t})$
Behaviour assessment	Log-likelihood of the last k taken actions of the driver $\log P\left(\boldsymbol{a}^{k:t} \| \boldsymbol{a}^{0:k-1}, \boldsymbol{o}^{0:t}\right)$, where $0 \leq k \leq t$ is a user-defined threshold

Table 4: outputs of BAD MoB model and their meaning.

Additional output, e.g., allowing for easier interpretation or an assessment of confidence, will be added during the course of the project.

3.2.3 Modelling the Adapted Automation System

A major role in modeling the adapted automation is played by the driver model, as it can be seen by the system architecture in chapter 3.5. For the adapted automation the driver modeling is done by tailoring the WP3 tool and method CONFORM (Conflict recognition by image processing methods) to the overtaking use case. Figure 23 shows this tailoring.

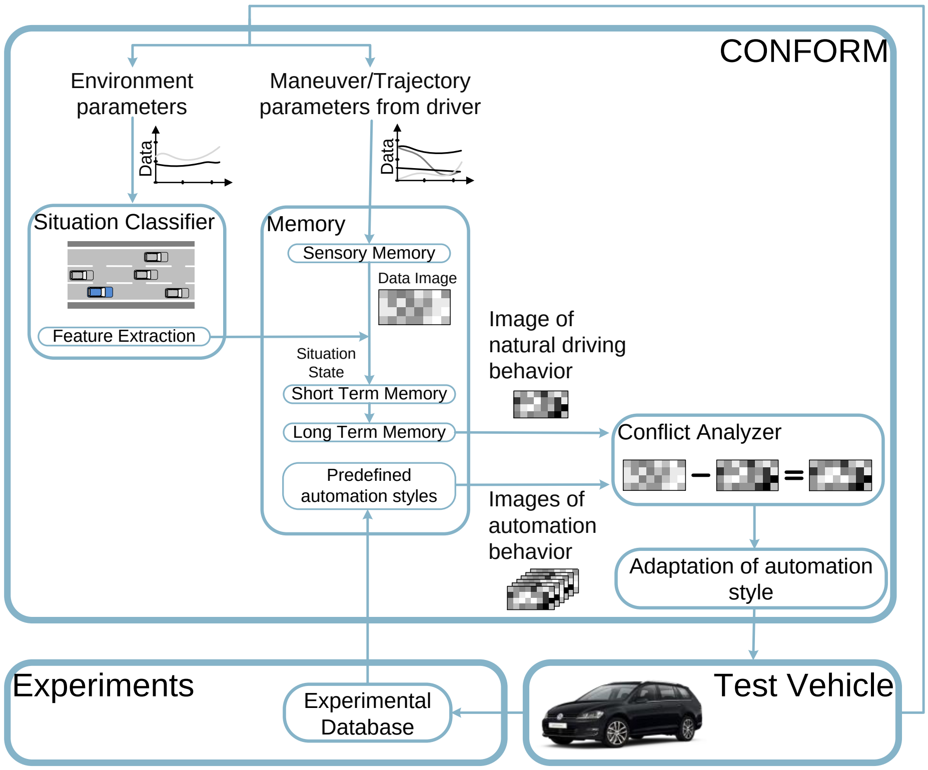


Figure 23: Structure of CONFORM tailored to WP9 use case

The aim of CONFORM is to improve the cooperation between the driver and the machine agent. Therefore CONFORM recognizes conflicts between the driver behavior and the automation behavior and avoids them afterwards through learning the natural driving behavior. More precisely CONFORM is able to adapt the system to certain driving styles situation-dependently by adapting the preference of maneuvers and their trajectory. This includes different driving styles for the approaching of the vehicle, different driving styles of following the vehicle as well as different driving styles for the overtaking maneuver itself.

For that reason CONFORM consists of three main modules as illustrated in the previous Figure 23:

- The situation classifier to adapt to the situation
- The memory to adapt to the natural driver behavior
- The conflict analyzer to recognize the urgency for an adaptation

As we explained in D3.4 CONFORM can be grouped in use case dependent parts and use case independent parts. Use case dependent parts are:

- The inputs to the situation classifier
- The extracted features in the situation classifier
- The inputs to the memory
- The definition of predefined automation styles

Use case independent parts are:

- The methods within modules, i.e. the methods for clustering the situation, learning the natural driving behavior and recognizing conflicts.

A first draft of modeling the use case dependent part can be already found in D9.2. There we defined the inputs to the situation classifier and to the memory. For the use case independent parts we refer to D3.4 for a detailed description. Currently we analyze the results from the WP5 study described in D5.3 to continue the modeling of the use case dependent parts. One goal will be to reduce the inputs to feasible parameters. Another goal will be to fine tune the feature extraction in the situation classifier with focus on the definition of the state intervals. Table 5 shows with an example from D3.4 how state intervals and feature extraction are connected.

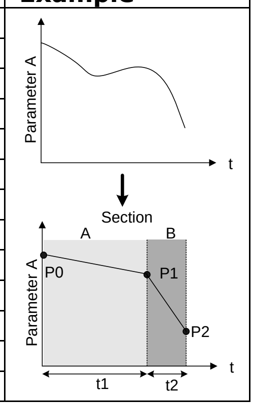
Feature	**Value**	**State**	**Example**
Slope in section A	<-10	0	
	[-10,0)	1	
	0	2	
	[0,5)	3	
	>5	4	
Length of section A in seconds	[0,0.2)	0	
	[0.2,0.5)	1	
	[0.5,1.5)	2	
	>1.5	3	
Start value in section A	[0,0.2)	0	
	[0.2,0.5)	1	
	[0.5,0.75)	2	
	[0.75,1]	3	

Table 5: Example of a feature extraction based on linearization and categorization

Also the definition of the predefined automation styles is currently on-going. A first analysis (see Table 6) confirms different driving styles.

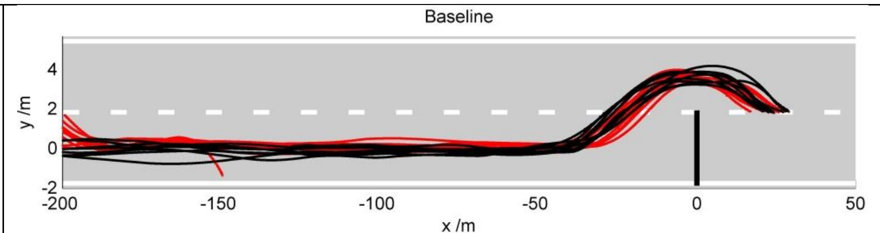
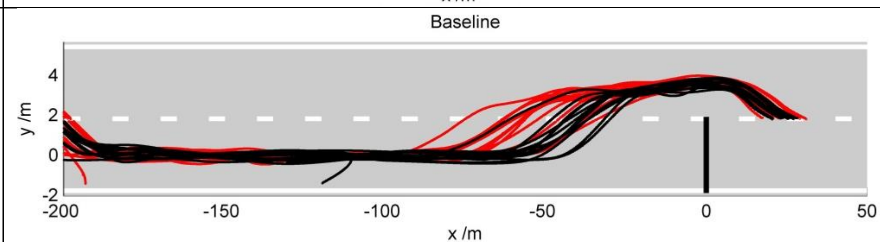
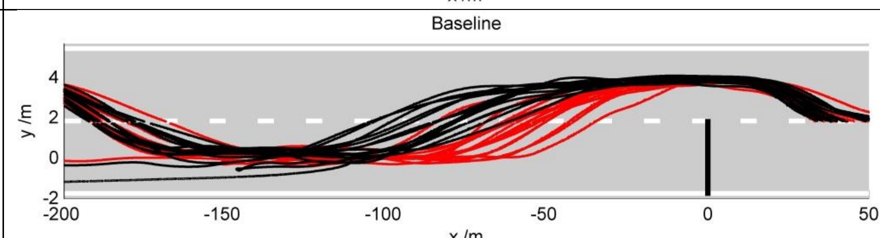
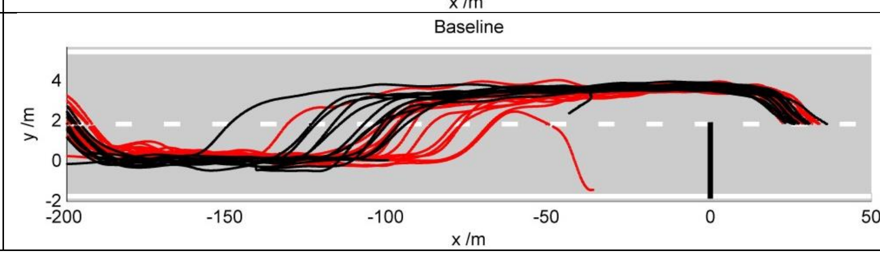
Lane changes from different drivers. The driver approached different vehicles which drove 100kmh. The ego velocity was 120 kmh. No other vehicles were on the left lane. This scenario was done twice (black: first run, red: second run).	
Time point: very late Length of lane change: short	
Time point: late Length of lane change: medium	
Time point: early Length of lane change: long	
Time point: early Length of lane change: short	

Table 6: First Results from a simulator study conducted in WP5 (see D5.3).

Based on these first findings we will implement 3-5 different driving styles in the next step for the lane change maneuver.

3.2.4 Modelling and Evaluation of the Adaptive HMI

With the current progress in automation of the driving task and the benefits such automation offers, a shift in task allocation from human operator to machine agent is the most promising in terms of traffic safety. However, automation of the driving task or parts of it might not always be wanted. One reason to avoid automatically switching to automatic driving is the costs of taking over again <u>after</u> automation. These costs might be higher than the benefits offered by taking over small parts of the driving task. Therefore, adapting the agents and their interaction might also be an option. Such adaptation can be achieved by adapting the HMI. Regarding the evaluation of its AdCoS, TAKATA plans to implement both options. The preliminary experimental plan is shown in the following table:

		Factor A: (visual and cognitive) Distraction [RM]					
		A_1 not distracted		A_2 distracted			
		Factor B: Adaptive HMI [RM]					
		B_1 no	B_2 yes	B_1 no		B_2 yes	
		Factor C: Adaptive ADAS [RM]					
		C_1 no		C_1 no	C_2 yes	C_1 no	C_2 yes
Factor D Driving mode [RM]	D_1 manual	$\bar{Y}_{1111}$	$\bar{Y}_{1211}$	$\bar{Y}_{1111}$	$\bar{Y}_{2111}$	$\bar{Y}_{2211}$	$\bar{Y}_{2221}$
	D_2 automatic	$\bar{Y}_{1112}$	$\bar{Y}_{1212}$	$\bar{Y}_{2112}$	$\bar{Y}_{2122}$	$\bar{Y}_{2212}$	$\bar{Y}_{2222}$

Table 7: Experimental conditions for the TAKATA AdCoS.

Note: RM=repeated measures.

This ambitious test schedule will be implemented as repeated measures design and will only be applied for the predefined scenario. Depending on the implementation in the simulator and the time needed for testing, a reduction of the schedule might have to be considered.

3.2.5 Simulation of LCA and Adapted Automation System on a Virtual-HCD platform

In addition to real AdCoS for real cars, a *Virtual Human Centred Design (V-HCD)* platform is developed in HOLIDES by IFSTTAR, in partnership with

INTEMPORA and CIVITEC. In summary, this V-HCD platform is supported by 2 main tools, Pro-SIVIC (provided by CIVITEC) and RT-MAPS (coming from INTEMPORA), interfaced with IFS ADAS/AdCoS simulation models and a Cognitive Simulation Model of the Driver, named COSMODRIVE. The following figure provides an overview of this V-HCD platform.

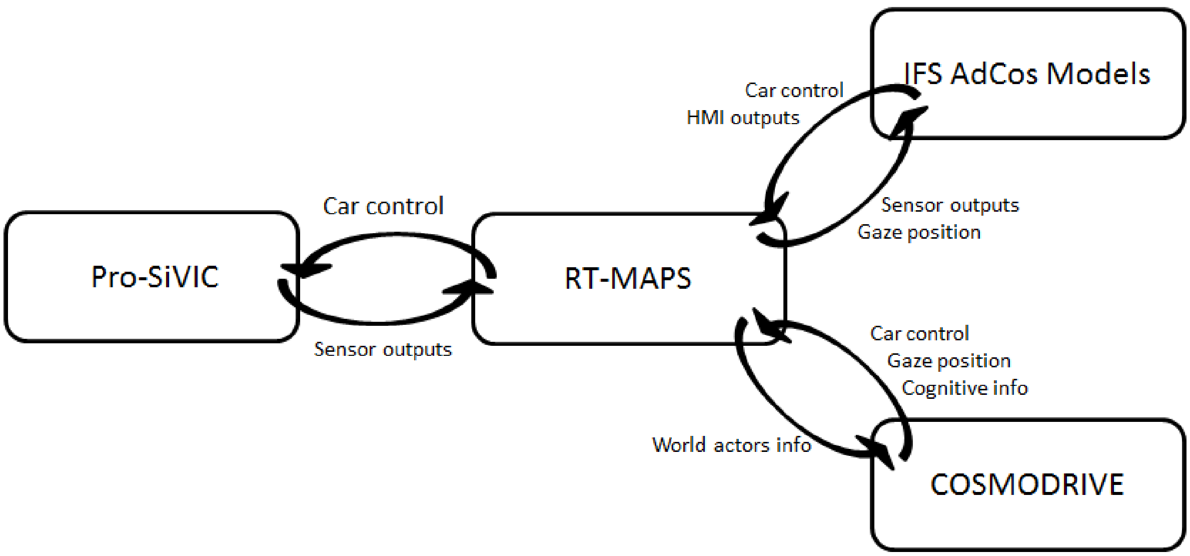


Figure 24: Overview of "Virtual HCD platform"

In the frame of HOLIDES, the challenge is to have a simulation platform for virtual design and test of future AdCoS that integrates (1) a driver model able to drive (2) a virtual car equipped with (3) virtual ADAS and (4) AdCoS functionalities (based on MOVIDA functions: Monitoring of Visual Distraction and risks Assessment).

The following figure provides a more detailed view of this V-HCD platform use in HOLIDES, as an example of a tailored HF-RTP in WP9 for AdCoS virtual design in the automotive domain. This V-HCD platform will be used in HOLIDES to simulate an AdCoS (based on MOVIDA) and its uses, for both Normal and Critical Use Cases presented before. The AdCoS based on MOVIDA functions to be simulated on the V-HCD is an integrative co-piloting system supervising several simulated Advanced Driving Aid sub-Systems (ADAS), to be managed in an adaptive and cooperative way by the MOVIDA module, according to the drivers' visual distraction states and to the assessment of situational risks (Figure 25).

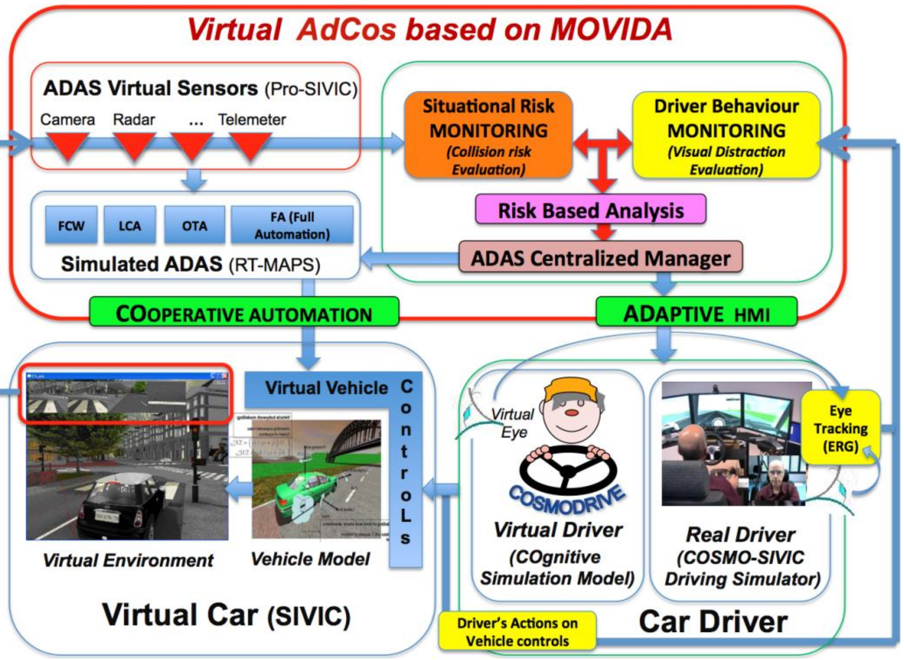


Figure 26: Virtual AdCoS (based on MOVIDA) simulated on the V-HCD platform (COSMODRIVE + Pro-SIVIC + RT-MAPS)

The main ADAS simulated in this V-HCD platform and to be combined in this AdCoS are a Collision Avoidance Systems (e.g. FCW; Forward Collision Warning), a Lane Change Assistant (LCA), an Overtaking Assistant (OTA) and Full Automation devices (FA). Regarding their Human-Machine Interaction modalities, all these ADAS will be liable to interact with the driver in 2 main ways: warning and vehicle control taking (via partial or full automation). Adaptive and cooperative abilities in this AdCoS are supported by a set of monitoring functions (i.e. MOVIDA module), which supervise the drivers' activity (by assessing their behaviours and their visual distraction state) and, from the other side, to evaluate the risk of accident in the current traffic situation (forward collision risk of car A with the slow vehicle C ahead, or rear collision risk with vehicle B).

A part of this platform and tool chain is already operational and has been jointly used by INT, CVT and IFS (LIVIC) for ADAS design and development in other projects. Figure 27 presents an example of a stereovision algorithm and a lane marking detector (provided by IFSTTAR / LIVIC) executed in RTMaps and fed by synthetic sensor data generated in the ProSIVIC simulator (here a stereovision camera with simulated fog).

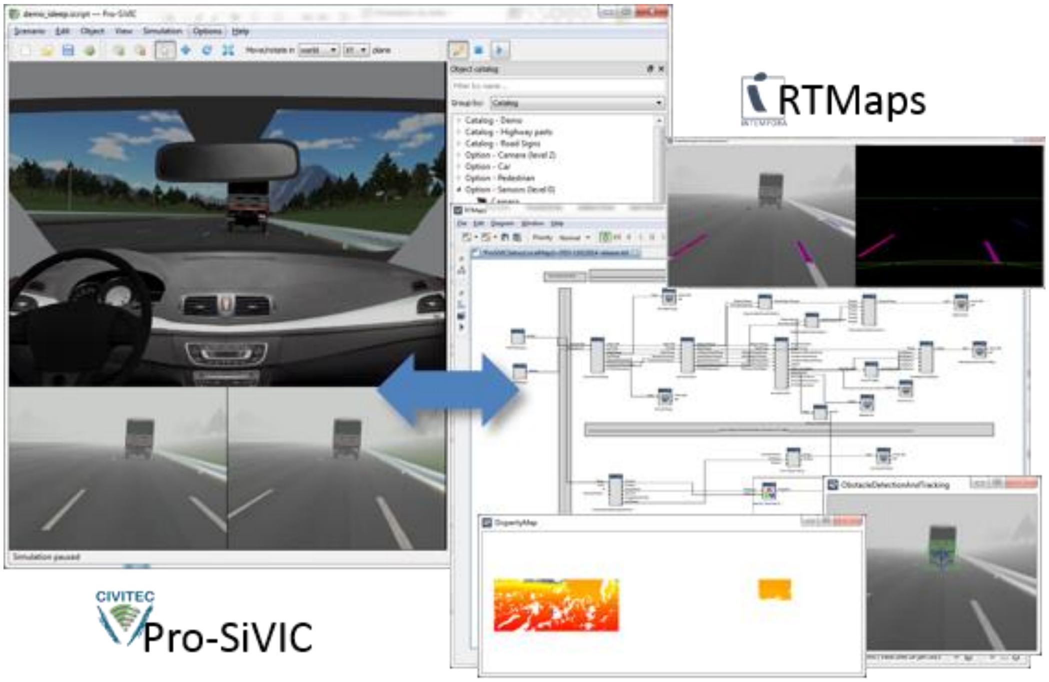


Figure 28: Stereovision algorithm virtual testing in RTMaps

By adding a cognitive simulation model to RT-MAPS, PRO-SIVIC and IFS DASA models, it is expected to simulate, not only the technical components of future driving aids, but also the final use by a real human driver, which is of prior importance for ergonomic specifications and test of AdCoS. According to thes specific human factor issues, this integrative V-HCD platform will be firstly used in WP9 to simulate the driving performance of a human driver with and without the driving AdCoS (from normal behaviours to critical behaviour, due to visual distraction) in order to support the AdCoS virtual design process at 2 main levels. At the earliest design stages, the COSMODRIVE driver model will allow us to estimate human drivers' performances in case of unassisted driving, in order to identify critical driving scenarios for which a given AdCoS could be provided for supporting real drivers. Then, after virtual AdCoS development, it will be possible to virtually assess its *effectiveness* for different variations of the critical scenarios previously identified, and to check its *efficiency* according to real human drivers' needs (as assessed through COSMODRIVE-based simulation).

It must be also noted that this virtual design approach and tool chain will be easily to share with real car demonstrators, due to the RT-MAPS functionalities. Indeed, the software tools presented above are interfaced together to be able to inter-operate in real-time and to cover the maximum range of the AdCoS development steps:

- Offline developments, tests, validation
- Online real-time operation in standalone or distributed mode

The following Figure (Figure 29) presents the workflow proposed for the AdCoS development tool chain. The upper part presents the tool chain for offline developments in a simulator environment or via sensors data playback functions, while the lower part presents the porting of the developed applications onto a real prototype.

The left column presents the sensors and actuators interface (either simulated, recorded and played back, or interfaced in real-time).

The column in the middle presents RTMaps, in charge of integration and execution of high-level functions such as data acquisition, image processing, signal processing, decision and data visualization (eventually via Qt / QML).

And the column on the right presents command-control laws which can be run either in co-simulation (offline) in Simulink, or in real-time on a dSPACE target for instance.

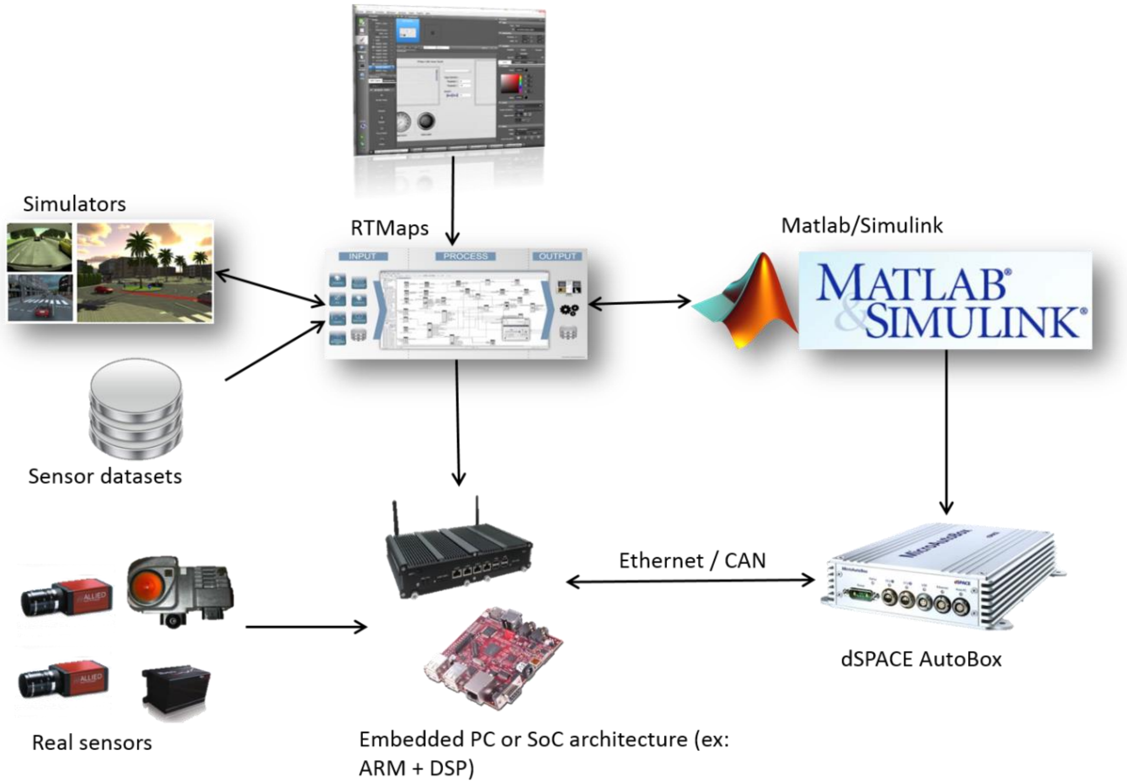


Figure 29 : AdCoS development tool chain: from virtual design to reality

This tool chain presented above will be used in WP9, among others, in the CRF demonstrator and by the IFSTTAR V-HCD simulator.

3.3 Human-Machine Interaction for the AdCoS

This section describes which input/output channels are available to carry the communication between the operator (human-agent) and the AdCoS (machine-agent). Whenever possible, special considerations will be pointed out, in terms of innovative and unconventional user interaction (e.g. gestures, voice interaction, etc.).

3.3.1 Human Machine Interaction for LCA System

The preliminary HMI concept has been based on the information included in Table 2. The concept also includes alternative graphics for the adaptive HMI.

The HMI of the LC Assistant app will be mainly visual, thus it is important that it achieves a good legibility, contrast, resolution and field of view.

In particular, for visual information, important properties would be:
- brightness/intensity
- visual acuity/spatial contrast sensitivity
- colour specification
- perceptual organisation:
    - figure-ground organisation,
    - grouping principles (proximity, similarity, continuity, closure, common elements with common motion tend to be grouped) etc.

Acoustic outputs will be also employed, for example in association with visual information to warn the driver when his/her visual load is expected to be already high.

Like the visual output, also the acoustic output has to respect some general requirements in terms of sound pressure level, frequency range, and spatial resolution. First of all, it is important that the sounds used are well distinguishable from the other environment and traffic sounds, secondly the sounds' loudness level need to be calibrated in order to transmit the proper level of urgency without creating annoyance.

Some example of signals that can be used are:
- sweeping sounds for immediate acoustic feedback
- patterns of segments with constant pitch for short-term acoustic feedback
- two-times chimes, high-low non recurrent for long-term acoustic feedback

As regards the priority of the different interaction modality, the visual channel has to be considered as the main feedback channel, followed by the acoustic one.
In some cases the two channels could work separately (or only one output channel is available), but it is better to design them to work together (in multimodality) in order to give a more appropriate feedback to the driver.

In multimodality mode, the prioritization of the output channels is planned according to the information in Table 8.

Task		Decision	Preparation	Execution
1. Changing the original lane	**Cognitive load**	medium	medium	medium
	Visual load	high	high	medium
	Information based on driver intention?	Information provided **only** if intention of changing lane was detected (driver intention module OR left indicator activated)	Information provided **only** if intention of changing lane was detected (driver intention module OR left indicator activated)	Information based on detection of LC manoeuvre
	Interaction modality	**Option 1.1** (driver not distracted AND no car approaching on the left): **visual** **Option 1.2** (driver distracted OR cars approaching on the left): **visual + acoustic**	**Option 1.1** (driver not distracted AND no car approaching on the left): **visual** **Option 1.2** (driver distracted OR cars approaching on the left): **visual + acoustic**	**Option 1.1** (driver not distracted AND no car approaching on the left): **visual** **Option 1.2** (driver distracted OR cars approaching on the left): **visual + acoustic**
2. Vehicle Passing	**Cognitive load**	low	low	low
	Visual load	medium	low	medium
	Information based on driver intention?	Information provided **only** if intention of changing lane was **NOT** detected (driver intention module OR right indicator activated)	Information provided **only** if intention of changing lane was **NOT** detected (driver intention module OR right indicator activated)	Information provided **only** if intention of changing lane was **NOT** detected (driver intention module OR right indicator activated)
	Interaction modality	**Option 2.1: Visual**	**Option 2.1: Visual**	**Option 2.1: Visual**
3. Re-entering into the original lane	**Cognitive load**	medium	medium	medium
	Visual load	high	high	medium
	Information based on driver intention?	Information provided **only** if intention of changing lane was detected (driver intention module OR indicator activated)	Information provided **only** if intention of changing lane was detected (driver intention module OR indicator activated)	Information based on detection of LC manoeuvre
	Interaction modality	**Option 3.1** (driver not distracted AND no car on the right): **visual** **Option 3.2** (driver distracted OR cars on the right): **visual + acoustic**	**Option 3.1** (driver not distracted AND no car on the right): **visual** **Option 3.2** (driver distracted OR cars on the right): **visual + acoustic**	**Option 3.1** (driver not distracted AND no car on the right): **visual** **Option 3.2** (driver distracted OR cars on the right): **visual + acoustic**

Table 8: HMI overall concept (information provided and interaction modality).

3.3.1.1 Changing the original lane

Option 1.1 (driver not distracted AND no cars approaching on the left): visual interaction

Option 1.2.1 (driver distracted): visual + acoustic interaction

Option 1.2.2 (car approaching on the left): visual + acoustic interaction

Figure 30: HMI to support the driver while changing the original lane (task 1)

3.3.1.2 Vehicle Passing

Option 2.1: visual interaction

Figure 31: HMI to support the driver while passing a vehicle (task 2)

3.3.1.3 Re-entering into the original lane

Option 3.1 (driver not distracted AND no cars on the right): visual interaction

Option 3.2.1 (driver distracted): visual + acoustic interaction

Option 3.2.2 (car on the right): visual + acoustic interaction

Figure 32: HMI to support the driver while re-entering into the original lane (task 3).

3.3.2 Human Machine Interaction for ID System

The human machine interaction for the ID AdCoS can be separated into:

1. Adaptation of the driver model
2. Driver requests
3. Warnings

The underlying driver model analyses the driving behaviour when the vehicle is driven in the manual mode. In order to improve the acceptance of the autonomous driving style, the machine agent adapts to the driver's characteristics provided by the driver model.

Driver requests are sent to the machine agent when the vehicle is driven by the machine agent. The input devices to send requests are:

- Brake pedal

- Acceleration pedal
- Steering wheel

It can be seen that the input devices are standard devices in all vehicles which can be intuitively used by all drivers. The inputs are interpreted by the machine agent to perform a requested task, if possible. For instance, if the machine agent does not keep a comfortable distance to the preceding vehicle, a short impulse to the brake pedal will lead to a larger distance and the machine agent will be adapted not to decrease the distance again into an uncomfortable range.

Warnings will be sent by the machine agent in an acoustic and visual way to the human driver when either a request cannot be satisfied while the machine agent has the driving task or when the human driver is not fully aware of the situation and performs risky manoeuvres.

3.3.3 Human Machine Interaction for the Adaptive HMI

There are several issues that need to be considered when designing an Adaptive HMI. First of all, it has to be assured that all requirements that were defined from a human factors perspective are met. These requirements were defined in Deliverable D9.1 and are constantly updated (see below for an example). Once this is assured, the conditions in which the HMI will be adaptive have to be defined. These conditions were defined in **Fehler! Verweisquelle konnte nicht gefunden werden.**. For the TAKATA use cases at least two elements of the HMI need to be modelled regarding interaction:

- the instrument cluster and
- the infotainment system in the center stack.

Other elements concern the display of the blind spot detection or additional hard- or software features that direct the drivers attention. Possibilities for the latter are currently under review by TAKATA. For the main elements named above prototypical HMI solutions are shown in the next figures.

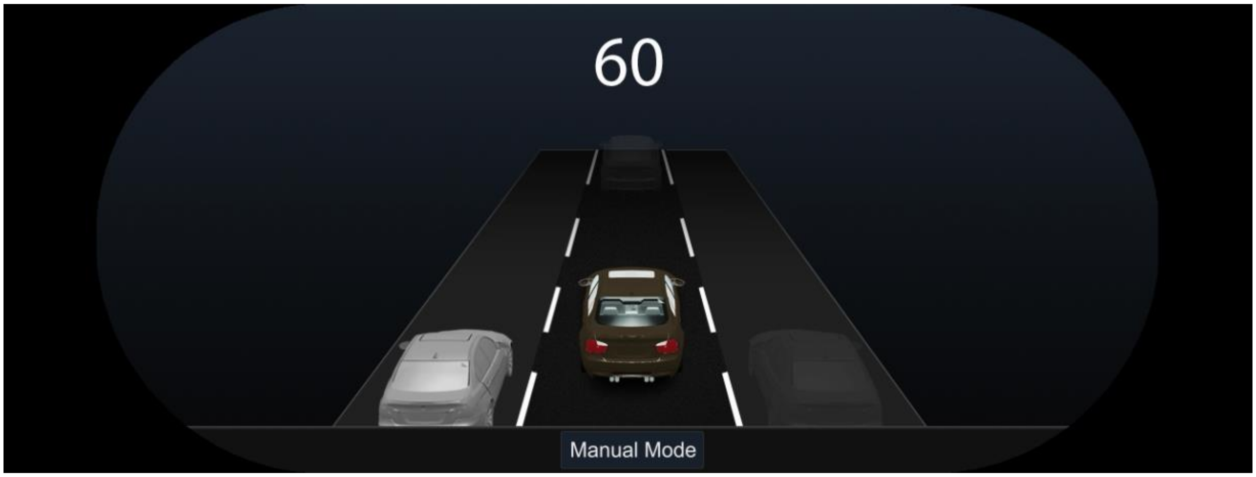


Figure 33: Draft HMI for the instrument cluster for manual mode in a noncritical situation without lane-change intention of the driver.

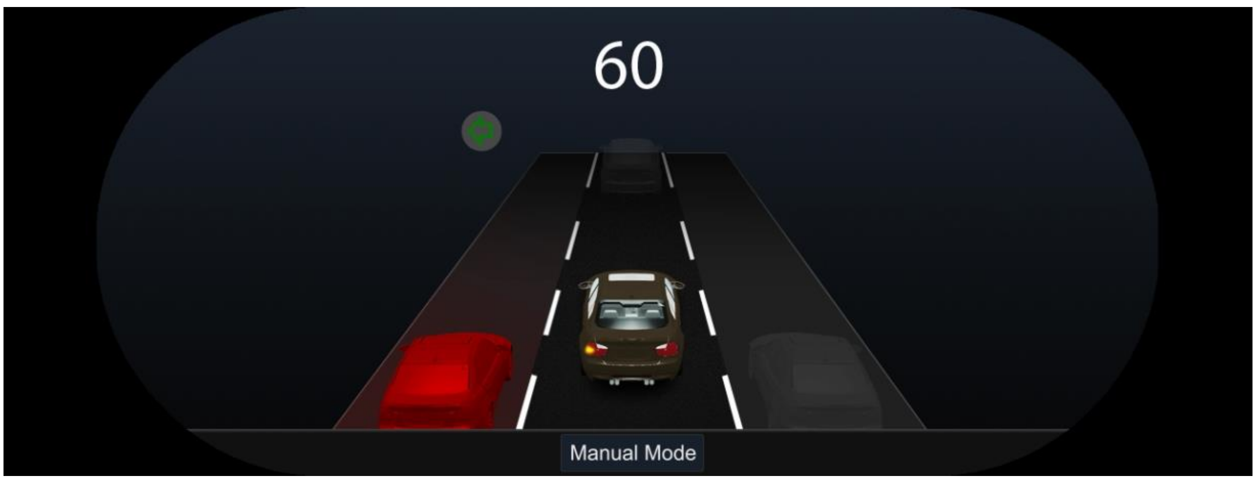


Figure 34: Draft HMI for the instrument cluster for manual mode in a critical situation with lane-change intention of the driver.

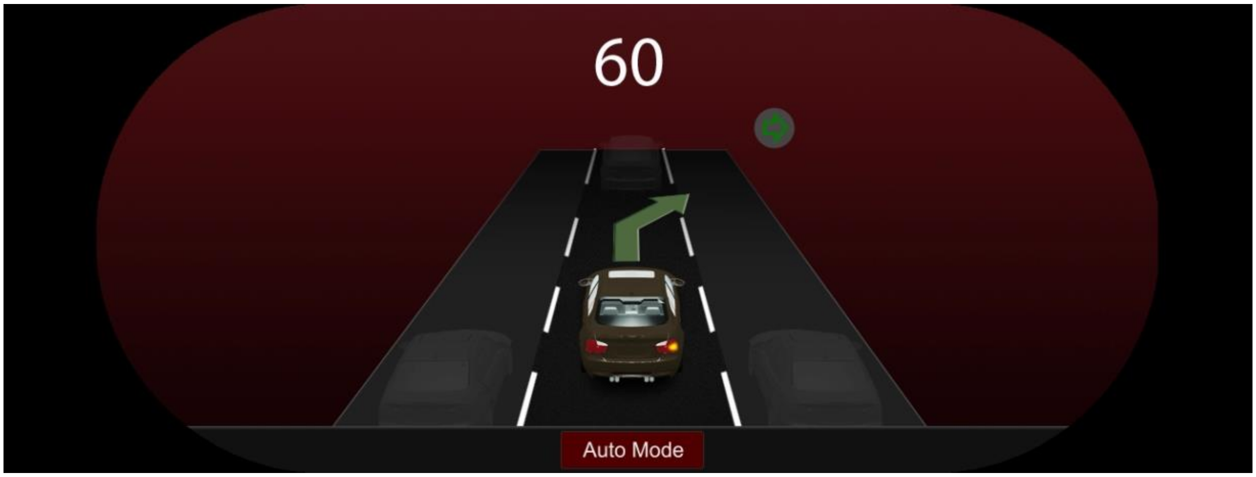


Figure 35: Draft HMI for the instrument cluster for automatic driving with the automation informing the driver of an upcoming lane change.

On the second HMI element (the centre stack display) the distraction task will be presented. In case of an upcoming critical situation (both in manual and automatic mode) the display will change. Different options for this change are available. One option is shown in 36 and 37. More details are available inside the WP5 deliverable, where these experiments are carried out.

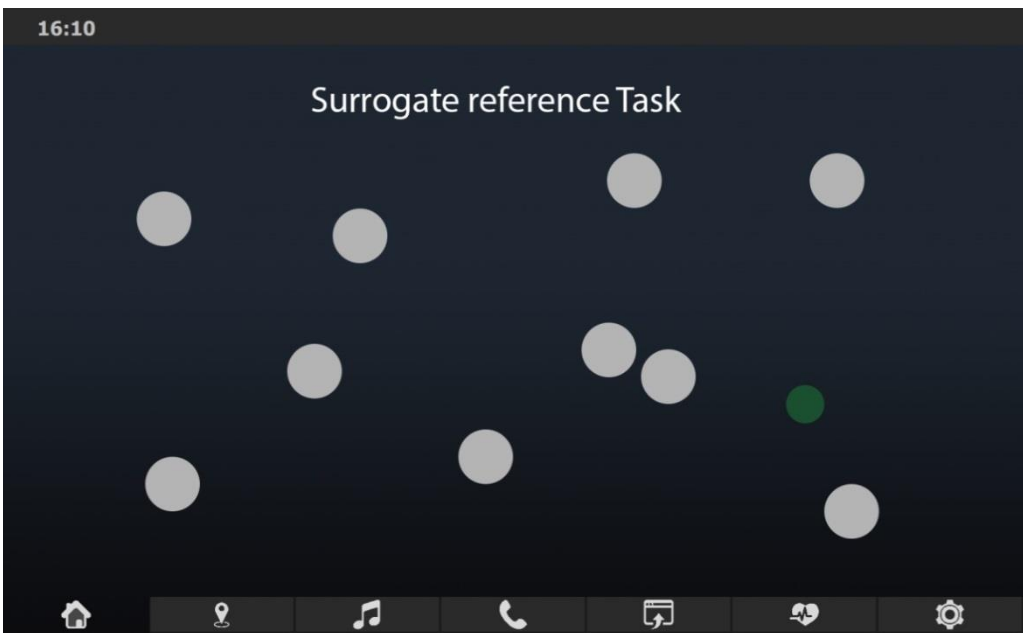


Figure 36: Draft HMI for the center stack display showing the distraction task.

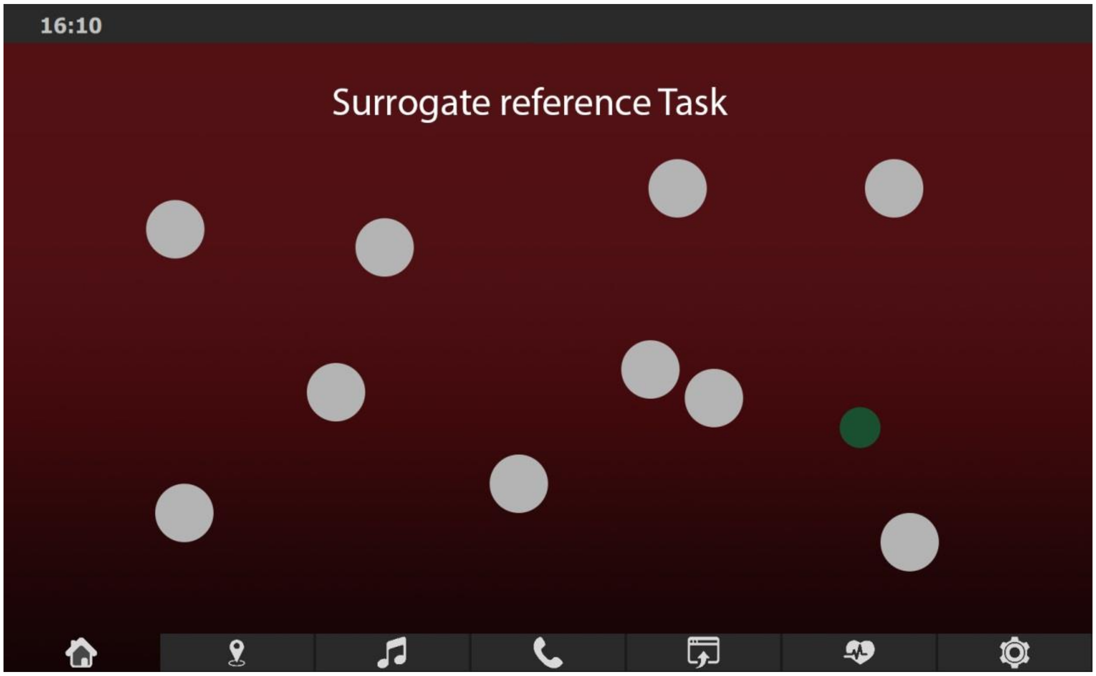


Figure 37: Draft HMI for the centre stack display showing the distraction task in the condition when attention is required on the road.

From a human factors perspective it is important that adaptation is self-explaining to the driver and that it does not lead to additional distraction. Nevertheless it must insist enough to make sure that the driver re-directs their attention to the road. Thus, the draft shown in Figure 37 will undergo further development depending on the outcome of user tests and expert ratings. Whether the final version fulfils the requirements will be tested in simulator experiments.

3.4 Requirements Update

Requirements for the automotive domain were first collected in Deliverable D9.1. As mentioned there, the following scheme has been proposed for the process of requirements collection:

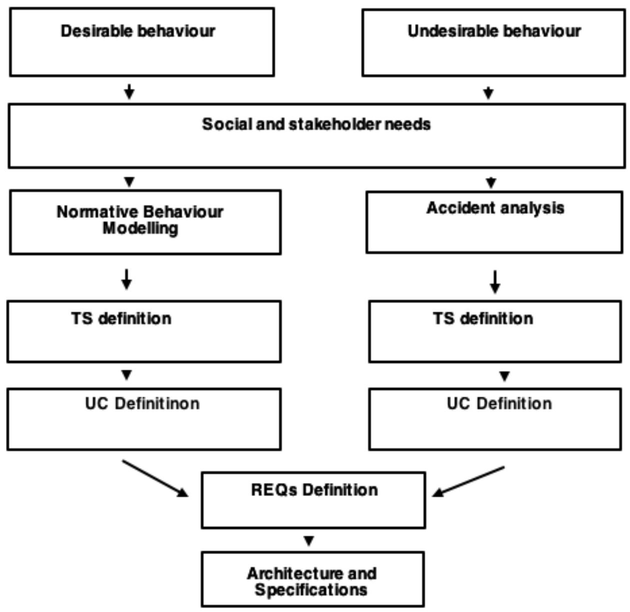


Figure 38: logic scheme for requirements definition process.

The requirements are based on a two-sided requirements-analysis and each one is characterized by the following parameters:

- Relevance
- Development Process Step
- Classification
- Type ((non-)functional, operational)
- Proof (criteria how fulfilment is measured)
- UC-reference

In the automotive domain, a total of 166 requirements have been collected, which are based on the target scenario and the related use cases (again, see D9.1 for details).

Because of the dynamic nature of the development process in research projects, these requirements must be checked regularly against potential new project states. This checking might result in requirements having to be updated. In the case of the automotive application WP, most requirements remained the same. However, the following changes were made (explanation for changes given in brackets):

- WP9_TWT_AUT_REQ05_v0.1: Deleted. (Covered by WP9_TWT_AUT_REQ03)

- WP9_TWT_AUT_REQ07_v0.1: Deleted (Eye-tracking is replaced by facial features, eye gaze and eye blinking)
- WP9_TWT_AUT_REQ09_v0.1: Deleted (Not really a requirement, describes approach).

In addition and with reference to the adaptation issue in HOLIDES project, the following requirements have been identified and considered (three are provided as example):

- WP9_CRF_AUT_REQ3_v1.0 ⇒ The classifier of the driver cognitive state shall be able to perform the classification with a Correct Classification Rate (CCR) of (80÷85)%. The AdCoS is based on driver's status, so this classification is used for the adaptive strategies.
- WP9_OFF_AUT_REQ8_v1.0 ⇒ After an initial offline learning phase, the Bayesian driver model must be able to classify the driver's manoeuvre intention (e.g. lane-change) with a CCR of (80÷85)%. The AdCoS is based on the driver's status, so this classification is used for the adaptive strategies.
- WP9_CRF_AUT_REQ10_v1.0 ⇒ When the vehicle aims at leaving the current lane (e.g. for an overtaking) the system shall assist her/him, indicating the right time and moment, taking into account the internal and external situation. Driver is supported in lane changing or lane departure, for overtaking manoeuvre.

The process of updating requirements will be repeated constantly.

3.5 System Architecture and Specifications

The main goal of this session is to provide a description of the main AdCoS specifications and the related system architecture.

3.5.1 System Architecture for the Adapted Assistance AdCoS

The Adapted Assisted AdCoS – that is the LCA system – is implemented on the CRF test-vehicle (TV), which is a 500L, sketched in the figure below together with the sensors installed on-board.

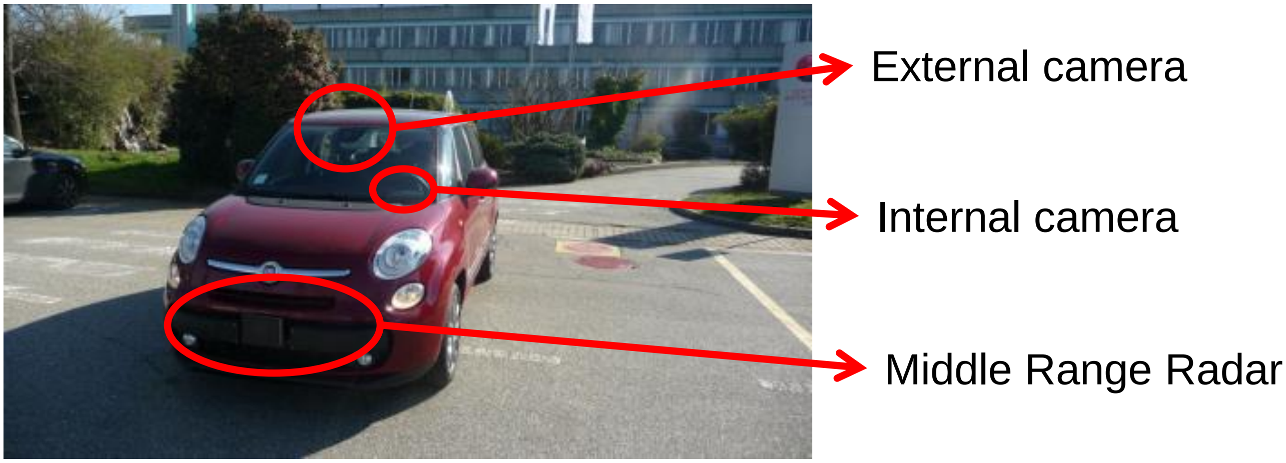


Figure 39: CRF test-vehicle where the AdCoS will be implemented. Sensor installation is also represented.

The TV is equipped with the following sensors, for the detection of both the external and the internal environment:

- External camera to detect the edges of the lanes on the road and the relative position of the ego-vehicle in the lane.
- Middle Range Radar (MRR) sensor to detect surrounding obstacles, whose main characteristics are.
    - Range = 50m for pedestrians; 280m for obstacles.
    - Field of View = ±30 deg.
    - Range for obstacles = 280m
    - Classification capability: cars and pedestrians.
- Internal camera to detect the head position of the driver (and where he/she is looking at).

The following logical scheme shows the core of the LCA AdCoS, represented by a MDP co-pilot framework. This framework is composed by four modules (i.e. Distraction Module, Cognitive Module, Decision Module and Actuation/Adaptation Module) which interact to provide the ADAS functionalities.

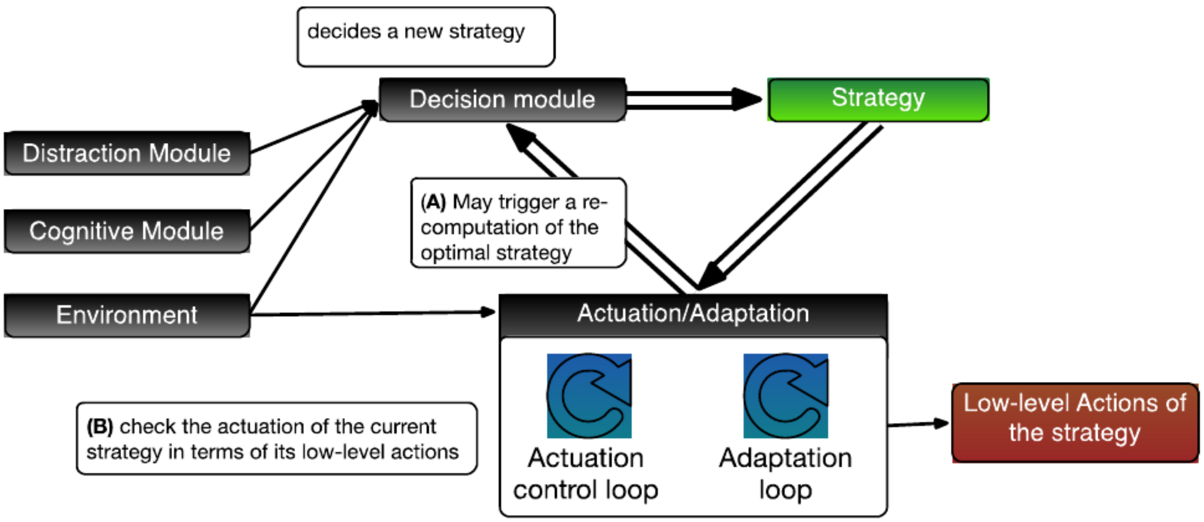


Figure 40: MDP co-pilot for the LCA AdCoS.

In detail, the Decision Module takes as input the human intention (by the Cognitive Module), the human distraction level (by the Distraction Module) and the environment data (i.e. sensors' information), and it computes the optimal driver's strategy in terms of a temporal sequence of pairs <state, macro-action>. Therefore, a macro-action (i.e. drive, accelerate, decelerate, brake, change lane, overtake) is associated with any possible predicted future state.

Then the Actuation/Adaptation Module takes as input such a computed strategy and decomposes any macro-action into a sequence of lower actions that will be actually implemented by the ADAS or human agent.

This module controls the correct temporal execution of these low level actions, and performs two different levels of adaptation:

1. An optimal strategy adaptation task, which can trigger a re-computation of the optimal strategy;
2. A macro-action adaptation task, which can trigger a reconfiguration of the macro-action decomposition without recomputing the optimal strategy.

More details of this framework and its instantiation on UC4/WP9 can be found in Deliverable 3.4.

The following logical scheme shows the Core of the LCA AdCoS, represented by the Co-pilot concept:

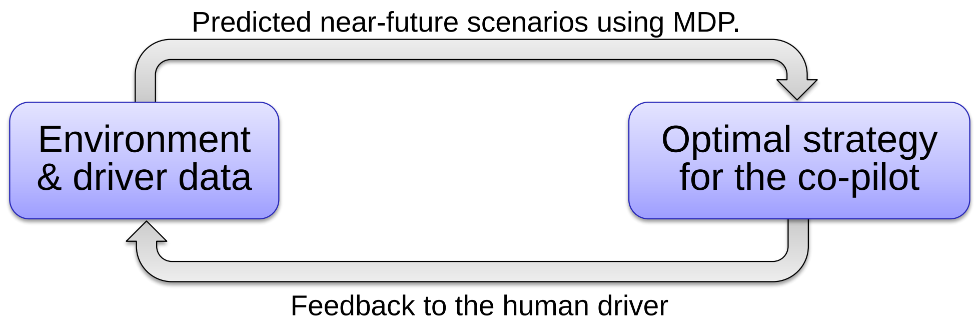


Figure 41: visual representation of the co-pilot concept for the LCA AdCoS.

The Co-pilot is an AdCoS module that computes the optimal driver strategy by predicting the (probabilistic/non-deterministic) future evolution(s) of the car system. It consists of a physical simulation of external objects, including the vehicle. The closed loop system updates in 1 tenth of a second (100ms).

The AdCoS architecture for the CRF test-vehicle is represented in the following figure:

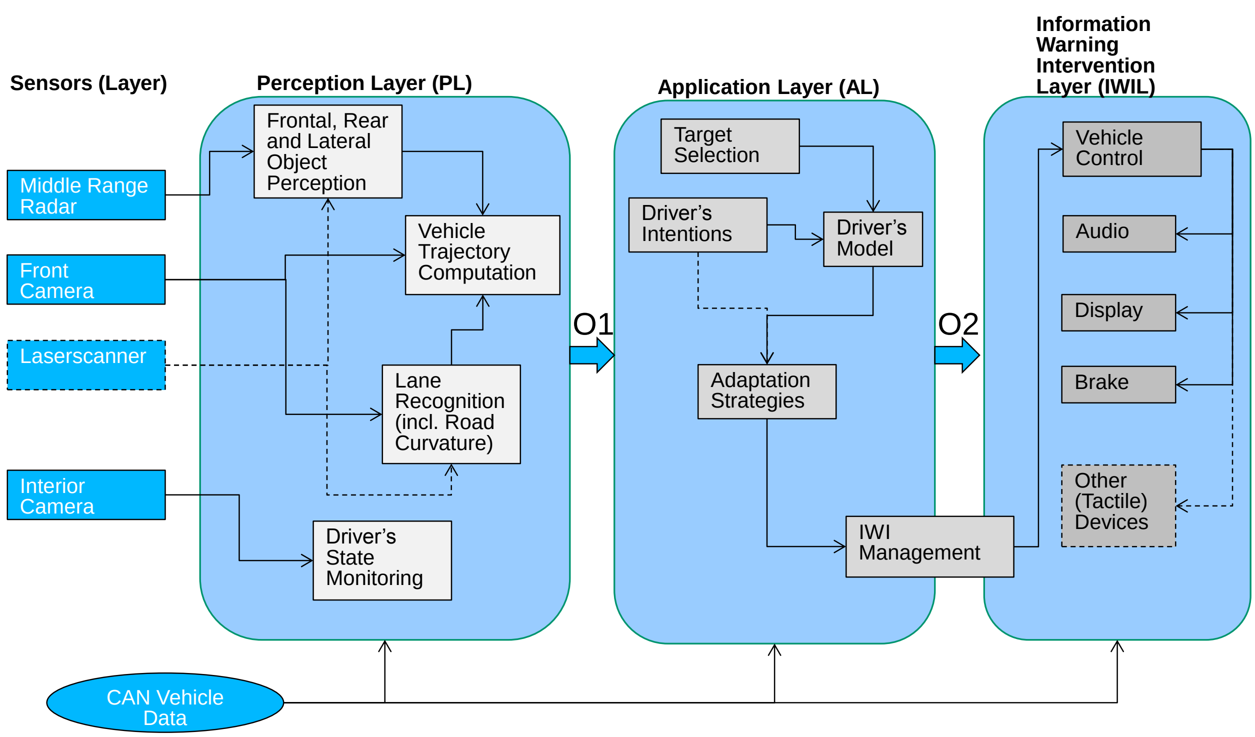


Figure 42: system architecture for the LCA AdCoS implemented in CRF demonstrator, where the main layers are highlighted.

As sketched in the figure, there are four main layers:

- **Sensors Layer** (SL) = it includes all physical sensors used to detect the internal (driver's behaviour and state) and external scenarios (the environment). The MRR will be substituted by at least two laser-scanner sensors to detect the vehicle surroundings during this project year and for this reason the related block has a dashed border.
- **Perception Layer** (PL) = it includes all the modules used to perceive and understand both scenarios and in particular the detection (and possibly classification) of the frontal, rear and lateral objects (All-around Object Perception, AOP module), as well as the recognition of the lane/road curvature and the related position of the ego-vehicle (Lane Recognition, LR module). These modules are used for the reconstruction of the environment, including the EV. This layer comprehends also the Vehicle Trajectory Computation (VTC) module, which is actually more related to the planning phase in a corresponding cognitive scheme (we have put here this block for convenience and sake of readability). In addition, the PL layer includes also the Driver State Monitoring (DSM) module, where the visual distraction classifier is implemented.
- **Application Layer** (AL) = it includes all modules used for the applicative part of the system, considering both aspects of interpretation and planning. In particular, the Target Selection (TSE) module receives as inputs the list of the obstacles and the EV path prediction, in order to provide as output the object(s) of interest (OOI), that is the risky obstacle on the EV trajectory. The Driver's Intention (DI) module aims at predicting the intention of the driver, that is to classify and recognize in advance the vehicle manoeuvre. Inside the HOLIDES project, the following actions will be considered: lane-change (LC), car-following (CF), lane-keeping or free-ride (LK or FR) and overtaking (OV), which can be regarded as a combination of more simple manoeuvres (see Section 3.2 for details). Thus the output of this module will be a probability for each of these manoeuvres. Then the Driver Model (DM) module is the core of the system, since it represents the implementation of the aforementioned co-pilot. The output is represented by the optimal manoeuvre that the system suggests to the driver. Then, the Adaptation Strategy (AS) module takes as inputs this information, the driver's intention and possibly the

EV trajectory to provide the final "suggestion" (warning, advice, or information) to the driver.

- **Information Warning and Intervention Layer** (IWIL) = it includes the modules for the management of the driver's HMI. The first one is the IWI Management (IWIM) module which is borderline between IWIL and AL: its goal is to "translate" the best strategy identified by the AS module into the correct HMI channel, managing also the priority and the possible conflicts if more than one functionalities are present. The Vehicle Control (VC) module aims at providing the control strategies for the actuators (mainly the braking system, but also the steering and the accelerator systems). Since in the CRF AdCoS direct intervention is not foreseen, the output of the IWIM module is directly sent to some HMI channels, such as the display, audio and tactile devices (see section 3.3 for details on the preliminary HMI concepts). With regard to the cognitive architecture, the IWI layer has its correspondence in the action category.

With reference to architectural scheme of figure 42, there is also the **CAN vehicle dat**a block, which provides the vehicle CAN data to all the other layers, as necessary. The next section describes the complete list of messages.

3.5.2 System Specifications for the Adapted Assistance AdCoS

The inputs, information flow and outputs between components are illustrated in the following figure, which is a more compact representation of the LCA system architecture:

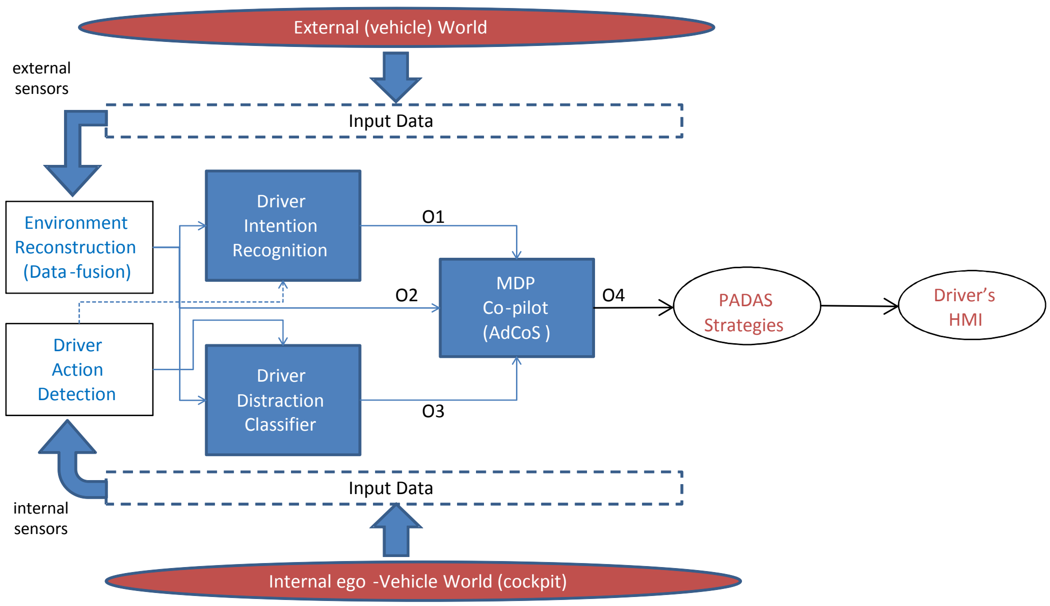


Figure 43: another representation architectural scheme to show input, information flow and output.

In figure 43, three main modules are highlighted: the driver intention recognition, the driver distraction classifier and the MDP Co-Pilot. The first two modules represent the adaptation strategies of the AdCoS, while the third module is the core (optimal strategy) of system.

The complete list of signals follows:

- Data from the external camera:
  - Yaw angle [deg]
  - Lane width [m]
  - Lane detection distance [m]
  - Lane curvature [m^{-1}]
  - Lane lateral distance [m]
- Data of the selected obstacle:
  - Object class [nb]
  - Object id [nb]
  - Obstacle speed (component Ox) [m/s]
  - Obstacle speed (component Oy) [m/s]
  - Coordinate X (longitudinal distance) [m]
  - Coordinate Y (lateral distance) [m]
- Data from the internal camera
  - Head position [deg]
  - Head orientation [deg]

  - Visual Time Sharing Distraction (VTSD) [s]
  - Visual Distraction Detection (VDD) [nb]
- Data of vehicle
  - Velocity [m/s]
  - Steering angle [deg]
  - Yaw-rate [deg/s]
  - Direction indicator left [nb]
  - Direction indicator right [nb]

Finally, we give some considerations about the operative scenarios of the LCA system. The ego-vehicle speed should be in the range (v_{min} – v_{max}), whose values have yet to be identified; a preliminary hypothesis is v_{min} = 5 km/h, v_{max} = 150 km/h. In detail, the minimum operative speed can be a modifiable parameter in a given range. The lower limit of the speed range can be set in order to prevent misuse and/or activations in specific scenarios (e.g. parking situations), while the upper limit is set depending on the sensors' performance.
For the operative road type, it is inter-urban roads and – above all – motorways, as well as straight and curved roads with road curvatures ≥125m.
For what concerns the environmental conditions, all weather conditions should be taken into account, even if heavy rain and snow could reduce the performance of the system, due to the conditions of the road surface.

It is worth noting that the driver shall be made aware of the system's operative scenario limitations.

3.5.3 System Architecture and Specifications for the Adapted Automation AdCoS

The system architecture is illustrated in figure 44. The artificial intelligence for the automated driving is implemented within the machine agent. The driving process can be broken down into four stages, in the same way as a human driver performs the driving: (1) the perception of the traffic environment around the host vehicle in real-time, (2) the interpretation and assessment of the current traffic situation, (3) the planning of appropriate maneuvers and actions and (4) the action to control the vehicle and guide it safely along the planned trajectory.

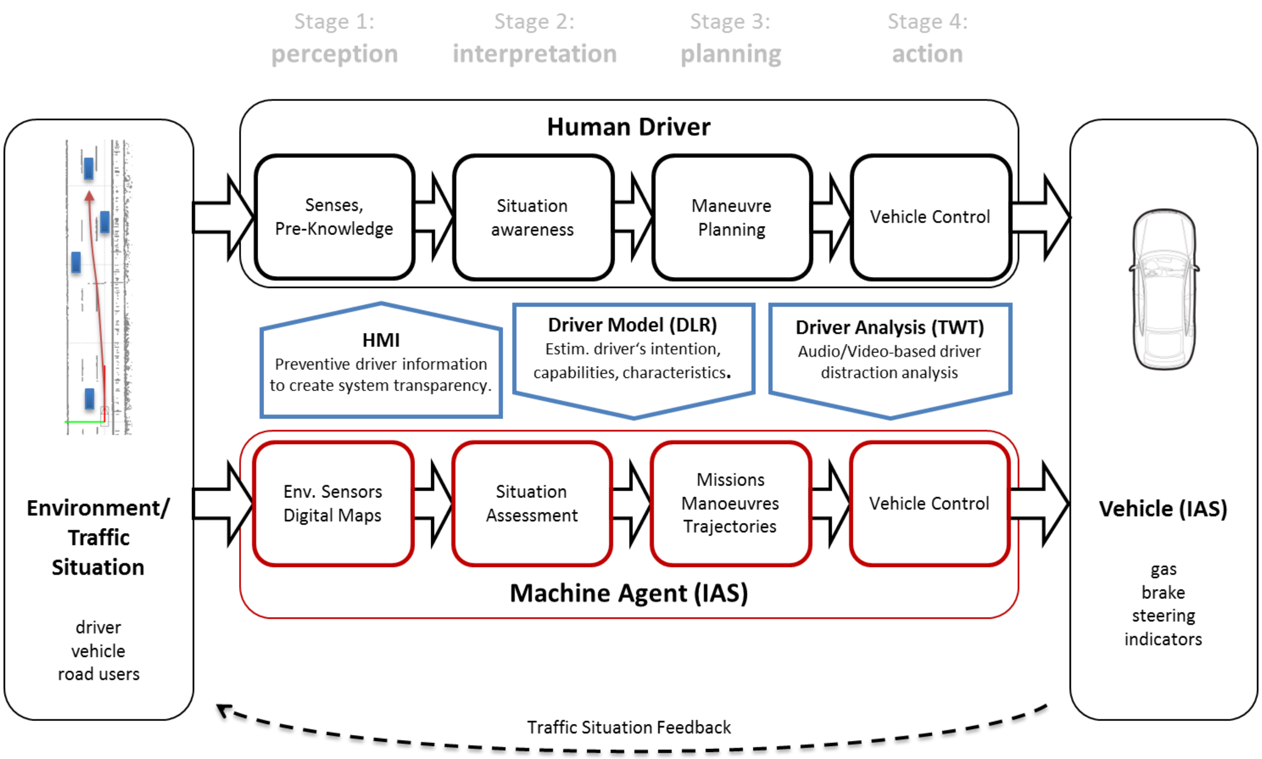


Figure 44: system architecture of the IAS AdCoS.

Three dedicated modules close the loop between the human driver and the machine agent and allow for interaction between them:

1. The advanced HMI provides situation dependent information to keep the driver informed about the interpretation of the traffic situation and planned manoeuvres. This allows the driver to take action and confirm or reject the planned actions of the artificial intelligence. A main challenge in the development of a suitable HMI for automated driving is selecting appropriate communication channels, as the driver might be distracted from the task of driving during automatic driving.
2. The driver model estimates the driving characteristics and intentions from manual control inputs of the human driver and provides these to the artificial intelligence. Manoeuvres are selected accordingly and the driving style is adapted to the individual characteristics of the driver. The focus of development is on the adaptation of the automated vehicle according to the manual driving style of the human driver.
3. The driver analysis estimates the distraction of the driver and adapts warning thresholds and take-over requests according to the situation awareness of the human. The distraction estimation is based on inner-

vehicular audio and video analysis. Research in this area focuses on determining the distraction of the driver from inner vehicular sensors.

Key aspects of the automated driving systems are a semantic situation assessment and adaptive planning algorithms considering input from the driver model and the driver analysis.
The following figure shows four main aspects to achieve an improved human-machine interaction:

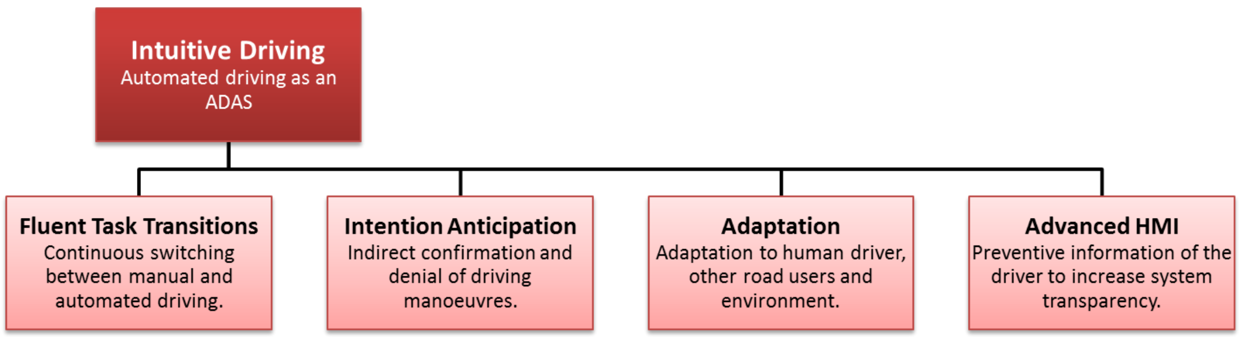


Figure 45: main aspects of intuitive driving approach.

Fluent Task Transitions. The driver can take over control at any time, either lateral or longitudinal control or both. Also, he can give control back to the artificial intelligence by just releasing the pedals and the steering at any time while the system is active. The driver uses the traditional control inputs – pedals, steering and indicators – to give feedback to the automated system to either confirm or override planned manoeuvres. Desired speed and inter-vehicular spacing can be adjusted via these controls.
Intention Anticipation. From the manual control inputs the automated vehicle anticipates the driver's intention to take different manoeuvres. Together with the fluent task transition this is the basis for a convenient interaction of the vehicle and allows the driver to cooperate on planned manoeuvres by the artificial intelligence.
Adaptation. Two types of adaptation are distinguished in this AdCoS: internal and external adaptation.
Internal Adaptation. The behaviour and driving style of the automated system adapts to the individual characteristics, capabilities and the awareness of the human driver. The adaptation is based on the input from the driver model and the driver distraction estimation. Inter-vehicular spacing, as well as accelerations, are adapted to the individual characteristics

of the human driver. Warning thresholds and timings for overtake requests are adapted according to the estimated situation awareness of the human driver in order to minimize the number of false or unnecessary warnings.
External Adaptation: The machine agent will adapt its behaviour according to the flow of other road users. It is characterized by a risk preventive behaviour taking into account different situation prediction hypotheses.
Advanced HMI. An HMI keeps the driver informed about the interpretation of the current traffic situation and the planned manoeuvres of the artificial intelligence. Information is presented to the driver depending on the current traffic situation and the awareness of the driver.

The AdCoS developed by IAS will be used as a demonstrator showing the previously described functionality. It is not planned to be used as an evaluation platform, since repeatability cannot be achieved.

3.5.4 Architecture & Specification of the Virtual AdCoS based on MOVIDA

As previously presented, the virtual AdCoS based on MOVIDA developed by IFS is an integrative co-piloting system supervising several simulated ADAS (e.g. Forward Collision Warning [FCW], a Lane Change Assistant [LCA], an Over-Taking Assistant [OTA] and Full Automation devices [FA]), to be managed in an adaptive and cooperative way by MOVIDA module, according to the drivers' visual distraction states and to the situational risks assessment (Figure 46).

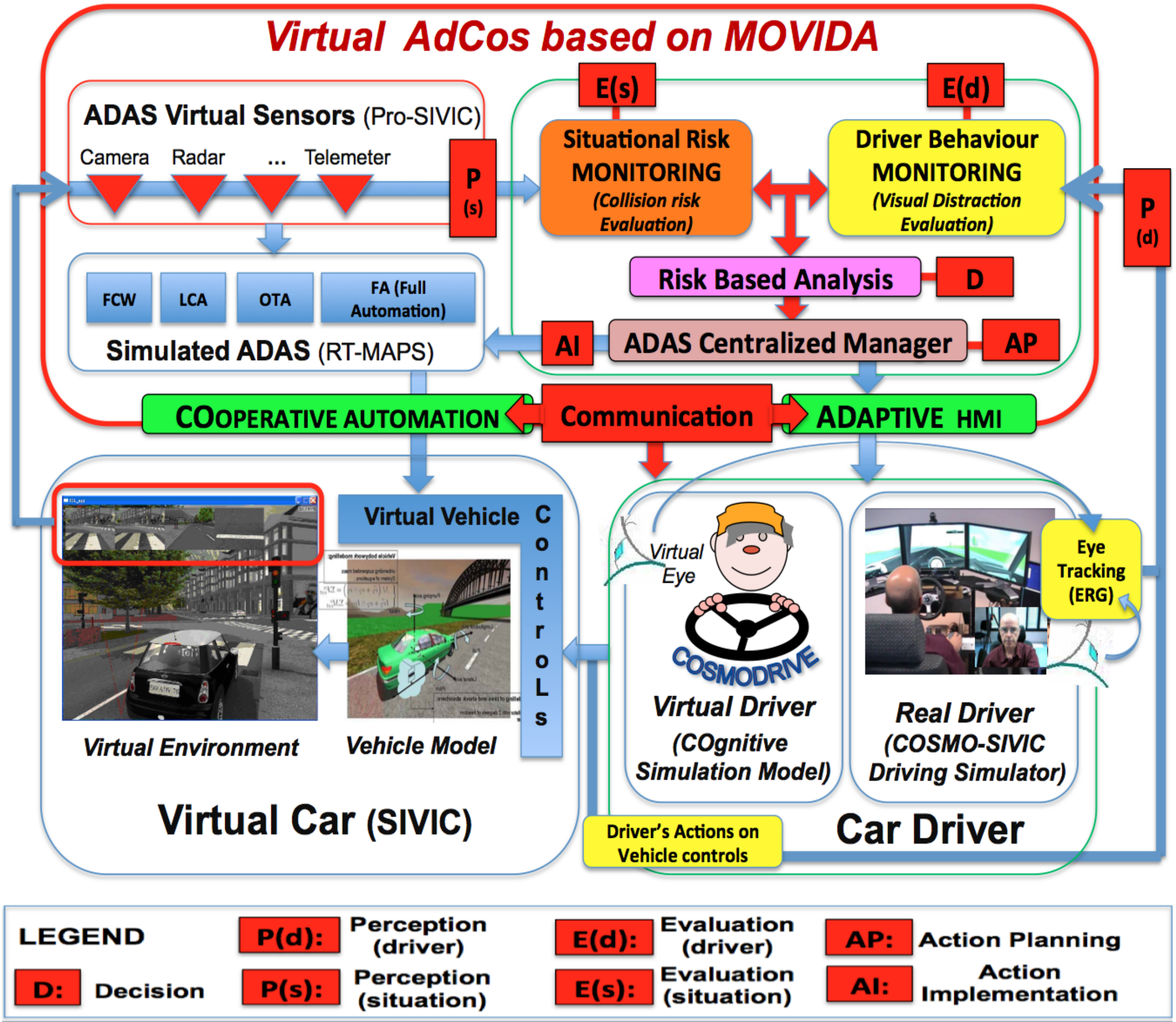


Figure 47: Virtual AdCoS based on MOVIDA (including P-E-D-AP-AI-C cycle)

Regarding the **P**erception-**E**valuation–**D**ecision-**A**ction **P**lanning-**A**ction **I**mplementation-**C**ommunication cycle defined in WP3 (D.3.4) The "Ad" and the "CoS" parts of the MOVIDA-AdCoS are respectively supported by a set of specific sub-components and/or modules of this AdCoS.

The "Ad" part includes the 3 **P**erception-**E**valuation–**D**ecision processes, respectively presented as P(s) & P(d), E(s) & E(d), and D red squares in Figure 44:

- **Perception** (P[s] & P[d]): correspond to the inputs of the AdCoS. **Two types of Perception processes** are combined in MOVIDA-AdCoS.

From one side, the Perception of the driving situation (P[s]) is supported by a set of virtual sensors simulated with Pro-SIVIC software. From the other side, the Perception of the Driver's activity (P[d]) is based on the driver's visual scanning (supported by ERG and/or Tobii eye tracking systems) and driving action analysis (implemented on the vehicle pedals and the steering wheel) collected on the virtual SIVIC car (simulated with Pro-SIVIC).

- **Evaluation** (E[s]) & E[d]): the evaluation process in MOVIDA-AdCoS is supported by **two complementary Monitoring Functions** (modelling as a set of state-transition diagrams). From P[d] data, drivers' behaviours monitoring algorithms (E[d])monitor the drivers in order to assess their visual distraction state and the adequacy of their driving behaviour in the current driving context. From the road environment, as perceived by car sensors (P[s]), situational risk monitoring algorithms will evaluate the traffic situation criticality in terms of collision risk E[s] (e.g. collision risk with a car ahead or with a car in rear/lateral position in the case of a lane change manoeuvre).

- **Decision** (D): Then, **Risk-Based Analysis** algorithms (implemented as a set of decision rules) will combine the assessment provided by the two preceding Monitoring Functions, in order to jointly assess the adequacy of the driver's behaviours and/or the visual distraction risk according to the traffic situation characteristics, for providing an overall measure of the criticality of the driving situation as a whole.

The "CoS" part of includes the 3 "**A**ction **P**lanning" – "**A**ction **I**mplementation" – "**C**ommunication" processes, respectively presented as AP, AI and Communication red squares in Figure 44:

- **Action Planning** (AP): From the diagnoses provided by the Monitoring Functions and the Decision Module, a **Centralized Manager of ADAS** will determine which kind of ADAS, integrated in the MOVIDA-AdCoS (like collision avoidance support or lane change aids), can support the drivers to manage the risk of accidents due to driver's visual distraction, and/or how to intervene to avoid the crash (Warning or Automation) in accordance with the criticality of the driving situation.

- **Action Implementation** (AI): Then from the analysis provided at the preceding level of MOVIDA-AdCoS, the centralized manager of ADAS will effectively activate the ADAS sub-system to avoid accidents and determine the best way to interact with the COSMODRIVE driver, ranging from delivery of warnings to taking control of the car.
- **Communication**: Finally, two core modules will manage interactions with the driver, in order to provide an adaptive and cooperative support specifically adapted to the current needs of the driver.

The Adaptive HMI will adapt HMI modalities of information delivery and warning signals in accordance with driver state (visual distraction) to alert the driver. The Cooperative Automation support (Partial or Full) will take the control of the car (Automatic Brake or Lane Change, for instance) in the case of behavioural errors (e.g. critical visual distraction or risky lane change manoeuvre implemented) or due to the situational collision risk with the other vehicles (rear or forward).

3.5.5 System Architecture for the Adaptive HMI

The Adaptive HMI AdCoS will be implemented in the TAK driving simulator (see Figure 48). There are three parts to be integrated:

- The distraction detection hardware and software
- The adaptive assistance systems
- The adaptive HMI.

All three parts will be integrated in the simulator. The integration of the sub-systems and their communication will be implemented with the help of RT-Maps.

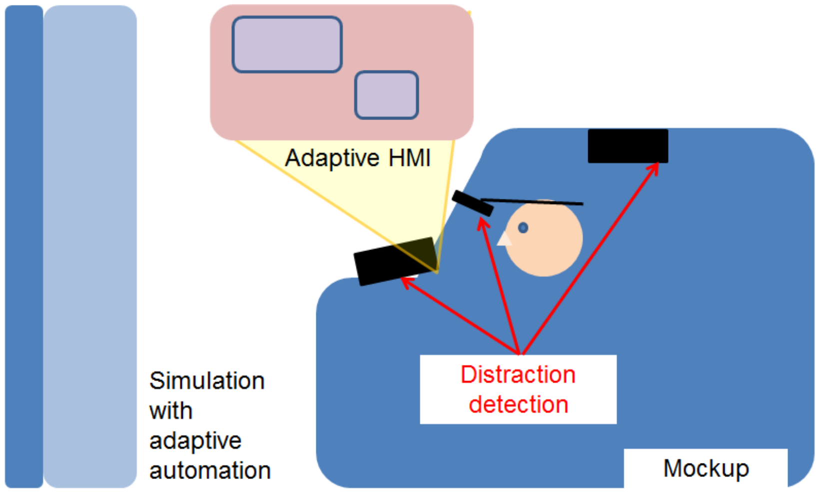


Figure 48: Draft of system architecture of the TAKATA Adaptive HMI AdCoS.

The distraction detection hard-and software will be a combination of the eye-tracker provided by ERGONEERS and the distraction detection system provided by TWT. The adaptive assistance system will again consist of three parts:

- A blind spot detection system (BSDS) with a warning functionality.
- A longitudinal ADAS (with similar functionality as a standard ACC/Distronic).
- An automatic lane change system, which is based on and is an extension of an LKAS (lane keeping assistance system.

The HMI system consists of the rear view mirrors, in which information from the BSDS will be displayed. These mirrors are implemented as TFT-monitors that are integrated into the simulation software. Furthermore, it includes the instrument cluster and a display for the entertainment system in the center stack.

3.5.6 System Specifications for the Adaptive HMI

The Adaptive HMI AdCoS consists of several parts as described above. These parts are shown in Figure 49.

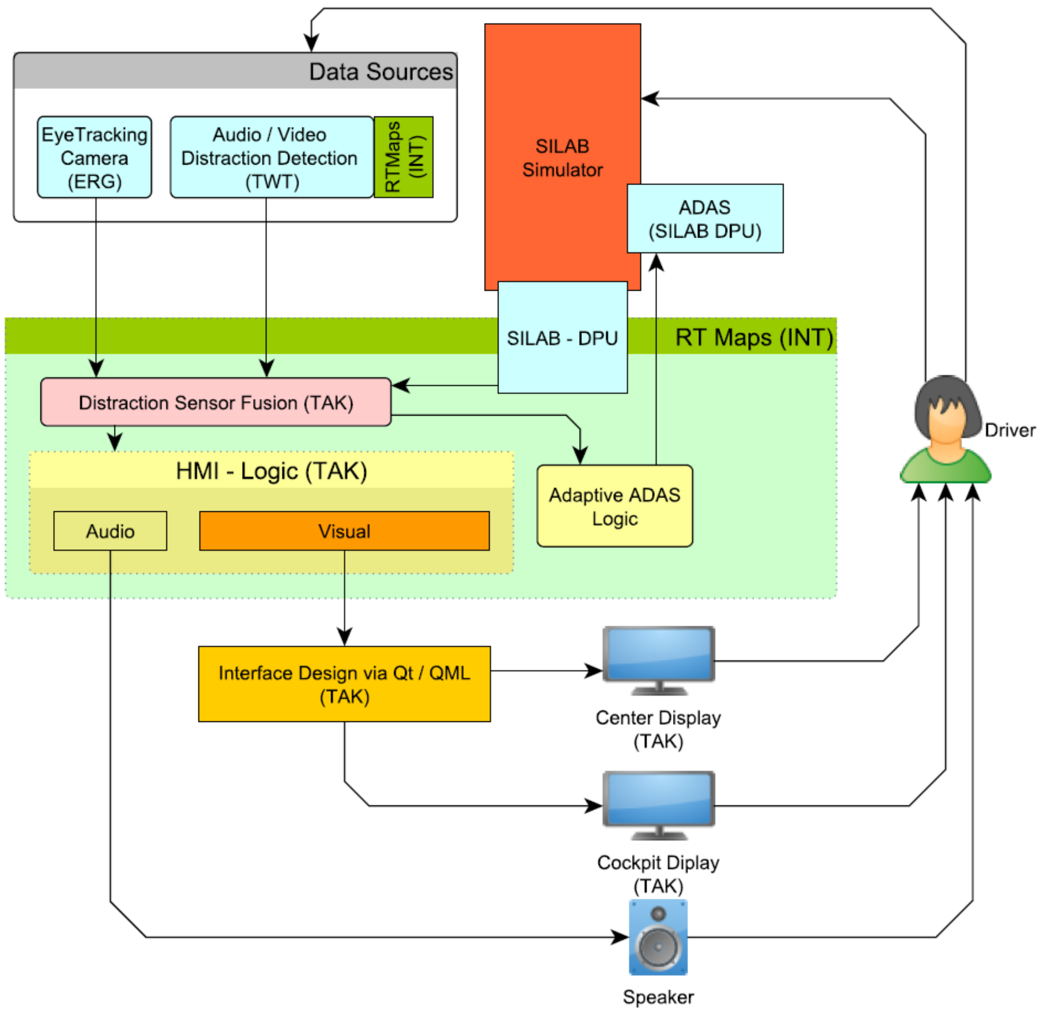


Figure 49: Draft of system architecture of the TAKATA Adaptive HMI AdCoS.

The TAKATA driving simulator consists of a mock-up, an operator workplace with simulation hard- and software and the screening system. The mock-up is the driver's quarter chassis and bodywork of a 3 series BMW which is mounted on a metal frame. The instrument cluster, the entertainment system in the centre stack and the mirrors are all modelled by TFT-displays. The simulation software used is SILAB which is a user friendly solution developed by WIVW that offers the possibility for own development. Mock-up, simulation and additional hardware and software can be integrated via several connection interfaces. The screening system is a three channel system offering a 130 degrees view with three additional displays representing the two outer mirrors and the inner rear-view mirror.

3.6 Monitoring Systems and Use Case CRF

Monitoring Systems shows in real time the state of the car sensors, the messages received by the driver in the HMI console, covering the requirements proposed by CRF Test Vehicle Use Case, displaying in a monitor screen the features described below:

- Grid of messages in real-time showing the requirements covered in the CRF vehicle Test Use Case. For instance: risk of collision in straight roads, lane change inhibition, etc.
- State of the car sensors: radar detection, car sensors (hand brake, brakes, acceleration, speed, ….).
- Messages received from Assistants, Lane Change Assistance, Overtaking Assistance.
- Webcam streaming in real time.

The aim is to use Data Distribution Services (DDS) in real-time to connect RTMaps and the Monitoring System. DDS functions could be created in Prysmtech cloud, via VORTEX Data, sharing a Platform internet of things application:

- Built on open standards to enable application portability, interoperability and component re-use.
- Device to Device, Device to Cloud and Device to Device Cloud-based data sharing.
- Dynamic discovery for applications publishing and subscribing to data.
- Plug-and-play for applications enabling systems to evolve more easily.

DDS Functions allow to subscribe the MS as a client in a publish-subscribe pattern, and RT-MAPS as a publish-subscribe server receiving information in a publish-subscribe pattern methodology, as depicted in the following figure:

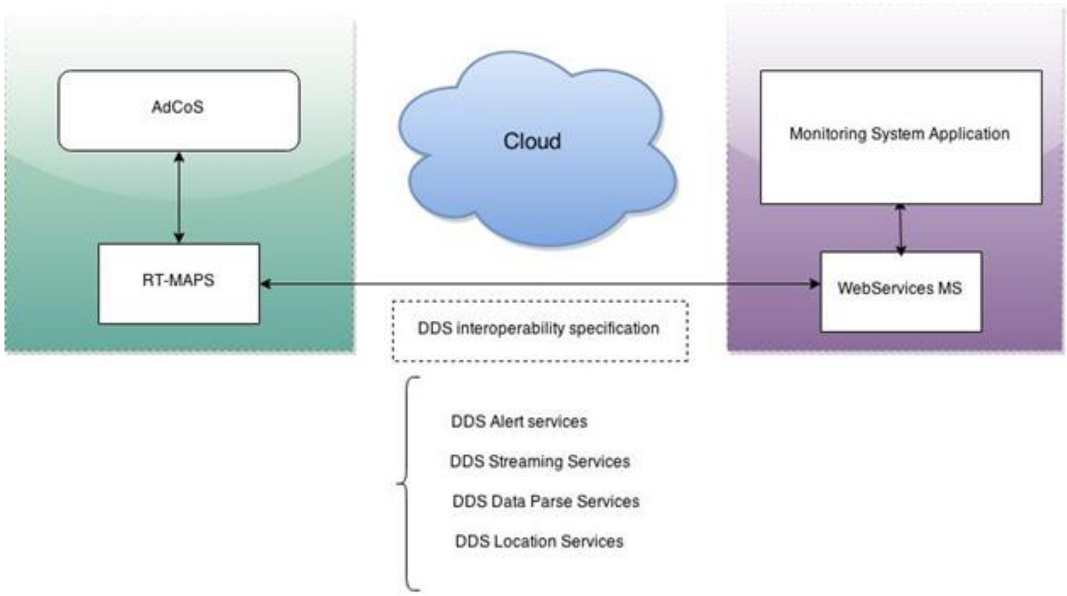


Figure 50: schema for DDS function.

DDS services should give the possibility to transmit data classified by types of data, transforming data from different systems, and adding specific functionalities: location, video streaming, and others.

It is worth noting that the Monitoring System will be connected with RT-Maps, in order to monitor all state of sensors and different messages from AdCoS; if the specific uses-case does not use RT-Maps, the connection between them should be created.

The system architecture is specified and shown in the picture below:

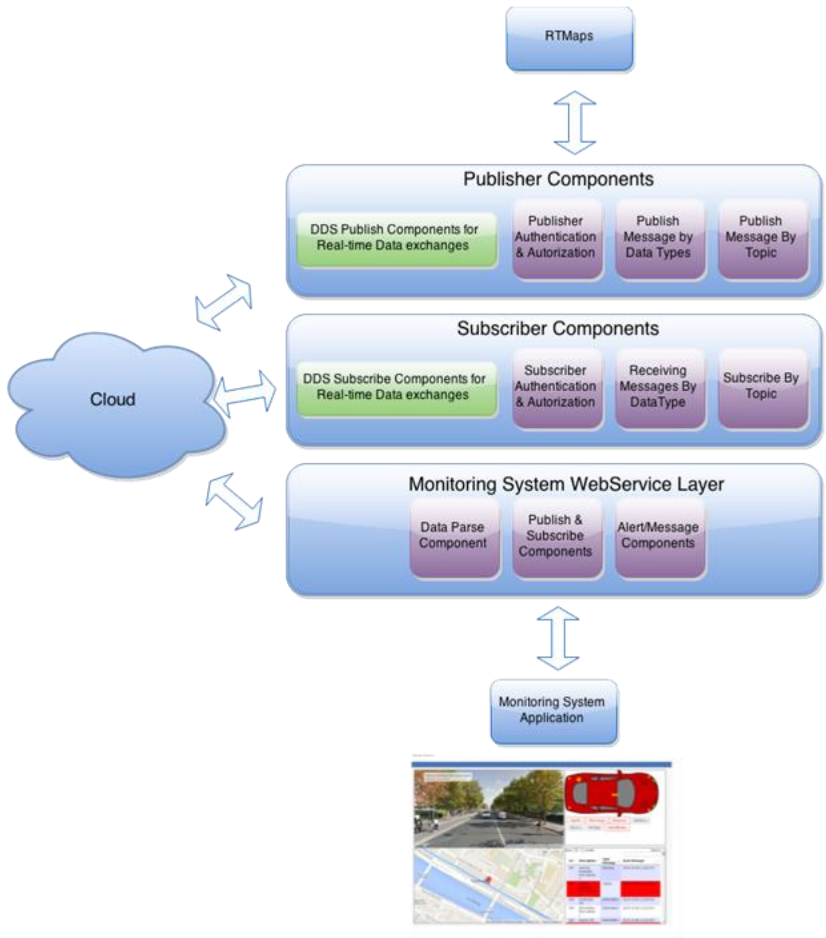


Figure 51: Monitoring System architecture.

The connection between RT-Maps and the Monitoring System is via Data Distribution Service publishing and subscribing services by topics as a Publish Subscribe pattern. Each device (webcam, sensors) should be connected between each publication (e.g. a publication of one sensor value) and the subscriber (e.g. a logic sensor defined in the DDS subscriber component), sending values (characters, streaming video, etc.) in real-time.

The Monitoring System Web-Service Layer contains services to manage this data, and is linked with Monitoring System Application, which shows the information in a synchronized mode.

4 Feedbacks from WP1-5

The deliverable D9.1 extensively described Automotive domain requirements delivered to WP1-5, including the ones specifically addressing the HF-RTP. All these requirements were thus associated with stages of the development life-cycle and described how the WP1-5 results were expected to reduce development cost and times-cycles, as well as to improve system performances. With this respect, Automotive AdCoS requirements and HF-RTP requirements represented an important input for both the definition and, subsequently, tailoring to Automotive purposes of the HF-RTP itself.

In the automotive domain, a total of 166 requirements have been collected, which are based on the target scenario and the related use cases. Due to the dynamic nature of the development process in research projects, these requirements must be checked regularly against potential new project states. This checking might result in requirements having to be updated. In the case of the automotive application WP, most requirements remained the same. For a more detailed description, please see the related EXCEL table (where the column 'Status' refers to current state of a requirement with a brief justification) and the summary provided in section 3.4.

WP1-5 partners have worked since then at the revision, refinement and update of the WP9 requirements which have now been incorporated in the D1.4 and D2.4 deliverables as tables in the Annexes including the list of refined and updated requirements (for the MTTs and the platform) from WP6-9 application domains. By now, these exhaustive tables represent the most relevant feedback from WP1-5 thus constituting a major input for further HoliDes cycles devoted to the development of Automotive AdCoS applications.

5 Conclusions

As stated in the conclusions of the other deliverables DX.3[4], during the first year of the project, WP9 partners have identified, refined and updated a number of use-cases (based on selected target-scenarios), functional / non-functional requirements and system specifications of AdCoS applications implemented in AUT domain, that is (examples from the four main implementations):

- Adapted Assistance
- Adapted Assistance on Virtual HCD Platform
- Adapted Automation
- Adaptive HMI

This report describes the outcome of such activities and tasks, with further reference to the underlying links with the development of the HF-RTP and the activities carried out in the scientific WPs2-5, in particular with reference to the MTTs developed here.

The collection and analysis of the operational definitions, the HMIs and the tools used for the applications development, as well as their requirements and specifications, has highlighted some constituent relations which are consistent with the development process in the first and second cycles of HoliDes.

In fact, the requirements of the AdCoS identified and collected in D9.1 have provided a solid basis, on which the AdCoS concepts, which play a key role in determining (for each use case) the relative specifications and architecture, have been developed.
Furthermore, the analysis of requirements for WP9 AdCoS has also enabled a better understanding of which methods, tools and techniques were relevant to account for human factors in WP9 applications (true for all the related AdCoS), thus constituting an important input for WP1 (the HF-RTP), as well as the other scientific WPs2-5 (the MTTs).
Finally, the horizontal WPs (WPs1-5), in turn, have provided feedback concerning the main WP9 requirements that have supported to a certain extent the definition of the functional specifications (including system architecture) and preliminary HMI concepts. This interaction between

[4] Where X = 6, 7, 8, 9.

application domains (WPs6-9) and scientific activities (WPs1-5) has granted an effective iterative process which constitutes an added value of HoliDes project.

References

[Aij85] Ajzen, I.: From intentions to actions: A theory of planned behavior. In J. Kuhl and J. Beckman (Eds.), Action-control: From cognition to behaviour, pp. 11-39. Heidelberg, Germany: Springer. (1985)

[Aij02] Ajzen, I.: Perceived Behavioral Control, Self-Efficacy, Locus of Control, and the Theory of Planned Behavior. Journal of Applied Social Psychology, 32, pp. 665-683. (2002)

[Bör13] Börger, J.: Fahrerintentionserkennung und Kursprädiktion mit erweiterten Maschinellen Lernverfahren. Dissertation, Universität Ulm. (2013)

[FA75] Fishbein, M. and Ajzen, I.: Belief, Attitude, Intention, and Behavior: An Introduction to Theory and Research. Reading, MA: Addison-Wesley. (1975)

[GJ97] Ghahramani, Z. and Jordan, M. I.: Factorial Hidden Markov Models. Machine Learning, p.1-31 (1997)

[JG96] Jordan, M. I., Ghahramani, Z., and Saul, I. K..: Hidden Markov Decision Trees. In: Proceedings of the 9th Conference on Advances in Neural Information Processing Systems, pp. 01-507. (1996)

[Kob11] Kobiela, F.: Fahrerintentionserkennung für autonome Notbremssysteme. Springer. (2011)

[KF09] Koller, D. and Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT Press. (2009)

[LP97] Liu. A. and Pentland, A.: Towards real-time recognition of driver intentions. In Intelligent Transportation System, 1997. ITSC'97, pp. 236-241. (1997)

[OP00] Oliver, N. and Pentland, A.: Driver behavior recognition and prediction in a SmartCar. In *AeroSense 2000*, pp. 280-290, International Society for Optics and Photonics. (2000)

[NHT13] National Highway Traffic Safety Administration (NHTSA), "Preliminary Statement of Policy Concerning Automated Vehicles", Washington, USA, May 2013

[RAAK13] Rausch, Aeberhard, Ardelt, Kämpchen, "Autonomes Fahren auf der Autobahn – eine Potentialstudie für zukünftige Fahrerassistenzsysteme", Tagung Fahrerassistenz, TU München, München, Germany, 2013

[Bm13] Myra Blanco, "Human Factors Evaluation of Level 2 and Level 3 Automated Driving Concepts", Virginia Tech, Automated Vehicle Systems Group Research, 2013