

Sensor Platform and Models including V&V results from 2nd cycle

Project Number:	690705
Classification Deliverable No.:	Public
Work Package(s): Document Version: Issue Date: Document Timescale: Start of the Document: Final version due:	WP2 Vs. 0.1 31.12.2017. Project Start Date: September 1, 2016 Month 14 Month 16
Compiled by:	A. Knapp, BIT
Authors:	D. Käthner, DLR A. Giralt, CAF M. Eilers, HMT S. Feuerstack, OFF L. Weber, OFF P. P. Fouopi, DLR F. Tango, CRF M. Graf, ULM E. Landini, REL A. Knapp, BIT Z. Jakó, BIT
Technical Approval:	Fabio Tango, CRF
Issue Authorisation:	Andreas Lüdtke, OFF

© All rights reserved by AutoMate consortium

This document is supplied by the specific AutoMate work package quoted above on the express condition that it is treated as confidential to those specifically mentioned on the distribution list. No use may be made thereof other than expressly authorised by the AutoMate Project Board.

DISTRIBUTION LIST		
Copy type ¹	Company and Location	Recipient
T	AutoMate Consortium	all AutoMate Partners

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

RECORD OF REVISION

Date	Status Description	Author
04.10.2017	Preparation of initial document	A. Knapp (BIT)
23.11.2017	Document structure update	E. Landini (REL)
30.11.2017	Introduction	E. Landini (REL)
04.12.2017	V2x communication	A. Knapp (BIT)
07.12.2017	Semantic enrichment of the situation model	P. P. Fouopi (DLR)
15.12.2017	Driver Intention Recognition Predicting the future evolution of the traffic scene	M. Eilers, (HMT)
18.12.2017	Driver monitoring system with driver state model for distraction and drowsiness	A. Giralt, (CAF)
19.12.2017	Pre-final version for review	A. Knapp (BIT)
21.12.2017	Updates based on reviewers comments	A. Giralt, (CAF) M. Eilers, (HMT) A. Knapp (BIT)
22.12.2017	Final version to submit	A. Knapp (BIT)

Table of Contents

1	Introduction	6
2	How the WP2 enablers contribute to the implementation.....	9
3	Status of WP2 enablers in cycle 2	14
3.1	E1.1 – Driver monitoring system with driver state model for distraction and drowsiness	15
3.1.1	Scenario and uses case where E1.1 is relevant.....	15
3.1.2	Implementation.....	15
3.2	E1.2 – V2x communication	24
3.2.1	Scenario and uses case where E1.2 is relevant.....	24
3.2.2	Implementation.....	25
3.3	E2.1 – Driver Intention Recognition	27
3.3.1	Scenario and uses case where E2.1 is relevant.....	27
3.3.2	Concept.....	28
3.3.3	Implementation.....	31
3.4	E3.1 – Situation and vehicle model	42
3.4.1	Scenario and uses case where E3.1 is relevant.....	42
3.4.2	Implementation.....	42
3.5	E3.2 – Driving Task model	55
3.5.1	Improvements	55
4	Validation of enablers	59
4.1	E1.1 – Driver monitoring system with driver state model for distraction and drowsiness	60
4.1.1	Drowsiness experiment	60
4.1.2	Visual Attention/Distracton experiment.....	66
4.2	E1.2 – V2x communication	70

4.3	E2.1 – Driver Intention Recognition	75
4.3.1	Experiments for data gathering	75
4.3.2	Data Preparation	80
4.3.3	Validation process	81
4.3.4	Datasets used	83
4.3.5	Results.....	83
4.4	E3.1 – Situation and vehicle model	87
4.4.1	Semantic enrichment of the situation model	87
4.4.2	Predicting the future evolution of the traffic scene	89
5	Conclusions and Outlook.....	96

1 Introduction

The activities in the Automate project have been organized in 3 cycles to guarantee that the maturity of the technologies developed in the project is iteratively increased while assessing that the progresses are consistent with the needs of the demonstrators and, in turn, with the overall concept and objectives of the project.

As shown in Figure 1, the first 2 cycles are focused on the development and technical validation of the components (i.e. the enablers) performed in WP2, WP3 and WP4. The experience acquired in the 1st cycle (lesson learnt) has been used at the beginning of the 2nd cycle to review the requirements and metrics for the design and development of the enablers and, as a consequence, to improve them.

At the end of the 2nd cycle, the enablers are planned to be integrated into the demonstrators in WP5, and the performances of the 1st version of the demonstrators are evaluated against their baseline in WP6.

In the 3rd cycle, WP2, WP3 and WP4 are fed with the results of this evaluation process to deliver the final version of the enablers. The 3rd cycle ends with the evaluation of the final version of the demonstrators.

This deliverable describes the current state of the enablers developed in WP3 in the first half of the 2nd cycle, as well as the experiments conducted and proposed to technically validate them according to the validation plan and the requirements and metrics defined in D3.4.

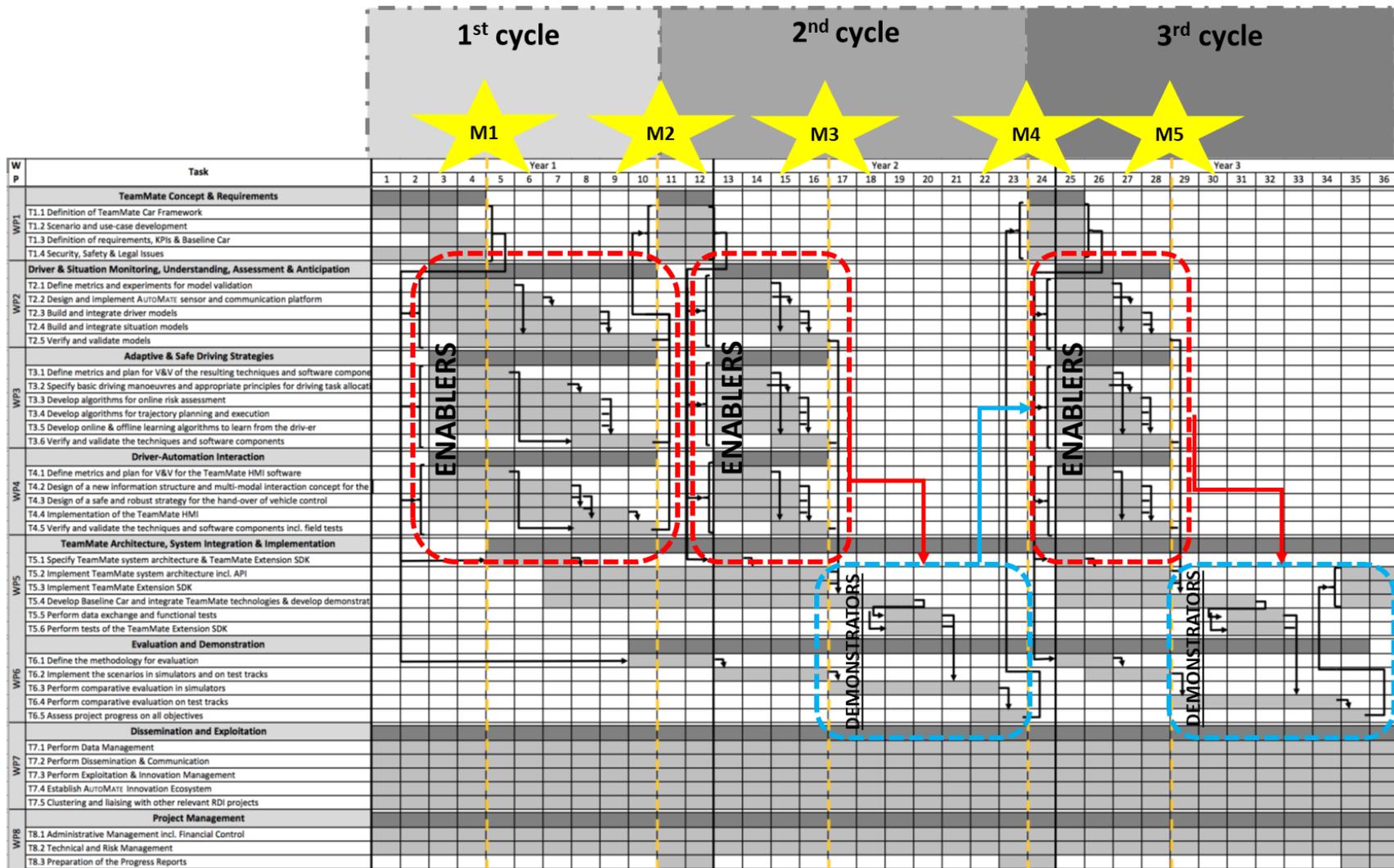


Figure 1: Project cycles, milestones and link between enablers (WP2, WP3 and WP4) and demonstrators (WP5 and WP6)



The development of all enablers follows the same process for WP2, WP3 and WP4. Therefore, the deliverable D2.4, D3.5 and D4.4 that describe the status of the development and validation of the enablers have been structured with the same chapters to reflect the common (parallel) process followed in WP2, WP3 and WP4 to deliver all enablers in time to be integrated into the demonstrators.

The document is structured as follows. After the introduction the general approach of the project regarding WP2 is described in Chapter 2. Then, the status of the enablers is presented in Chapter 3 including the improvements and latest developments of them. Next, Chapter 4 describes the validation of enablers along with validation methodologies and the results. Finally, Chapter 5 concludes the document.



2 How the WP2 enablers contribute to the implementation

The top-level objective of AutoMate is to develop, evaluate and demonstrate the “TeamMate Car” concept as a major enabler of highly automated vehicles. This concept consists of considering the driver and the automation as members of one team that understand and support each other in pursuing cooperatively the goal of driving safely, efficiently and comfortably from A to B.

As a consequence, in order to show how the enablers contribute to the implementation of this concept, it is important to briefly explain why the cooperation is needed, and how the human and the automation can support each other to create a safe, efficient and comfortable driving experience.

As shown in Figure 2, both the human and the automation have **limits** that can negatively affect the safety as well as the efficiency, the comfort, the trust and the acceptance of the autonomous driving.

For the human, the limits are often related to their driving performance: they are likely to affect the safety, and cause accidents. For the automation, the limits, mostly at perception and decision level, may affect the efficiency and the comfort of the trip, and then, in turn, the acceptance of the automation.

The AutoMate approach is based on the mutual complementarity between the driver and the automation: this support is achieved through the cooperation, between the team members.

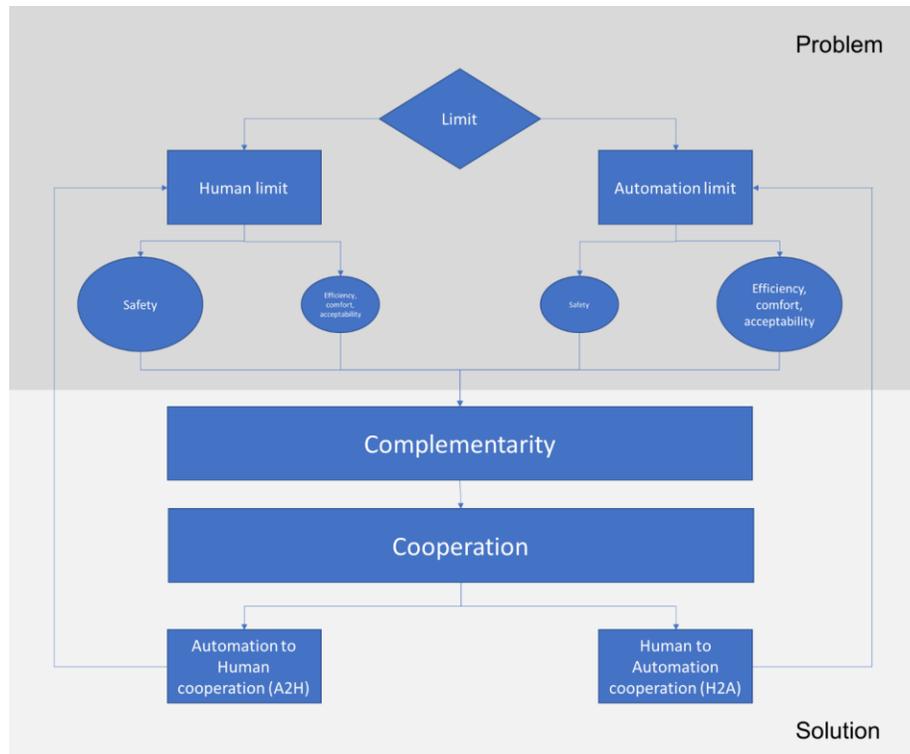


Figure 2: Schematic representation of the overall concept of the project.

While the Automation to Human Cooperation (A2H) is used to complement the human limits, the Human to Automation Cooperation (H2A) is implemented to allow the driver to support the automation to overcome its limits.

The complementarity between the driver and the automation is the conceptual solution to compensate the reciprocal limitations, while the cooperation is how the complementarity is implemented. Figure 3 shows how both the A2H and the H2A cooperation can be implemented in perception (state A and B) and in action (state C and D).

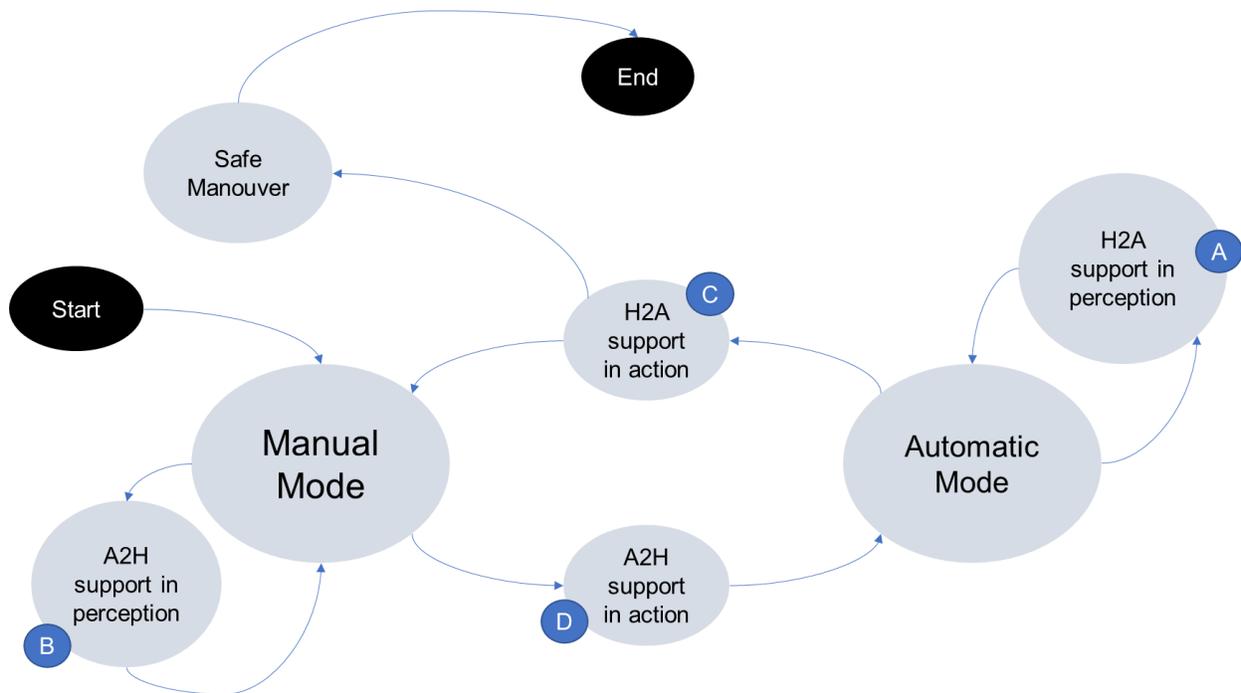


Figure 3: State machine that shows how the cooperation is implemented

The scenarios and use cases selected to demonstrate the relevance of each enabler are therefore representative and consistent with the direction of cooperation implemented by that enabler, as well as the modality of support (i.e. either in action or perception). Since the cooperation is implemented through the enablers developed in the project, Table 1 shows the role and relevance of each enabler in the cooperation.



Table 1: Role and relevance of the WP2 enablers for the cooperation

WP	ID	Enabler	Enabler Owner	Aim of the enabler	Direction of the support	
					Automation to Human	Human to Automation
WP2	Enabler 1: Sensor and communication platform					
	E1.1	Driver monitoring system with driver state model for distraction and drowsiness	CAF	Sensors and models for driver's visual distraction and drowsiness detection and classification	Enabler E1.1 is needed to implement a support in perception to complement the perception of the driver about the his/her state	
	E1.2	V2X communication	BIT	Allow the communication between the vehicle and everything.	Enabler E1.2 is needed to implement a support in perception to complement the perception of the driver about the environment	
	Enabler 2: Probabilistic Driver Modelling and Learning					
	E2.1	Driver intention recognition	OFF	Classify the current driver state, describe the interdependencies between the driver's state, type, behaviour and environment and predict the driver intention	Enabler E2.1 is needed to implement a support in perception to complement the perception of the driver about his/her state	
Enabler 3: Probabilistic Vehicle and Situation Modelling						



E3.1	Situation and vehicle model	DLR OFF	Estimate the dynamic vehicle and object state and position	Enabler E3.1 is needed to implement a support in perception to complement the perception of the driver about the situation and the vehicle	
E3.2	Driving Model task	DLR	Define the driver's tasks to understand the expected behaviour (Paper Enabler)	Enabler E3.2 is needed to implement a support in action along with E6.1 (Interaction Strategy) to provide the driver with an effective means to interact with the automation in case of need.	Enabler E3.2 is needed to implement a support in action along with E6.1 (Interaction Strategy) to provide the driver with an effective means to answer and give feedback to the automation.



3 Status of WP2 enablers in cycle 2

This section describes the detailed progress done during the 2nd cycle of the project. Modified assumptions and new approaches based on the knowledge acquired from the 1st cycle are written along with the improvements and current state of the developments. Furthermore, the testing methodologies are presented as well that served to validate the enablers. Finally, future plans and works for the 3rd cycle are highlighted.



3.1 E1.1 – Driver monitoring system with driver state model for distraction and drowsiness

3.1.1 Scenario and uses case where E1.1 is relevant

As shown in Table 1, Enabler E1.1 is needed to implement a support in perception from the automation to the human (A2H) to complement the perception of the driver about his/her state.

One of the use cases of MARTHA scenario has been revised to highlight and clarify the role of E1.1 to implement this cooperation.

Martha is driving in an extra-urban road in Manual Mode. She receives an incoming call: the car detects that she is distracted and this could lead to an unsafe behaviour. The TeamMate car offers a cooperation in action, suggesting a handover in order to shift to Automatic Mode. Martha accepts the suggestion and cedes the control.

3.1.2 Implementation

3.1.2.1 Driver monitoring system hardware

The driver monitoring system includes the following main parts (details are described in another project document [2] of task 2.2):

- one camera which operates in the Near Infrared spectral range placed on top of the steering wheel column observing the driver's face through the steering wheel;

² Driver Monitoring Sensor, AutoMate_WP2_Driver_State_Sensor_01.pdf



- a set of Near Infrared LEDs synchronized with the camera shutter which illuminate the driver's face;
- a LED controller box which provides the required power to the LEDs;
- a PC connected to the camera which runs the Driver Monitoring applications including a User Graphic Interface.

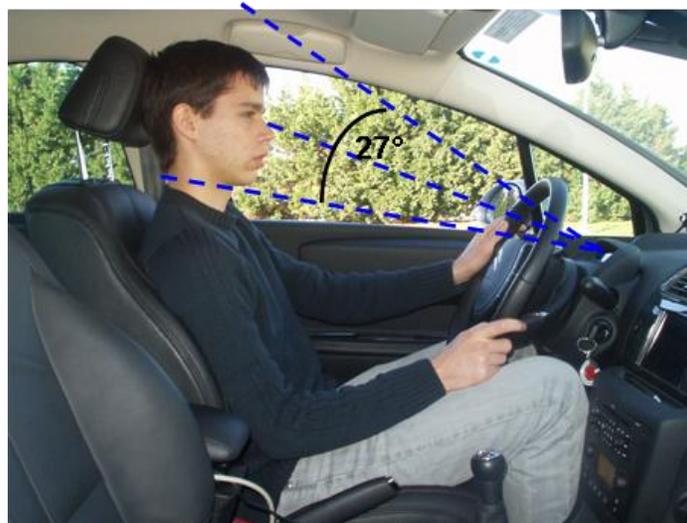


Figure 4: Camera integration and field of view

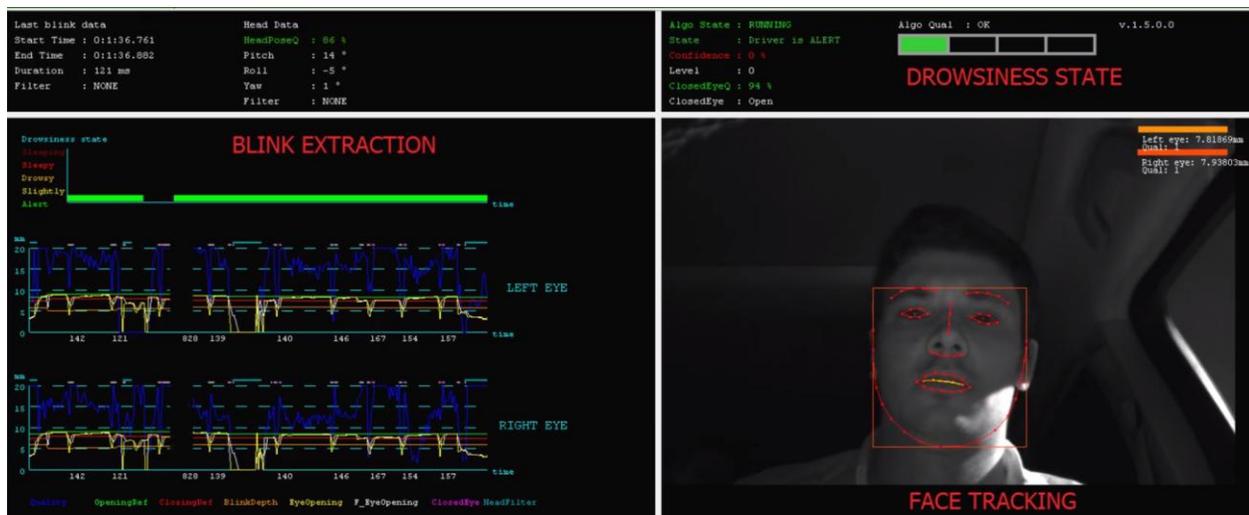


Figure 5: Display of the driver monitoring system



Within cycle 3 the driver state sensor hardware will be improved with a better performing camera and light units. The new HW includes an HDR, NIR 1.2 Mpx camera running at 30 fps and a set of distant 940nm NIR lights avoiding the bright spot on the glasses.

3.1.2.2 Drowsiness

Drowsiness is characterized by many physiological symptoms: increase of the blink duration, yawning, head leaning forward, reduced eyelid opening, and eye gaze staring etc. The development started in the 1st cycle focused on improving the eyelid/eye opening based model by using head specific movement.

The analysis of the head movements is used to reinforce the eye/eyelid drowsiness model when this one is available. It also provides a drowsiness model by its own when the eye/eyelid drowsiness model is not available. Such situation may occur when the driver wear sun glasses, when the reflections of the NIR lights occlude a too important part of the eyes, heavy make up on the eyes, out of range head inclination etc.

The development initially focused on head gaze staring considering the assumption that a drowsy driver performs less head movements than when he is alert. A fixation duration was defined on the basis of the drowsiness database. The duration was set to 10 seconds during which the driver head rotations do not exceed 2°. So we expect that the frequency of head fixations would increase with drowsiness.

The graphs in the figure below show on top the drowsiness level on the KSS scale as labelled by an expert, the bars on graph in the middle indicate a head fixation, the bottom graph shows the number of fixation per period of 5



minutes. One can note that only very high level of drowsiness are correlated with an increase number of head fixation.

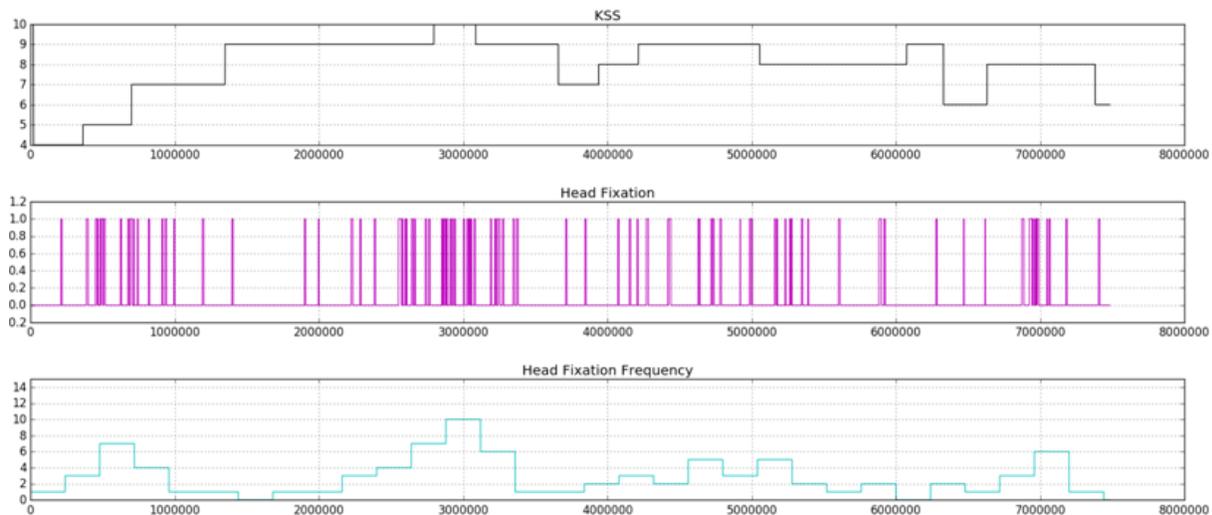


Figure 6 : Head fixations vs drowsiness

The tests performed on a simulator database have confirmed some correlation for the highest drowsiness level. Still the validity of this result has to be confirmed in real driving conditions where the driving conditions, the traffic, the presence of a passenger etc. may change significantly the head movement behaviour compared to the monotonous simulator driving scenarios.

The main working line is then to exploit the changes of the characteristics of the head movement when the driver becomes drowsy. This approach is based on a state of the art on psychophysiology studies which demonstrate that drowsiness is characterized by a loss of the muscular tonicity. One of the consequence is that driver react by changing their posture and repositioning



head movements. J-C. Popieul, P. Simon and P. Loslever [³] also note a reduction of the head movement speed. D. Lee, S. Oh, S. Heo, M. Hahn [⁴] propose a drowsiness model based on head leaning and head proximity to the head rest. Jürgen Schmidt [⁵] uses intentional head movements defined as wide and fast movements to detect alert state.

Considering these effects the 2nd cycle work is focusing on exploring intended head movements to detect alert phases and exploring the variation of the movement characteristics to detect drowsy phases.

The work includes various steps:

- Labelling of head movements;
- Development of an algorithm which detects head movements, measurement of the head movement characteristics and classification;
- Development of a head based drowsiness model.

The head movements were classified into the 10 classes shown in Table 2.

³ J-C. Popieul, P. Simon, P. Loslever, "Using driver's head movements evolution as a drowsiness indicator", Intelligent vehicles symposium – Proceedings IEEE, (2003), France

⁴ D. Lee, S. Oh, S. Heo, M. Hahn, "Drowsy driving detection based on the driver's head movement using infrared sensors", Second international symposium on universal communication – IEEE, (2008), South Korea

⁵ J. Schmidt, Christian Braunagel, Wolfgang Stolzmann, and Katja Karrer-Gauß, "Driver Drowsiness and Behavior Detection in Prolonged Conditionally Automated Drive", 2016 IEEE Intelligent Vehicles Symposium (IV) Gothenburg, Sweden, June 19-22, 2016



Table 2: Head movement classes

No.	Movement definition
1	Movement to look at the rear-view mirror exterior left
2	Movement to look at the rear-view mirror interior (central)
3	Movement to look at the rear-view mirror exterior right
4	Movement to reposition the head
5	Movement while the driver is touching the face
6	Movement to look at the window
7	Movement to look at the radio/central screen
8	Movement to look something unknown
9	Movement of head but keep looking the road
10	Movement of head while the driver is talking

For each movement the following characteristics are computed: amplitude, duration, duration of the rise (toward the area the driver wants to observe), duration of the descent (back to the default position), duration of the plateau and speeds. The plot shown in Figure 7 highlights the yaw of typical head movement of a driver looking at the central mirror, fixes the mirror then goes back to the on road position.

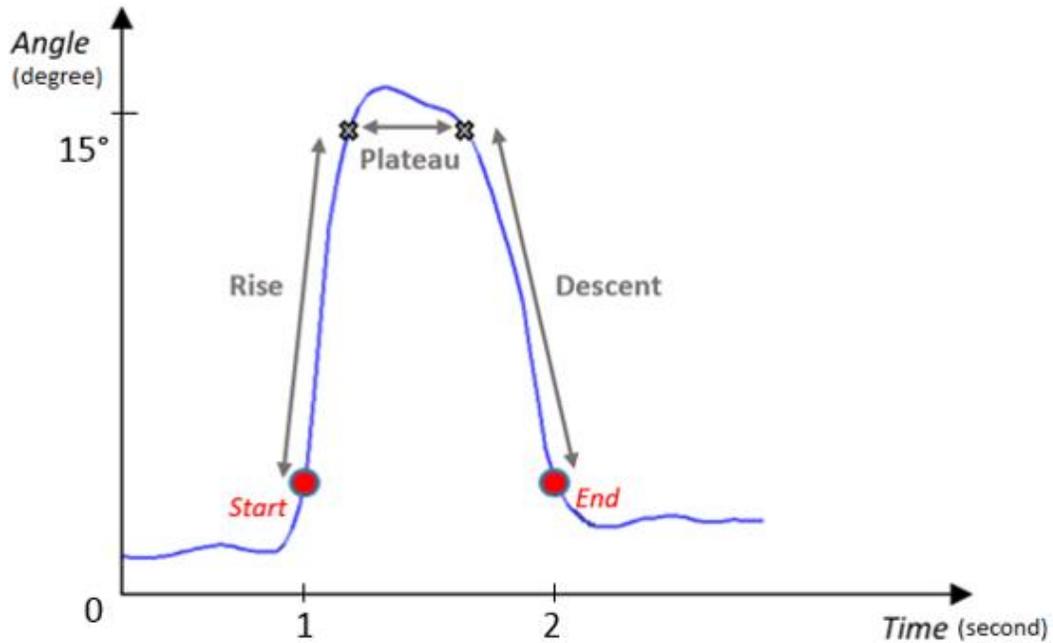


Figure 7: Yaw of head movement when the driver looking at the central mirror, fixes the mirror then goes back to the on road position

Figure 8 shows the same action but the driver fixes the gaze to the central display before going back to the on road position. This behaviour illustrate the difficulty to extract robust measures.

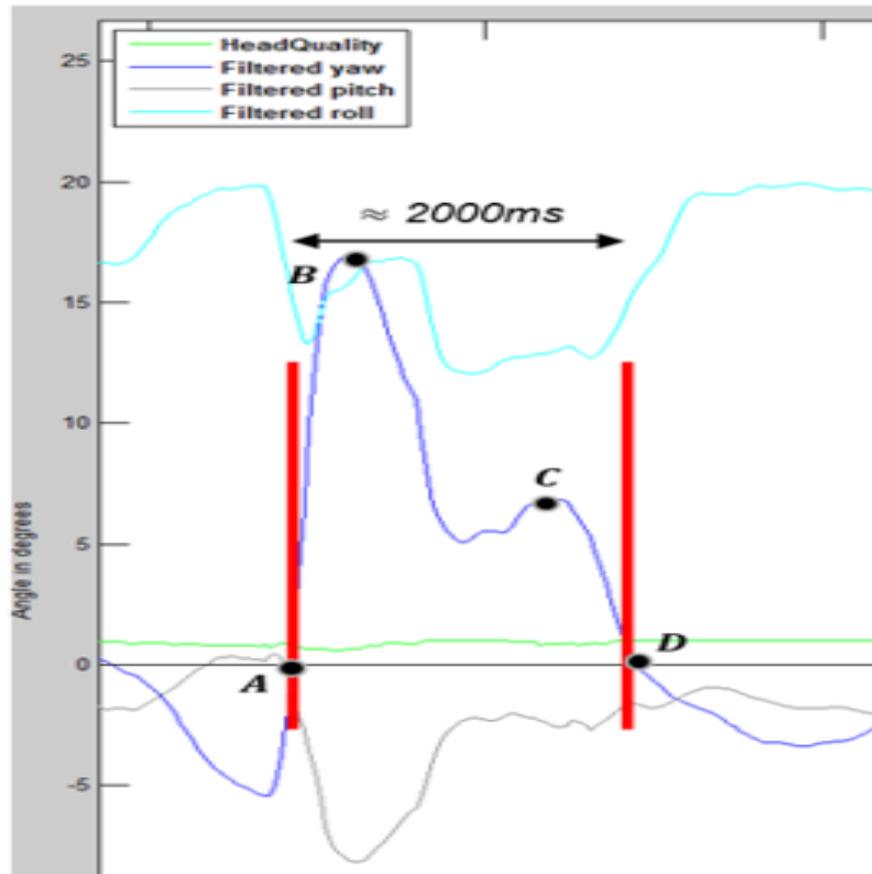


Figure 8: Yaw of head movement when the driver looking at the central mirror, fixes the mirror then fixes the gaze to the central display before going back to the on road position

A drowsiness model was developed based on the occurrence of repositioning movements combined with a diminution of the movement speed. We have to deal with two major issues:

- The first one is related to the accuracy of the measurement of the head movement characteristics.
- The second one is related to the monotonous driving conditions in highway with very few traffic. Indeed, in such conditions, driver's head



rotation are very seldom. It is then difficult to filter out noisy measure from the individual variability.

The ongoing work on drowsiness model focuses on the development of a drowsiness model based on head movement characteristics along with the improvement of the eye/eyelid based diagnostic. Other features like yawning and face rub will also be exploited.

3.1.2.3 Visual attention/distraction

The visual attention/distraction work focused mainly on improving the accuracy and quality of the eye gaze and head gaze signals. Indeed the performance of distraction model is directly related with the accuracy and the robustness of the eye gaze and head gaze output of the face tracker.

The work lines are then twofold:

- Improved face tracker algorithm. An updated version of the current face tracker has been integrated. The rotation range are significantly improved to almost $\pm 90^\circ$ in Yaw, $\pm 45^\circ$ in roll, $>25^\circ$ upward pitch and $>50^\circ$ downward.
The new face tracker is now robust to driver wearing masks or hand on mouth.
- Improved driver's state HW as mentioned before.



3.2 E1.2 – V2x communication

3.2.1 Scenario and uses case where E1.2 is relevant

V2x is essential part of future ITS systems. It directly improves the security of transportation, and it can improve the traffic flow and transport efficiency.

In AutoMate project, there are two scenarios in which V2x has a crucial role, even if in these scenarios the utilization of V2x differs.

In the MARTHA scenario V2I acts as an additional sensor, meaning that the TeamMate car is able to receive information about the environment (i.e. road works ahead), which would not be available in time for a safe cooperation. This is based on the simple information sharing concept of V2x.

On the other hand, in the EVA scenario, the cooperation concept of V2V is highlighted, namely Eva and the TeamMate car do not just receive information about the roundabout related traffic situation, but they can also affect other cars' behaviour by automatically negotiate their entrance in the roundabout with the other vehicles. Practically, this means that the TeamMate car's embedded intelligence is able to conclude from periodic V2V message exchanges, if there is another vehicle, which would like to allow into the roundabout or there is enough space to drive in safely. This allows to solve the traffic situation cooperatively with other vehicles, i.e. when and how the TeamMate car can enter in the roundabout.



Therefore, as shown in Table 1, Enabler E1.2 is needed to implement a support in perception from the automation to the human (A2H) to complement the perception of the driver about the environment.

Two use cases of MARTHA and EVA scenario have been revised to highlight and clarify the role of E1.2 to implement this cooperation.

Martha is driving in an extra-urban road in Manual Mode. She receives an incoming call: the car detects that she is distracted, and this could lead to an unsafe behaviour. The TeamMate car offers a cooperation in action, suggesting a handover in order to shift to Automatic Mode. Martha accepts the suggestion and cedes the control.

Eva is driving in an urban road in Manual Mode. When she arrives in a roundabout, she is not able to decide the right moment to get into it. Without a support, Eva would wait several minutes before performing the manoeuvre. Through the V2V communication, the vehicle negotiates the entrance into the roundabout with the other cars and suggests to Eva the right moment to perform the manoeuvre. Eva enters the roundabout still in Manual Mode.

3.2.2 Implementation

In the 2nd cycle we identified the specific V2x messages, which have been used in the scenarios:

- Road Works Warning (RWW) message is a type of Decentralized Environmental Message (DENM) that describes how to drive through or bypass the road work area; DENM is sent by the infrastructure to vehicles; RWW message is used in the Martha scenario;



- similarly to RWW, DENM is able to carry information about traffic condition, i.e. about traffic density; this information could be used in the Eva scenario;
- Cooperative Awareness Message (CAM) provides by means of periodic sending of status data (basic status, position, current speed etc.) a cooperative awareness to surrounding vehicles, therefore it is sent by vehicles to vehicles; CAM message is used in the Eva scenario.

We use Cohda MK5 [6] devices of a third party supplier as V2x communication capable units. These devices have customized embedded Linux operating system to support the special needs of the V2x communication hardware. Furthermore, application is provided to transmit or receive CAM and some kind of DENM messages.

In this cycle, we customized and configured the mentioned application to transmit the proper RWW message (there was no need to change CAM). Then, we developed test scripts to be able to verify the capability of the Cohda devices and get initial test results on their performance.

In addition, we started to implement a gateway application, which is able to aggregate and distribute V2x message on socket basis in the local network (e.g. LAN, Wi-Fi) for other applications. The benefit of such gateway application is that it makes the lower layers of V2x communication transparent for other applications. In addition, it allows to handle custom V2x messages in a more simple way.

⁶ Cohda Wireless MK5 OBU, <http://www.cohdawireless.com/solutions/hardware/mk5-obu/>



3.3 E2.1 – Driver Intention Recognition

3.3.1 Scenario and uses case where E2.1 is relevant

As shown in Table 1, Enabler E2.1 is needed to implement a support in perception from the automation to the human (A2H) to complement the perception of the driver about his/her intention.

One of the use cases of PETER scenario has been revised to highlight and clarify the role of E2.1 to implement this cooperation.

Peter is driving in a narrow rural road in Manual Mode. He approaches a tractor that causes limited visibility on the road. The TeamMate car detects a car approaching from the opposite lane. Since Peter is not aware of the car, he decides to overtake, and the TeamMate car detects his intention. In order to avoid an imminent collision, the TeamMate car informs Peter about the approaching vehicle and warns him about the risky manoeuvre. Peter suddenly becomes aware of the risk, and he does not perform the overtaking until it is safe.

Moreover, E2.1 has also been used to implement a strategy to trigger the decisions of the automation. In particular, when the TeamMate car has to overtake the tractor in Automated Mode in PETER scenario, a trigger is necessary to identify the right moment for the automation to start the manoeuvre. Therefore, E2.1 is used in this case to identify the conditions for the automation to overtake by learning from the behaviour that the driver had in Manual Mode. In this case the E2.1 is not an enabler of the cooperation, but it is an enabler of the automation strategy (without E2.1, the TeamMate car



would never overtake the tractor, creating frustration to the driver and thus reducing the acceptance of the automation).

3.3.2 Concept

Driver intention recognition most commonly addresses the problem of anticipating driving manoeuvres a driver is likely to perform in the next few seconds. As early knowledge about such manoeuvre intentions may serve as a potential enabler to generate adaptive warnings and early interventions before a potential dangerous manoeuvre is initiated, driver intention recognition is of ever increasing importance for the development of advanced driver assistance systems and has become a popular research topic in recent years.

Approaches reported in the literature (some comparative reviews are provided e.g., by Doshi and Trivedi [⁷] and Lefèvre et al. [⁸]) mainly differ in respect to the selected scenarios and addressed manoeuvres, modelling techniques used, and the sensor input considered.

On a conceptual level, we will roughly distinguish between two potential different sources of information for driver intention recognition: *causes* and *effects*. Here, causes should be understood as information perceived by the

⁷ Doshi, A. and M. M. Trivedi (2011), "Tactical Driver Behavior Prediction and Intent Inference: A Review", in *Proceedings of the 14th International IEEE Conference on Intelligent Transportation Systems*, pp. 1892-1897.

⁸ Lefèvre, S., D. Vasquez, and Ch. Laugier (2014), "A Survey on Motion Prediction and Risk Assessment for Intelligent Vehicles", in *Robomech Journal*, 1, 1, pp. 1-14.



driver that results in the formation of an intention, e.g., a slow driving lead vehicle in the case of overtaking intentions. In contrast, effects should be understood as the observable effects on the overall behaviour of the driver and vehicle, resulting from the existence of an intention, e.g., to stay with the example of overtaking behaviour, a head movement to check the blind spot or the initiation of an overtaking manoeuvre.

Traditionally, driver intention recognition focusses on modelling the relations between manoeuvre intentions and their effects on the overall behaviour of the vehicle and driver. As such, considered input for driver intention recognition is usually limited to information about the *vehicle state* and the *driver*, to be provided by internal and external sensor systems of the vehicle.

Here, information about the vehicle state should be understood as subsuming information about the vehicle dynamics, such as velocity and acceleration, steering wheel angle and yaw rate, pedal positions etc., usually readily available via the Controller Area Network (CAN) bus of the vehicle, as well as information about the lateral position and alignment of the vehicle in the road, and the current speed limit, provided e.g., by camera systems or derived from GPS and digital maps. Most commonly, driver intention recognition based on information about the vehicle state is realized by comparing the observable vehicle state sequence with expected sequences for each addressed manoeuvre intention (c.f., Section 3.4.2.2).

A severe limitation of such approaches in respect to an early recognition of driver intentions, however, results from the fact that a manoeuvre needs to be already initialized to be recognized. However, as aptly stated by Ohn-Bar



et al. (p. 720) [9] “*the intent to perform the manoeuvre existed before the trajectory of the vehicle was altered and can be observed earlier*”.

To overcome these limitations and extend the predictive capabilities for driver intention recognition, many authors emphasize the value and resulting need of additional *driver-based* information, to be understood as subsuming information like head, eye, foot, and hand positions and movements, usually obtained via driver monitoring systems. Such driver-based input provides valuable information due to the way that drivers prepare for manoeuvres, e.g., via head movements to check the blind-spot. Unfortunately, although the value of driver-based input for intention recognition must not be undervalued, their inclusion only shifts the recognition of manoeuvre intentions to earlier stages of execution.

Concerning the causes for intention, potential information is available from the situation context itself, e.g., by the means of information about the vehicles in the vicinity of the driver. Up to now, potentially due to limited sensor capabilities, such information has not been used thoroughly for the purpose of driver intention recognition, but it is either neglected entirely or restricted to the immediate surrounding of driver, e.g., to the lead vehicle or seldom potential vehicle in the blind spots. This is surprising, as information about the

⁹ Ohn-Bar, E., A. Tawari, S. Martin, and M. M. Trivedi (2014), “Predicting Driver Maneuvers by Learning Holistic Features”, in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 719-724.



current traffic situation should be able to provide information suitable to actually *predict* the intentions of the driver prior to their execution.

Furthermore, as addressed in AutoMate, the increasing introduction of automation to the vehicle, may result in the effects of intentions to become misleading, and, in the case of fully autonomous driving, potentially obsolete. Obviously, in the case of autonomous driving, intention recognition for the “passenger” is no longer needed for safety reasons. We do, however, believe that autonomous systems may gain value a currently unused value from knowing whether a driver *would* have the intention to overtake, if he was in control, such as to enable the autonomous system to comply with the usual behaviour of the driver and communicate when such compliance cannot be achieved.

3.3.3 Implementation

The probabilistic driver model for intention recognition and behaviour prediction (in the following simply referred to as *model*) is conceptualized as a Dynamic Bayesian Network that models the causal and statistical relations between the driver’s intentions, driving manoeuvres resp. behaviours, and the situational context, as observable by the TeamMate vehicle’s sensor and communication platform.

In the 2nd cycle of AutoMate, the development of the model focused on the Peter scenario, dealing with overtaking scenarios on rural roads.

The purpose of the model in the Peter scenario is to constantly provide the TeamMate vehicle with an online recognition of the current intentions of the driver (to be used for the cooperation) and also to collect data to be used by



E4.2 (Learning of intention from the driver) to learn when to trigger the overtaking in Automated Mode and to perform it in a human-like style.

If the driver is in control of the TeamMate vehicle (manual driving), the information provided by the model can be used to assess the safety of the intended driving manoeuvre.

If the automation is in control of the TeamMate vehicle (autonomous driving), the information provided by the model can serve as a mechanism to learn and trigger the most appropriate manoeuvres to the automation.

The problem addressed in the 2nd cycle is the recognition of the current manoeuvre intention and actually performed manoeuvre with respect to the driving behaviour of the driver, based on the situational context. In the Peter scenario, we consider three primary driving behaviours: performing lane changes to the left lane (LCL), to the right lane (LCR), and lane-keeping behaviour (LK), represented by a discrete variable $B, Val(B) = \{b_{LCL}, b_{LCR}, b_{LK}\}$.

Corresponding to these behaviours, we consider three potential intentions: the intention to change to the left lane (i.e., the intention to overtake), to return to the right lane (in order to complete an overtaking manoeuvre), and the absence of a lane change intention.

For modelling purposes, it is however more convenient to replace such lane change intentions with *target lane intentions*, i.e., whether the driver intends to drive on the left or on the right lane, represented by a binary variable $I, Val(I) = \{i_L, i_R\}$. Let, correspondingly, $L, Val(L) = \{l_L, l_R\}$ denote a binary variable



that represents whether the TeamMate vehicle is located in the left or right lane. It should be apparent that knowing the current lane, target lane intentions can easily be transformed into lane change or overtaking intentions, in that a lane change intention is present if the current lane and the target lane intentions differ.

For the 2nd cycle, we focus on a model for driver intention recognition that refrains from driver-based input and instead tries to focus on information provided by the traffic situation as potential causes for the formation of intentions. Let \mathbf{O}_I denote a set of discrete and continuous variables representing the causes for the formation of intentions and let \mathbf{O}_B denote a set of discrete and continuous variables representing the observable effects of intentions in terms of driving behavior.

The model is based on the assumption that the temporal evolution of intentions and behaviours can be expressed as two hidden first-order Markov processes. More specifically we assume that for any number of time steps $T \geq 1$, the conditional joint distribution $p(I^{1:T}, B^{1:T}, \mathbf{O}_B^{1:T} | L^{1:T}, \mathbf{O}_I^{1:T})$ can be factorized, according to the graph structure shown in Figure 9, as:

$$\begin{aligned} p(I^{1:T}, B^{1:T}, \mathbf{O}_B^{1:T} | L^{1:T}, \mathbf{O}_I^{1:T}) &= p(I^{1:T} | L^{1:T}, \mathbf{O}_I^{1:T}) p(B^{1:T}, \mathbf{O}_B^{1:T} | I^{1:T}, L^{1:T}) \\ &= p(I^1 | L^1, \mathbf{O}_I^1) p(\mathbf{O}_B^1 | B^1) p(B^1 | I^1, L^1) \prod_{t=2}^T p(I^t | I^{t-1}, L^t, \mathbf{O}_I^t) p(\mathbf{O}_B^t | B^t) p(B^t | B^{t-1}, I^t, L^t). \end{aligned}$$

As such, we assume that the model can be defined in terms of two components, a component for intention recognition, realized akin to a Maximum-entropy Markov Model, where for any number of time steps T the (conditional) joint distribution $p(I^{1:T} | L^{1:T}, \mathbf{O}_I^{1:T})$ is defined as



$$p(I^{1:T} | L^{1:T}, \mathbf{o}_I^{1:T}) = p(I^1 | L^1, \mathbf{o}_I^1) \prod_{t=2}^T p(I^t | I^{t-1}, L^t, \mathbf{o}_I^t),$$

and a component for behaviour recognition, realized akin to an (input-dependent) Hidden Markov Model with factorized observation model, where for any number of time steps T the (conditional) joint probability distribution $p(B^{1:T}, \mathbf{o}_B^{1:T} | I^{1:T}, L^{1:T})$ is defined as

$$p(B^{1:T}, \mathbf{o}_B^{1:T} | I^{1:T}, L^{1:T}) = p(B^1 | I^1, L^1) p(\mathbf{o}_B^1 | B^1, I^1, L^1) \prod_{t=2}^T p(B^t | B^{t-1}, I^t, L^t) p(\mathbf{o}_B^t | B^t, I^t, L^t).$$

These components can be interpreted as follows: we assume that intentions evolve based on the situational context encountered. Intentions then manifest themselves by the execution of driving manoeuvres whose effects can then be observed.

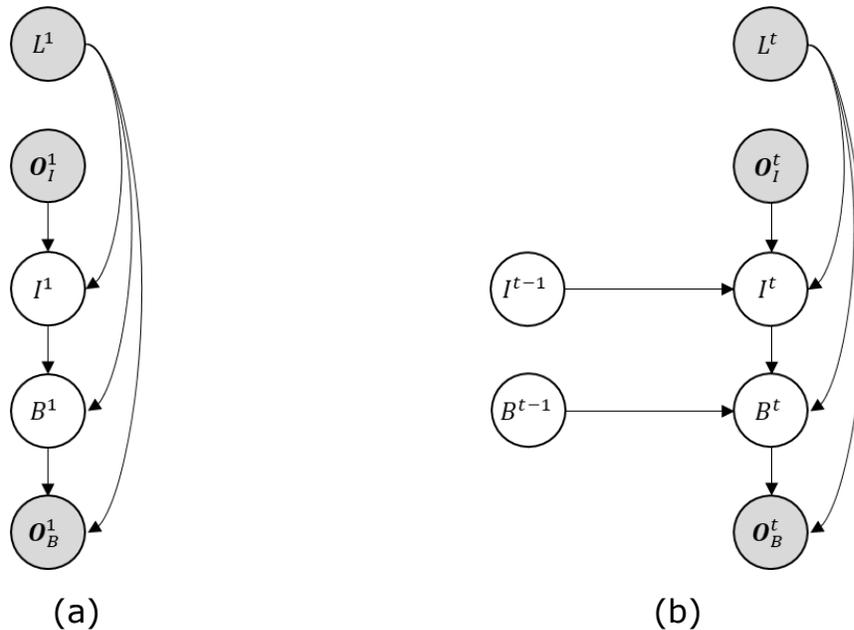


Figure 9: Conceptual graph structures of the initial BN and the 2TBN of the model. Grey nodes are assumed to be observed during inference.



During runtime, the model can be used to continuously maintain a joint belief state over the current target lane intentions and manoeuvres $p(I^t, B^t | l^{1:t}, \mathbf{o}_I^{1:t}, \mathbf{o}_B^{1:t})$ given all available input obtained thus far, via recursive Bayesian filtering adapted to the structure of the model:

$$p(I^t, B^t | l^{1:t}, \mathbf{o}_I^{1:t}, \mathbf{o}_B^{1:t}) \\ \propto p(\mathbf{o}_B^t | B^t, l^t) \sum_{b \in B} p(B^t | b^{t-1}, I^t, l^t) \sum_{i \in I} p(I^t | i^{t-1}, l^t, \mathbf{o}_I^{1:t}) p(i^{t-1}, b^{t-1} | l^{1:t-1}, \mathbf{o}_I^{1:t-1}, \mathbf{o}_B^{1:t-1}).$$

From this joint belief state, separate belief states over intention $p(I^t | l^{1:t}, \mathbf{o}_I^{1:t}, \mathbf{o}_B^{1:t})$ and behaviors $p(B^t | l^{1:t}, \mathbf{o}_I^{1:t}, \mathbf{o}_B^{1:t})$ can easily be derived via marginalization. If in autonomous mode, we can simply use the sub-component for intention recognition to derive $p(I^t | l^{1:t}, \mathbf{o}_I^{1:t})$.

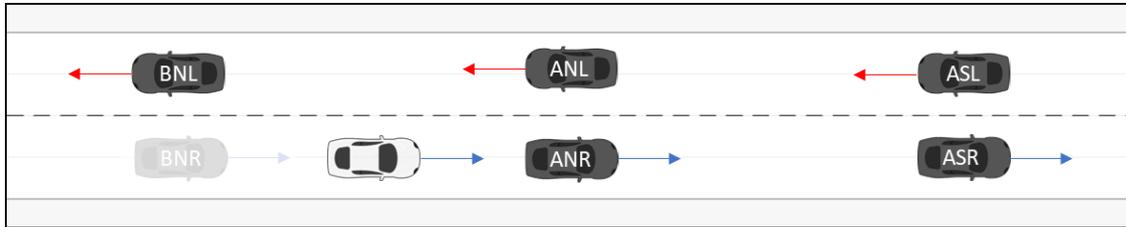
As provided, the model should be understood as conceptual, in that the parameters and finer structure, e.g., which variables constitute \mathbf{O}_I and \mathbf{O}_B , must be provided based on prior expert knowledge and/or derived from multivariate time-series of human behaviour data via the use of machine-learning methods.

OFF conducted a data collection study for the Peter scenario in the OFF driving simulator (described in details in section 4.3) to gather data for training (and validation) of the driver model for intention and behaviour recognition.

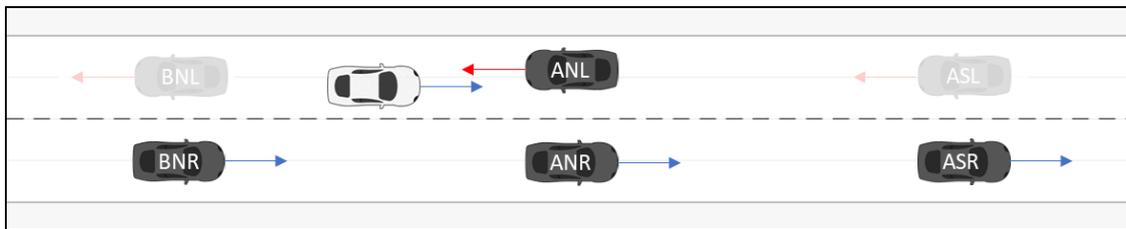
For the input for intention recognition in the 2nd cycle, we focused on a subset \mathbf{O}_I of the totally available input obtained in the simulator experiment, for which we considered a set of 25 variables, as described in Table 3. More specifically, we considered up to two vehicles in front and a single vehicle behind the TeamMate vehicle, on each the left and right lane (Figure 10). The vehicles are assigned to fixed “roles” based on their relative positions to the TeamMate



vehicles and each other. To improve the robustness to noise, we excluded the BNR vehicle for cases where the TeamMate vehicle is located on the right lane, and the BNL and BNR vehicles for cases where the TeamMate vehicle is located on the left lane.



(a)



(b)

Figure 10: Assignment of vehicles in the vicinity of the TeamMate vehicle (light)

Table 3: Overview of the observation variables (features) o_i , currently considered for intention recognition.

Variable	Type	Description
E_X	Binary	Represents, whether there exists a vehicle X in the traffic situation. $X \in \{BNL, BNR, ANL, ANR, ASL, ASR\}$.
T_X	Binary	Represents the type (PKW or LKW) of the vehicle X .
D_X	Continuous	Represents the distance between the TeamMate vehicle and a vehicle X along the course of the road.
S_X	Continuous	Represents the speed of a vehicle X .



V	Continuous	Represents a hypothetical viewing distance for the TeamMate vehicle.
-----	------------	--

As potential manifestations of a shown manoeuvre execution, for the 2nd cycle, we consider $O_B = \{Y, LP\}$, as described in Table 4.

Table 4: Overview of the observation variables (features) O_B , currently considered for behaviour classification.

Variable	Type	Description
Y	Continuous	Represents the heading angle, resp. yaw angle in respect to the course of the road.
LP	Continuous	Represents the lateral deviation from the centreline.

We aim to use machine-learning methods to derive important variables for intention recognition and behaviour classification and used discriminative learning techniques for deriving the sub-component for intention recognition. Basically, we tried to find a subset $\mathbf{O}_{Rel} \in \mathbf{O}_I$ of relevant features that allow for optimal intention recognition. We compared three different realizations of the CPDs $p(I^t|I^{t-1}, l^t, \mathbf{o}_I^t)$ resp. $p(I^1|l^1, \mathbf{o}_I^1)$, in the following denoted as M_{Gauss} , M_{GMM} , and M_{LR} . For M_{Gauss} , $p(I^t|I^{t-1}, l^t, \mathbf{o}_I^t)$ was learned in terms of an embedded Bayesian classifier

$$p(I^t|I^{t-1}, l^t, \mathbf{o}_I^t) = \frac{1}{Z(I^{t-1}, l^t, \mathbf{o}_I^t)} p(I^t|I^{t-1}, l^t) p(\mathbf{o}_I^t|I^t, l^t),$$

with $Z(I^{t-1}, l^t, \mathbf{o}_I^t) = \sum_{i \in I} p(i^t|I^{t-1}, l^t) p(\mathbf{o}_I^t|i^t, l^t)$ representing a normalization constant and $p(\mathbf{o}_I^t|I^t, l^t)$ factorizing according to an augmented naïve Bayesian classifier, with distributions over continuous variables being approximated by Gaussian distributions. For M_{GMM} , $p(I^t|I^{t-1}, l^t, \mathbf{o}_I^t)$ was learned the same as



M_{Gauss} , however, with distributions over continuous variables being approximated by mixture of Gaussians (also known as Gaussian Mixture Models (GMMs)), potentially better suited to approximate the underlying densities. Lastly, for M_{LR} , $p(I^t|I^{t-1}, l^t, \mathbf{o}_I^t)$ was learned as a logistic regression. Based on the intuition that information required for intention and manoeuvre recognition is strongly influenced by whether the TeamMate vehicle is located on the left or right, we allow for context-specific independence by considering two distinct realizations of $p(I^t|I^{t-1}, l^t, \mathbf{o}_I^t)$ to be used when travelling on the left $p(I^t|I^{t-1}, L^t = l_L^t, \mathbf{o}_I^t)$ or the right lane $p(I^t|I^{t-1}, L^t = l_R^t, \mathbf{o}_I^t)$ on the road, for each M_{Gauss} , M_{GMM} , and M_{LR} .

In contrast, for this cycle, the CPD $p(\mathbf{O}_B^t|B^t, L^t)$ was simply decomposed akin to a naïve Bayesian classifier

$$p(\mathbf{O}_B^t|B^t, L^t) = p(Y^t|B^t, L^t)p(LP^t|B^t, L^t),$$

with $p(Y^t|B^t, L^t)$ and $p(LP^t|B^t, L^t)$ approximated as GMMs.

We used the training set D_{Train} to learn the structure and corresponding parameters of the three different realizations M_{Gauss} , M_{GMM} , and M_{LR} . The resulting graph structure for M_{Gauss} is shown in Figure 11, Figure 12 shows the resulting graph structure for M_{GMM} , and Figure 13 shows the resulting graph structure for M_{LR} . In each case, the left side shows the graph structure realizing $p(I^t|I^{t-1}, L^t = l_L^t, \mathbf{o}_I^t)$, while the right side shows the graph structure realizing $p(I^t|I^{t-1}, L^t = l_R^t, \mathbf{o}_I^t)$, with the paled underlying structure implying the considered search space with non-considered variables being omitted.

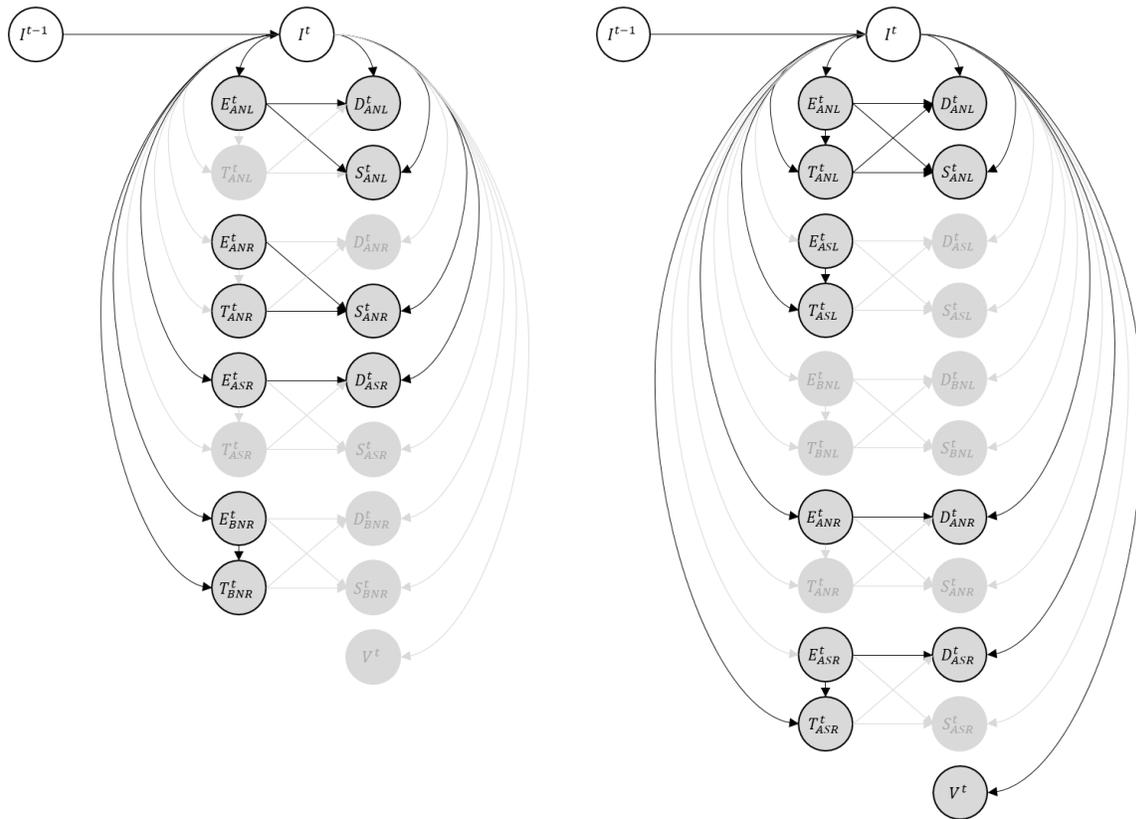


Figure 11: Learned graph-structures of the model M_{Gauss} .

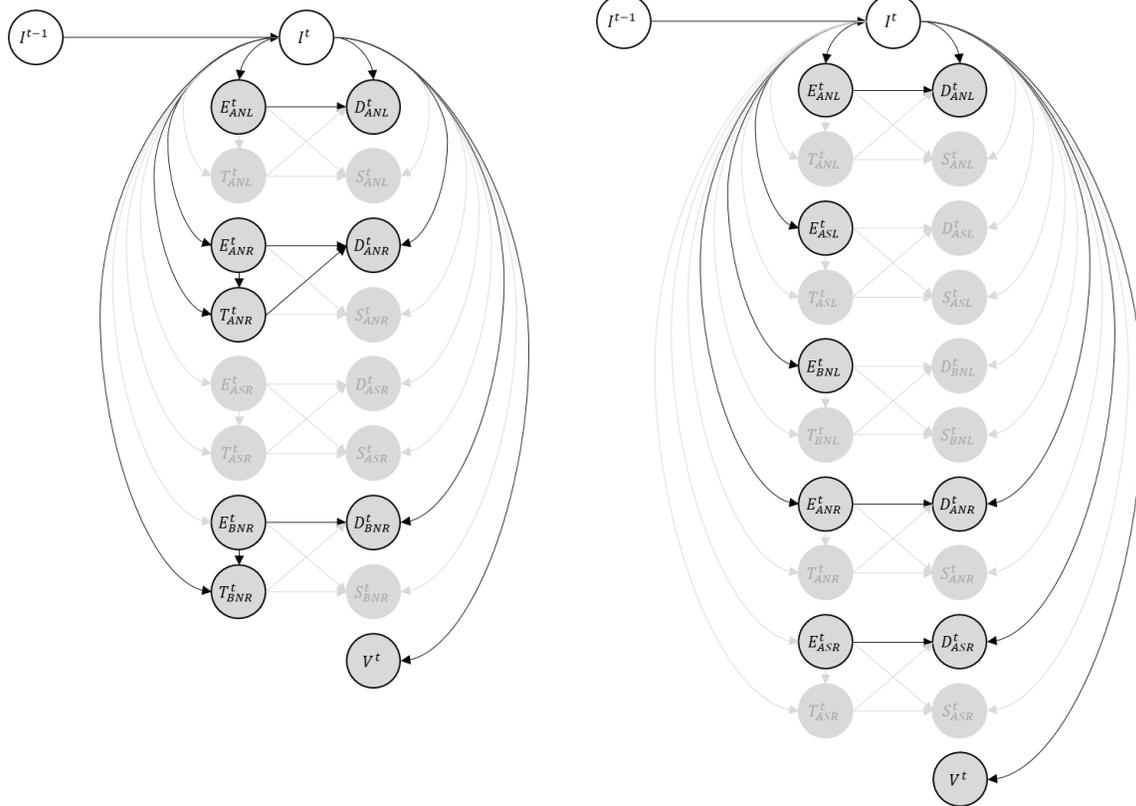


Figure 12: Learned graph-structures of the model M_{GMM} .

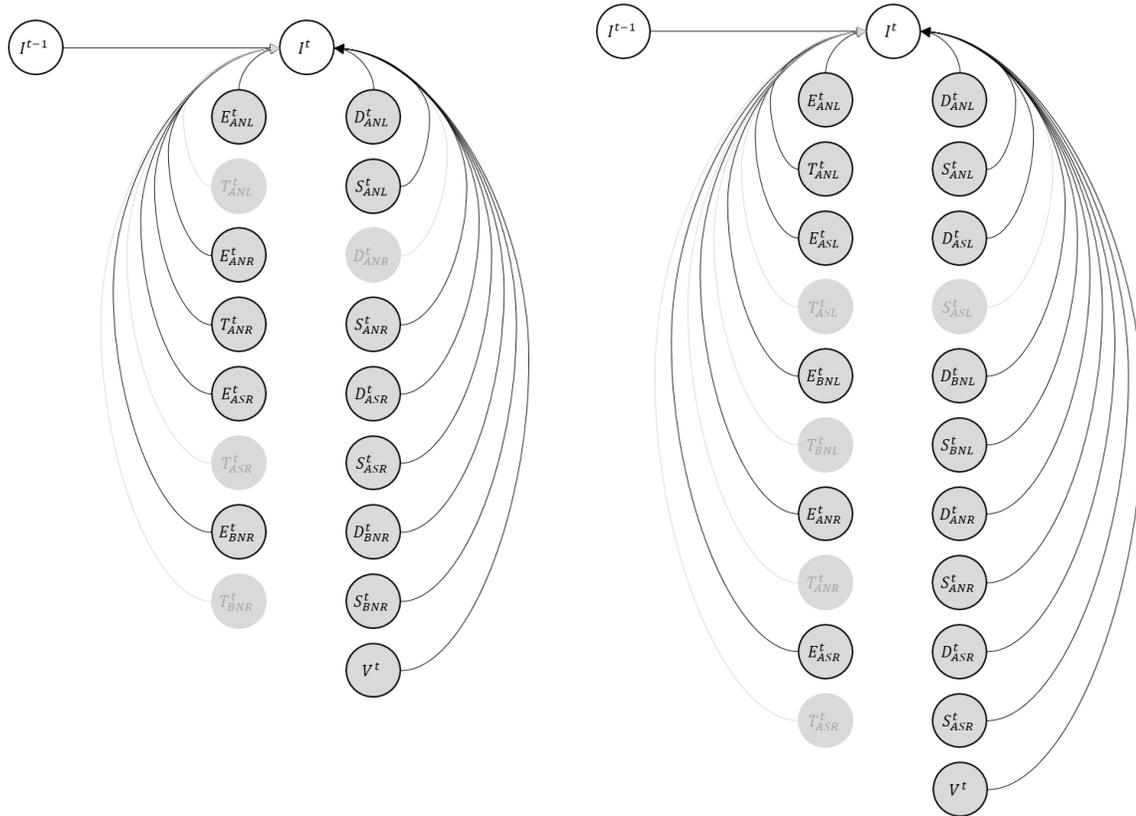


Figure 13: Learned graph-structures of the model M_{LR} .



3.4 E3.1 – Situation and vehicle model

3.4.1 Scenario and uses case where E3.1 is relevant

As shown in Table 1, Enabler E3.1 is needed to implement a support in perception from the automation to the human (A2H) to complement the perception of the driver about the situation and the vehicle.

The same use case of PETER scenario already described for E2.1 has been considered to highlight and clarify the role of E3.1 to implement this cooperation.

Peter is driving in a narrow rural road in Manual Mode. He approaches a tractor that causes limited visibility on the road. The TeamMate car detects a car approaching from the opposite lane. Since Peter is not aware of the car, he decides to overtake, and the TeamMate car detects his intention. In order to avoid an imminent collision, the TeamMate car informs Peter about the approaching vehicle and warns him about the risky manoeuvre. Peter suddenly becomes aware of the risk, and he does not perform the overtaking until it is safe.

3.4.2 Implementation

The situation and vehicle model consists of a centreline of the lane the vehicle is currently driving on. Also there are boundary lines of the lane included, to specify the available free space. Static obstacles are excluded by these boundary lines, so they are implicitly considered in the situation model. Other traffic participants (in the Peter scenario there are just cars regarded) are modelled by the x-y position of their rear axis, their speed, acceleration, orientation and of course of their shape.



The modelling provides probabilistic information about the state. This information can be used to calculate a more intelligent and safe manoeuvring behaviour for the vehicle.

The enabler has been developed by taking into consideration the following 2 features:

1. Semantic enrichment of the situation model
2. Prediction of the future evolution of the traffic scene

3.4.2.1 Semantic enrichment of the situation model

The semantic enrichment module extends scene objects provided by the perception layer with semantic information. This semantic information describes interaction between scene objects. Furthermore, the set of legal manoeuvres a vehicle can legally perform is inferred.

In the 1st project cycle, we proposed to use ontology with logical rules and a reasoner to address this task. The concept and first results on simulated data was presented in Deliverable 2.2 [10].

The goal of the 2nd cycle was to integrate the ontology, the logical rules and the reasoner into the situation interpretation module. For that purpose, we implemented the *JNIOWLBridge* module to access the OWL ontology and the reasoner in a C++ function, since the most suitable OWL API and reasoner exist only as Java implementations. The *JNIOWLBridge* therefore builds a bridge between the available Java OWL/reasoner API and our C++ module. The *JNIOWLBridge* module consists of classes for Loading the Java Virtual

¹⁰ Sensor Platform and Models including V&V results from 1st cycle



Machine (see Figure 14) and handling ontology as well as the reasoner (see Figure 15, Figure 16 and Figure 17).

```
class JVMLoader
{
private:

    JVMLoader(JVMLoader const&);    //

    void operator=(JVMLoader const&); //

    JVMLoader (); /* disable def. ctor */

    void closeJVM();

public:

    JavaVM * java_vm_;

    JNIEnv * java_env_;

    static JVMLoader& getInstance()
    {
        static JVMLoader instance; //
        // Instantiated on first use.
        return instance;
    }
}
```

Figure 14: JVMLoader class declaration



```
class OWLOntology
{
private:

    const string path_to_ontology_;

    jobject onto_access_obj_;

    jclass onto_access_cls_;

    map<string, jmethodID> JNIMethodIDMap;

    vector<OWLIndividual> individuals_;

    string exec(const char* cmd);

    string generateJNIMethodList();

    void loadOntology();
```

Figure 15: OWLOntology class declaration part 1



public:

```
vector<OWLIndividual> individuals(){return individuals_;}

/**
 * =====
 * base methods
 * =====
 **/
OWLOntology(string cls_path);

~OWLOntology(){

void resetOntology(OWLOntology ontology);

void saveOntology();

void saveOntology(string file_name);

/**
 * =====
 * add methods
 * =====
 **/

void addIndividualToOntology(string indi_name,string cls_name);

void addObjPropertyToIndividual(string indi1_name, string propertyName, string indi2_name);

void addDataPropertyToIndividual(string indi1_name, string propertyName, string indi2_name);

/**
 * =====
 * remove methods
 * =====
 **/

void removeAllIndividuals();
```

Figure 16: OWLOntology class declaration part 2



```
void removeAllIndividuals();

void removeClassAssertionFromIndividual(string indi_name);

void removeIndividualFromOntology(string indi_name);

void removeObjectPropertyFromIndividual(string indi_name,string propertyName, string indi_name2);

void removeObjectPropertyFromIndividual(string indi_name, string propertyName);

void removeDataPropertyFromIndividual(string indi_name,string propertyName, string value);

void removeDataPropertyFromIndividual(string indi_name,string propertyName);

/**
 * =====
 * inference methods
 * =====
 **/

void doInference();

bool isConsistent();

};
```

Figure 17: OWLOntology class declaration part 3

The *JNIOWLBridge* is linked to our C++ semantic enrichment module as a library. The ontology and logical rules used for the semantic enrichment was described in D2.2. The ontology contained the taxonomy and semantic relations of relevant scenes objects as pedestrian, road, vehicle, traffic light and signal. The logical rules described basic traffic rules in urban scenes. The semantic enrichment module takes as input the detected scene objects and the modelled ontology and executes the following steps:

1. loading of ontology by name
2. generation of relations between detected scene objects (which are treated as individuals), such as “vehicle x is on lane y”
3. adding of these individuals and theirs relations to the ontology



4. inference of new relations between the individual objects
5. inference of legally allowed manoeuvres per vehicle
6. deletion of previously added individuals and relations from the ontology
7. repetition of algorithm starting at step 2

3.4.2.2 Predicting the future evolution of the traffic scene (OFF)

The purpose of vehicle models is to predict the temporal and spatial evolution of the traffic scene, based on the information provided by the sensor and communication platform and the situation model, as a necessary input for online risk assessment (for more information on online risk assessment, we refer to D3.3 “Concepts and algorithms incl. V&V results from 1st cycle” and D3.5 “Concepts and algorithms incl. V&V results from 2nd cycle”).

3.4.2.2.1 Concept

In the following, we assume that the sensor and communication platform provides all necessary information as previously described in D2.2 “Sensor Platform and Models including V&V results from 1st cycle” to the situation model, where it is augmented by the semantic enrichment. To recapitalize, we expect that at each time step t , the situation model provides:

- A belief state $p(\mathbf{X}_{TM}^t | \mathbf{o}^{1:t})$ about the current state of the TeamMate vehicle, where $\mathbf{X}_{TM}^t = \{X_{TM}^t, Y_{TM}^t, \Theta_{RTM}^t, \Theta_{ATM}^t, D_{TM}^t, L_{TM}^t, V_{TM}^t, A_{TM}^t, W_{TM}^t, S_{LTM}^t, S_{W_{TM}}^t, A_{ATM}^t, A_{B_{TM}}^t, A_{S_{TM}}^t, G_{TM}^t\}$ represents the TeamMate vehicle’s state vector (a description is provided in Table 5) and $\mathbf{o}^{1:t}$ represents the history of unspecified raw sensor observations from the TeamMates’s sensors.



- Let $V = \{v_1, \dots, v_{n_V}\}$ denote a set of objects detected in the vicinity of the TeamMate vehicle, a belief state $p(\mathbf{X}_v^t | \mathbf{o}^{1:t})$ about the current state of each object $v \in V$, where $\mathbf{X}_v^t = \{X_v^t, Y_v^t, \Theta_v^t, V_v^t, A_v^t, W_v^t, S_{L_v}^t, S_{W_v}^t, E_v^t, C_v^t, L_v^t\}$ (a description is provided in Table 6).
- A map M that allows a reasonable reconstruction of the course of the road in the vicinity of the TeamMate vehicle.

Table 5: Description of variables for the representation of the TeamMate vehicle considered for the second cycle.

Variable	Type	Unit	Description
X_{TM}	Continuous	[m]	X-coordinate of the centre of the TeamMate vehicle in a two-dimensional spatial coordinate system relative to an origin synchronized with the map M
Y_{TM}	Continuous	[m]	Y-coordinate of the centre of the TeamMate vehicle in a two-dimensional spatial coordinate system relative to an origin synchronized with the map M
Θ_{RTM}	Continuous	[rad]	Yaw-angle relative to a global x-axis synchronized with the map M
Θ_{ATM}	Continuous	[rad]	Yaw-angle relative to the course of the road at the TeamMate's location
D_{TM}	Continuous	[m]	Lateral deviation to a reference on the road at the TeamMate's location, e.g. the centreline on a two-lane road
L_{TM}	Discrete	$\{0, \dots, [L_{TM}]\}$	The lane, the TeamMate is currently located in, e.g. fast or slow lane on a two-lane road
V_{TM}	Continuous	[m/s]	Longitudinal velocity along the heading
A_{TM}	Continuous	[m/s ²]	Longitudinal acceleration



W_{TM}	Continuous	[rad/s]	Yaw-rate
S_{LTM}	Continuous	[m]	Length (along the x-axis)
S_{WTM}	Continuous	[m]	Width (along the y-axis)
A_{ATM}	Continuous	[%]	Activation of the acceleration pedal
A_{BTM}	Continuous	[%]	Activation of the braking pedal
A_{STM}	Continuous	[rad]	Steering wheel angle
G_{TM}	Discrete	$\{0, \dots, [G_{TM}]\}$	Selected gear

Table 6: Description of variables for the representation of an object $v \in V$ in the vicinity of the TeamMate vehicle considered for the second cycle.

Variable	Type	Unit	Description
X_v	Continuous	[m]	X-coordinate of the centre of the object $v \in V$ in a two-dimensional spatial coordinate system relative to the position of the TeamMate vehicle
Y_v	Continuous	[m]	Y-coordinate of the centre of the object $v \in V$ in a two-dimensional spatial coordinate system relative to the position of the TeamMate vehicle
Θ_v	Continuous	[rad]	Yaw-angle relative to a reference axis
V_v	Continuous	[m/s]	Longitudinal velocity along the objects heading
A_v	Continuous	[m/s ²]	Longitudinal acceleration
W_v	Continuous	[rad/s]	Yaw-rate
S_{L_v}	Continuous	[m]	Length (along the x-axis)
S_{W_v}	Continuous	[m]	Width (along the y-axis)
E_v	Binary	{true,false}	Binary flag, whether the object $v \in V$ exists in the current traffic scene.
C_v	Discrete	$\{0, \dots, [C_v]\}$	Classification of the object $v \in V$, e.g. PKW, LKW, VRU, etc.



L_v Discrete $\{0, \dots, [L_v]\}$ The lane, the object $v \in V$ is currently located in, e.g. fast or slow lane on a two-lane road

As previously described in D2.2 "Sensor Platform and Models including V&V results from 1st cycle", the prediction of the spatial and temporal evolution of the traffic scene is based on so-called *Constant Turn Rate and Acceleration* (CTRA), resp. *Constant Yaw-Rate and Acceleration* (CYRA) motion models. The CYRA model is based on a state space

$$\mathbf{s}^t = (x^t, y^t, \theta^t, v^t, a^t, w^t)^T,$$

where x and y (in m) denote the spatial coordinates of the center of the vehicle, θ (in rad) denotes the yaw angle in respect to a reference axis, v (in m/s) denotes the longitudinal velocity along the heading, a (in m/s^2) denotes the longitudinal acceleration, and w (in rad/s) denotes the yaw-rate. Let Δ (in s) denote some prediction time, the state transition equation for this model is given by

$$\mathbf{s}^{t+\Delta} = f_{CYRA}(\mathbf{s}^t, \Delta) = \begin{pmatrix} x^{t+\Delta} \\ y^{t+\Delta} \\ \theta^{t+\Delta} \\ v^{t+\Delta} \\ a^t \\ w^t \end{pmatrix},$$

where

$$x^{t+\Delta} = \begin{cases} x^t + \frac{1}{w^t} \left[\frac{a^t}{w^t} (\cos \theta^{t+\Delta} - \cos \theta^t) + v^{t+\Delta} \sin \theta^{t+\Delta} - v^t \sin \theta^t \right], & w^t \neq 0 \\ x^t + \left(\frac{1}{2} a^t (\Delta)^2 + \Delta v^t \right) \cos \theta^t, & w^t = 0 \end{cases},$$

$$y^{t+\Delta} = \begin{cases} y^t + \frac{1}{w^t} \left[\frac{a^t}{w^t} (\sin \theta^{t+\Delta} - \sin \theta^t) - v^{t+\Delta} \cos \theta^{t+\Delta} + v^t \cos \theta^t \right], & w^t \neq 0 \\ y^t + \left(\frac{1}{2} a^t (\Delta)^2 + \Delta v^t \right) \sin \theta^t, & w^t = 0 \end{cases},$$



$$\theta^{t+\Delta} = \theta^t + \Delta w^t,$$

and

$$v^{t+\Delta} = v^t + \Delta a^t.$$

Let $p(\mathbf{s}_v^t | E_v^t = \text{true}, \mathbf{o}^{1:t})$ denote a current belief state for some object $v \in \mathcal{V}$, representing a six-dimensional multivariate Gaussian distribution. For simplicity of notation, we omit mentioning the conditions $E_v^t = \text{true}, \mathbf{o}^{1:t}$ in general, such that in the following, we will e.g., simply use $p(\mathbf{s}_v^t)$ instead of $p(\mathbf{s}_v^t | E_v^t = \text{true}, \mathbf{o}^{1:t})$. Given a belief state $p(\mathbf{s}_v^t)$ and using the CYRA motion-model, we obtain a prediction for a future time step $p(\mathbf{s}_v^{t+\Delta})$ by approximating

$$p(\mathbf{s}_v^{t+\Delta}) = \int f_{\text{CYRA}}(\mathbf{s}_v^t, \Delta) p(\mathbf{s}_v^t) d\mathbf{s}_v^t$$

using the technique of unscented transformation (as previously described in D2.2 “Sensor Platform and Models including V&V results from 1st cycle”).

3.4.2.2.2 Implementation

As described in D5.1 “TeamMate System Architecture incl. open API for 2nd cycle”, integration of components in the TeamMate architecture is planned on a client-server model. Each component may provide services to and require services from other components, realized by the exchange of information based on socket communication. Due to the tight coupling between the prediction of the spatial and temporal evolution of the traffic scene and online risk assessment, we currently opt to integrate both functionalities in a single component for online risk assessment. Here, we limit our explanations to the prediction of the spatial and temporal evolution of the traffic scene. For a more detailed description, we refer to D3.5 “Concepts and algorithms incl. V&V results from 2nd cycle”.



The prediction of the spatial and temporal evolution of the traffic scene requires services from the sensor and communication platform in terms of belief state about the current state of the traffic situation. Furthermore, it requires a high precision map of the road network, allowing to infer the future behaviour of traffic participants.

Provided with the most actual sensor data, the prediction of the spatial and temporal evolution of the traffic scene then works as follows:

- Each cycle (currently repeated every 50ms) begins with an initialization:
 - Assuming that the sensor and communication platform uniquely identifies each detected object $v \in \mathcal{V}$ with an ID, the prediction of the spatial and temporal evolution of the traffic scene maintains an individual DBN for each object $v \in \mathcal{V}$ to infer the belief state over the potential behaviour hypotheses of the object. The component then checks, whether any DBN can be discarded due to the corresponding object no longer being present (e.g., due to an object leaving the sensor range) or whether any new DBN must be created (e.g., due to an object entering the sensor range).
 - For each detected object, the corresponding DBN is used to update its beliefs about the most probable behaviour hypothesis, e.g., in the case of the Peter scenario, whether the object intends to stay on its own or change to an adjacent lane.
- After the initialization is complete, the algorithm attempts to sequentially predict the belief states $p(\mathcal{S}_v^{t+i\Delta} | h_{max}^t), i = 1, \dots, \eta_{max}$ for each object $v \in \mathcal{V}$. For this, a belief state $p(\mathcal{S}_v^{t+i\Delta} | h_{max}^t)$ is used to predict the next future state $p(\mathcal{S}_v^{t+(i+1)\Delta} | h_{max}^t)$. The process is repeated until either all belief states $p(\mathcal{S}_v^{t+i\Delta} | h_{max}^t), i = 1, \dots, \eta_{max}$ have been predicted, or the available time is over, and a new cycle must be started.

After the cycle is completed or has been aborted, the prediction of the spatial and temporal evolution of the traffic scene provides the most actual and



complete predicted belief states for all traffic participants as a service to other components.



3.5 E3.2 – Driving Task model

The driving task model is not a software module, and therefore it is of a different quality compared with the enablers documented in this deliverable. It has been used in the context of the Peter scenario, using the experiment conducted by ULM for the E6.1 (Interaction strategy) as the empirical basis for the construction of several models. An understanding of this experiment is therefore necessary to explain the use of the modelling approach. Consequently, the modelling approach description has been moved to D4.4 “TeamMate HMI design, implementation and V&V results from 2nd cycle”.

3.5.1 Improvements

For the prediction of the temporal and spatial evolution of the traffic scene in the 1st cycle, we assumed the yaw-rate and acceleration to be kept constant, such that $a^{t+\Delta t} = a^t$ and $w^{t+\Delta t} = w^t$ for any temporal step width Δ and number of steps η_{max} , i.e., we assumed that a traffic participant keeps the current yaw-rate and acceleration over the complete prediction horizon $\eta_{max}\Delta$. In the 2nd cycle, we addressed this severe limitation by incorporating:

- 1) Simple but computationally inexpensive driver-models to better predict the future behaviour of traffic participants for different behaviour hypothesis.
- 2) The use of Dynamic Bayesian Networks (DBN) to infer the most probable behaviour hypothesis for each traffic participant.

Concerning 1), we defined a set of behaviour hypotheses. As we currently focus on the Peter scenario, we consider two very basic but computationally inexpensive “driver-models”, corresponding to two hypothetical behaviours, lane-keeping (LK) and lane-changing (LC), that can be used to predict the



yaw-rate when following the resp. behaviour hypotheses. For this, we redefine the computation of the state $\mathbf{s}^{t+\Delta}$ as follows:

$$\mathbf{s}^{t+\Delta} = \begin{pmatrix} x^{t+\Delta} \\ y^{t+\Delta} \\ \theta^{t+\Delta} \\ v^{t+\Delta} \\ a^{t+\Delta} \\ w^{t+\Delta} \end{pmatrix} = f_{CYRA}(\mathbf{s}^t, M, \Delta),$$

with $x^{t+\Delta}$, $y^{t+\Delta}$, $\theta^{t+\Delta}$, $v^{t+\Delta}$ as provided by the original CYRA motion-model, but defining

$$a^{t+\Delta} = g_a(a^t, \Delta) = 0.95a^t$$

(as a placeholder for the utilization of more advanced models for the longitudinal control behaviour to be developed during the third cycle), and

$$w^{t+\Delta} = g_h(w^t, M, x^{t+\Delta}, y^{t+\Delta}, \theta^{t+\Delta}, \Delta), h \in \{LK, LC\}.$$

For lane-keeping behaviour, $g_{LK}(w^t, M, x^{t+\Delta}, y^{t+\Delta}, \theta^{t+\Delta}, \Delta)$, we assume that the vehicle adapts its yaw-rate in order to keep itself aligned with the course of the road, and, once the lateral deviation from the middle of its current lane exceeds a threshold of currently $\pm 0.5\text{m}$, attempts to minimize the lateral deviation. For lane-changing behaviour, $g_{LC}(w^t, M, x^{t+\Delta}, y^{t+\Delta}, \theta^{t+\Delta}, \Delta)$, we assume that the vehicle adapts its yaw-rate to minimize the angle between the current heading (in respect to the course of the road) and a target point on the middle of the adjacent lane in a distance of 50m.

Concerning 2), having a set of simple driver-models able to predict the behaviour for different behaviour hypotheses at our disposal, we can use them to maintain a belief state over the different behaviour hypotheses, to better predict the spatial and temporal evolution of the traffic scene. The basic idea is as follows: given some past belief state $p(\mathcal{S}_v^{t-\Delta})$ and present belief state $p(\mathcal{S}_v^t)$ for each object $v \in V$, provided by the sensor and communication platform, we



start from the past belief state $p(\mathcal{S}_v^{t-\Delta})$ and predict the short-term future behavior under the lane-keeping and lane-changing hypotheses. By comparing these predictions with the actual current belief state $p(\mathcal{S}_v^t)$, we can infer a posterior probability over the different hypotheses. We then use the current belief state $p(\mathcal{S}_v^t)$ to predict the long-term evolution of the state for the most probable hypothesis.

More specifically, for each $v \in V$, we maintain a simple DBN akin to a Hidden Markov Model, with a very simple transition model and a more complex observation model. Let $H_v, Val(H_v) = \{h_{LK}, h_{LC}\}$ be the hidden state, representing the considered behavior hypotheses, and let \mathbf{o}_{S_v} denote a yet to be defined observation vector. We assume a discretized time line with a time granularity of 50ms. Let $\Delta_H = \min(t - 1, 20)$, then for any number of T time slices, the DBN defines the following joint probability distribution:

$$p(H_v^{1:T}, \mathbf{o}_{S_v}^{1:T}) = p(\mathbf{o}_{S_v}^1 | H_v^1) \prod_{t=2}^T p(H_v^t | H_v^{t-1}) p(\mathbf{o}_{S_v}^t | H_v^t, \mathbf{o}_{S_v}^{t-\Delta_H}),$$

where

$$p(\mathbf{o}_{S_v}^t | H_v^t, \mathbf{o}_{S_v}^{t-\Delta_H}) = \int \int p(\mathbf{s}_*^{t-\Delta_H} | \mathbf{o}_{S_v}^{t-\Delta_H}, H_v^t) p(\mathbf{s}_*^t | \mathbf{s}_*^{t-\Delta_H}, H_v^t) p(\mathbf{o}_{S_v}^t | \mathbf{s}_*^t) d\mathbf{s}_*^{t-\Delta_H} d\mathbf{s}_*^t.$$

During runtime, at each time step t and while the object v is within the sensor range, we use the model to maintain a belief state $p(H_v^t | \mathbf{o}_{S_v}^{1:t})$. The observation model $p(\mathbf{o}_{S_v}^t | H_v^t, \mathbf{o}_{S_v}^{t-\Delta_H})$ should be understood as follows: $\mathbf{o}_{S_v}^t$, and $\mathbf{o}_{S_v}^{t-\Delta_H}$ resp., basically represent mean vector of the corresponding belief state $p(\mathcal{S}_v^t)$, resp. $p(\mathcal{S}_v^{t-\Delta_H})$, provided by the sensor and communication platform. More



specifically, let $p(\mathbf{s}_v^t) = N(\boldsymbol{\mu}_v^t, \boldsymbol{\Sigma}_v^t)$, $p(\mathbf{s}_v^{t-\Delta_H}) = N(\boldsymbol{\mu}_v^{t-\Delta_H}, \boldsymbol{\Sigma}_v^{t-\Delta_H})$, $p(\mathbf{s}_*^t) = N(\boldsymbol{\mu}_*^t, \boldsymbol{\Sigma}_*^t)$, and $p(\mathbf{s}_*^{t-\Delta_H}) = N(\boldsymbol{\mu}_*^{t-\Delta_H}, \boldsymbol{\Sigma}_*^{t-\Delta_H})$. We define

$$p(\mathbf{s}_*^{t-\Delta_H} | \mathbf{o}_{S_v}^{t-\Delta_H}, H_v^t) \propto p(\mathbf{o}_{S_v}^{t-\Delta_H} | \mathbf{s}_*^{t-\Delta_H}) p(\mathbf{s}_*^{t-\Delta_H} | H_v^t),$$

where for each $h \in H_v^t$, $p(\mathbf{s}_*^{t-\Delta_H} | h) = N(\mathbf{0}, \infty \mathbf{I})$ and $p(\mathbf{o}_{S_v}^{t-\Delta_H} | \mathbf{s}_*^{t-\Delta_H}) = N(\boldsymbol{\mu}_*^{t-\Delta_H}, \boldsymbol{\Sigma}_v^{t-\Delta_H})$. Given these, and when observing $\mathbf{o}_{S_v}^{t-\Delta_H} = \mathbf{o}_{S_v}^{t-\Delta_H}$ we have that $p(\mathbf{s}_*^{t-\Delta_H} | \mathbf{o}_{S_v}^{t-\Delta_H}, h) = p(\mathbf{s}_v^{t-\Delta_H})$. For each $h \in H_v^t$, we use the belief state $p(\mathbf{s}_*^{t-\Delta_H} | \mathbf{o}_{S_v}^{t-\Delta_H}, h)$ and the hypothesis-based vehicle-models to infer

$$p(\mathbf{s}_*^t | \mathbf{o}_{S_v}^{t-\Delta_H}, h) = \int p(\mathbf{s}_*^{t-\Delta_H} | \mathbf{o}_{S_v}^{t-\Delta_H}, H_v^t) p(\mathbf{s}_*^t | \mathbf{s}_*^{t-\Delta_H}, H_v^t) d\mathbf{s}_*^{t-\Delta_H}$$

via the technique of unscented transformation. Once again defining that $p(\mathbf{o}_{S_v}^t | \mathbf{s}_*^t) = N(\boldsymbol{\mu}_*^t, \boldsymbol{\Sigma}_v^t)$, we can derive the likelihood for $p(\mathbf{o}_{S_v}^t | H_v^t, \mathbf{o}_S^{t-\Delta_H})$ as

$$p(\mathbf{o}_{S_v}^t | H_v^t, \mathbf{o}_S^{t-\Delta_H}) = \int p(\mathbf{s}_*^t | \mathbf{o}_{S_v}^{t-\Delta_H}, H_v^t) p(\mathbf{o}_{S_v}^t | \mathbf{s}_*^t) d\mathbf{s}_*^t.$$

As such, we can use the model to maintain a belief state $p(H_v^t | \mathbf{o}_{S_v}^{1:t})$ about the probability of different behavior hypothesis of other traffic participants. From this belief state, we then obtain the most probable hypothesis $h_{max}^t = \arg \max_h p(H_v^t = h | \mathbf{o}_{S_v}^{1:t})$ and use it to predict the spatial and temporal evolution of an object $v \in \mathcal{V}$, by incrementally inferring $p(\mathbf{s}^{t+i\Delta} | h_{max}^t)$, $i = 1, \dots, \eta_{max}$, i.e., filtered predictions for the future state.

For online risk assessment, we currently use a temporal step width $\Delta = 1s$ and a maximal number of steps $\eta_{max} = 10$, resulting in an overall prediction horizon of $\eta_{max} \Delta = 10s$. Internally, we use a smaller temporal step width $\Delta_{int} = 0.05s$, and derive $\mathbf{s}^{t+\Delta}$ from \mathbf{s}^t by recursively using $\mathbf{s}^{t+i\Delta_{int}} = f_{CYRA}(\mathbf{s}^{t+(i-1)\Delta_{int}}, M, \Delta_{int})$, $i = 1, \dots, \frac{\Delta}{\Delta_{int}}$.



4 Validation of enablers

This section presents the verification and validation on component level of the enablers, i.e. how to validate that the enablers support the cooperation of the driver and the TeamMate car. In addition, different kind of results are included like pictures, sample data and, of course, quantitative results related to the described testing methodologies. Finally, conclusions are drawn from the results.



4.1 E1.1 – Driver monitoring system with driver state model for distraction and drowsiness

4.1.1 Drowsiness experiment

The experiment was conducted in the dynamic driving simulator. The aim of the test was to record data which reflect driver drowsiness and the process of getting drowsy while driving. Main part of the experiment was a two-hour drive on an empty highway in a night-time simulation. During that drive, various indicators of driver drowsiness including a Karolinska Sleepiness Scale (KSS) rating by an expert as well as camera videos were recorded. The experiment was performed with N=30 drivers from different age groups. On average, drivers started the drive on KSS level 4 and reached a maximum level of KSS 8 or KSS 9.

Table 7 shows the requirements used for the technical validation of E1.1.

Table 7: Requirements and metrics used for the technical validation of E1.1

Requirement	Metric	Success criteria
Drowsiness detection	Detection rate	>70% → acceptable >80% → good >90% → excellent

4.1.1.1 Test course

The main part of the experiment consisted of a roughly two-hour-drive on an empty highway at night. Before the driver entered the highway, he drove through a short urban section (about 5 minutes) with medium traffic, intersections, pedestrians etc. On the highway, there was only little traffic.



About every 5 minutes, the driver needed to overtake a slow vehicle on the right lane and there were repeatedly faster vehicles overtaking on the left lane. The speed limit for the driver was set to 120 km/h.

After about two hours of highway driving, the driver reached again a town and the test drive ended with roughly 5 minutes of driving through the city. The city scenario at the end was very similar to the scenario in the beginning.

On the highway, driver's state was rated regularly by the driver himself and by an expert watching the drive. For both ratings, the KSS was used in a version in which every point of the scale has a verbal anchor (Akerstedt, Anund, Axelsson & Kecklund [¹¹], 2014; see Figure 18).

About every ten minutes, the driver was shown the KSS on a display in the vehicle and he was asked to judge his own current state on the scale. The first KSS rating was given directly after entering the highway. On the highway, there were repeated ratings of the driver every ten minutes and of the expert every five minutes. Then there was a driver and expert rating directly before seeing the approaching city at the end of the drive. The last rating of driver and expert were during the urban scenario at the end.

Between the KSS-ratings, the expert annotated signs of fatigue. Different types of fatigue symptoms were counted for sections of about 5 minutes of driving. The different assessed symptoms were based on the ORD Behavior &

¹¹ Akerstedt, T., Anund, A., Axelsson, J. & Kecklund, G. (2014). Subjective sleepiness is a sensitive indicator of insufficient sleep and impaired waking function. *J Sleep Res.* 23, 240-252.



Mannerism Checklist (Wiegand, McClafferty, McDonald & Hanowski [¹²], 2009). The original ORD checklist was not used because it is too complex for a continuous online rating. Therefore, similar symptoms were combined and counted together.

4.1.1.2 Data logging

The data logging was performed by considering:

- The video of the Driver Monitoring camera which can be processed offline for the development of the driver's state model.
- A video of the cabin environment and the road scene.
- The annotation of the expert.

4.1.1.3 Study procedure

After arrival, the participant was asked to fill in a consent form in which he agreed that the data will be logged. Next, the participant was informed about the content of the study and the planned procedure and filled in a short pre-questionnaire mostly with demographic items. During the instruction, the KSS was explained to the participant and he was asked to give his first KSS rating in the pre-questionnaire.

Next, the participant was seated in the driving simulator, adjusted his seat. Before the test drive, a short reference measurement was recorded in which

¹² Wiegand, D.M., McClafferty, J., McDonald, S.E. & Hanowski, R.J. (2009). Development and Evaluation of a Naturalistic Observer Rating of Drowsiness Protocol. Report. VTTI, Blacksburg, Virginia.



the driver was asked to look at different areas in the vehicle mock-up and to make certain faces. Then, the two hours test drive was started.

4.1.1.4 Study sample

- In total, N=30 drivers took part in the experiment.
- Their age ranges between 23 and 79 years.
- There was 12 woman and 18 mans
- 9 were wearing glasses

4.1.1.5 Expert results

Figure 18 shows the change of KSS-ratings during the experiment. The expert rating at the beginning of the drive around level 4. Maximum values rated during the drive are on average on level 9 and in the urban section at the end of the drive the fatigue goes down to level 5 on average. 4 drivers suffered micro-sleeps.

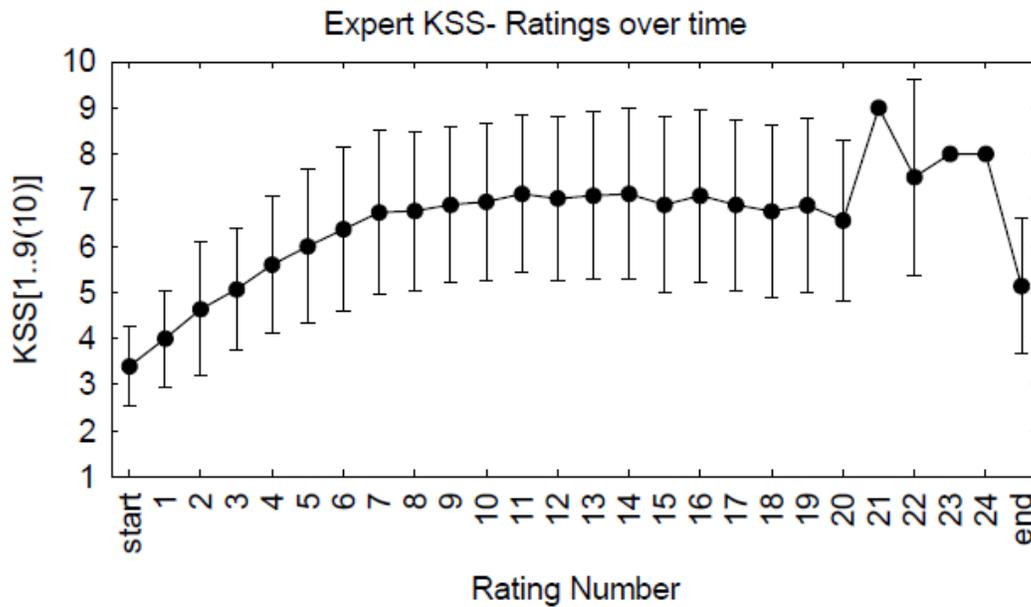


Figure 18: KSS-ratings over time

4.1.1.6 Test results

Table 8 shows the results of the evaluation done end of November on this database.



Table 8: Results of the evaluation at the end of November

KPIs	Results
Duration of database in s	170292.80
Head tracker availability ratio	0.91
Closed Eye Availability ratio	0.96
Drowsiness Diagnostic Availability ratio	1.00
Detection rate of Alert (KSS 1-5) events	45.40
Detection rate of SDrowsy (KSS 6-7) events	47.49
Detection rate of Drowsy (KSS 8) events	43.15
Detection rate of Sleepy (KSS 9) events	32.39
Detection rate of Sleeping (KSS 10) events	82.14
Detection rate of DSS (KSS 8-10) events	76.01
Number of False DSS events per hour	0.15

The results show that the model is performing when considering two states: Alert which include Alert and Slightly Drowsy and Sleepy which include Drowsy, Sleepy and Sleeping state (DSS). The achieved detection rate is 76% so above 70% which is acceptable according to the agreed Automate requirements.

A more detailed analysis of the results shows that the drowsiness model has difficulties to differentiate Alert and slightly drowsy state. This is not a big issue as drowsy state are mostly informative to the driver and don't require any intrusive warning. We noted also the same issue regarding the Drowsy and Sleepy states again this issue can be handle by the HMI as for both state the driver should be warned. We must also note that the evaluation of drowsy and sleepy state is biased by the accuracy of the expertise which has difficulties to differentiate these two states.



4.1.2 Visual Attention/Distraction experiment

The experiment was conducted in a static vehicle. The aim of the test was to record data of driver's looking at different area of the vehicle in agreement with the Automate requirements.

The experiment was performed with N=20 drivers. 10 men and 10 women; 5 wearing glasses. Drivers were asked to look at different areas of the vehicle for a defined duration.

4.1.2.1 Test protocol

The test protocol is divided in the following steps:

Step 1: Initialisation; duration: 40s

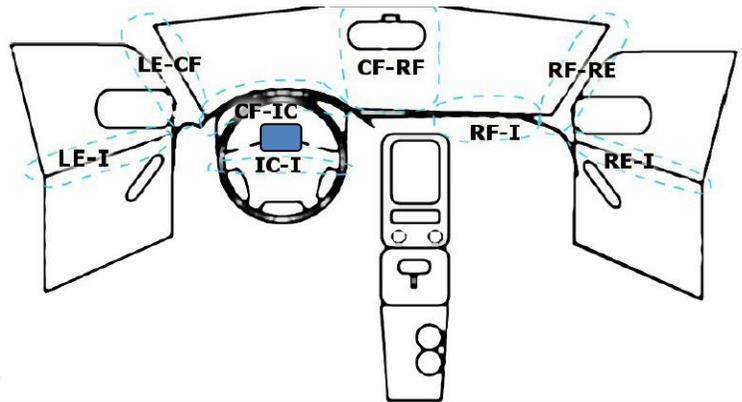
- Start Recording
- Open the door, enter the car
- Adjust seat, position, put Seatbelt ON
- Do a look around of all instruments in the car to be sure the face tracker is in place and everything works
- Focus the camera (no more talking and no head movements for the first two steps). The instructor will precise when to start moving the head.

Step 2: focus on camera; Duration: 25s

- Look at the camera
- Without moving the head do Look Around
 - Left (mirror)
 - Up (ceiling)
 - Right (mirror)

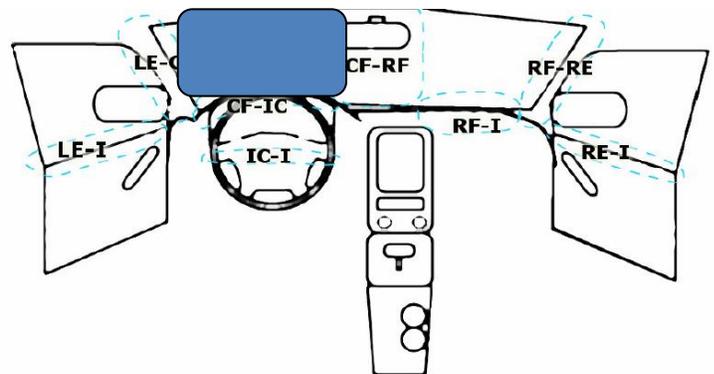


- Down (knees)
- Look at the camera
- Eye Closure
- Close your eyes
- Open your eyes
- Short blink
- Back to road (neutral position)



Step 3: focus on road; Duration: 25s

- Look at the road
- Without moving the head do Look Around
 - Left (mirror)
 - Up (ceiling)
 - Right (mirror)
 - Down (knees)
 - Look at the road
 - Eye Closure
 - Close your eyes
 - Open your eyes
 - Short blink

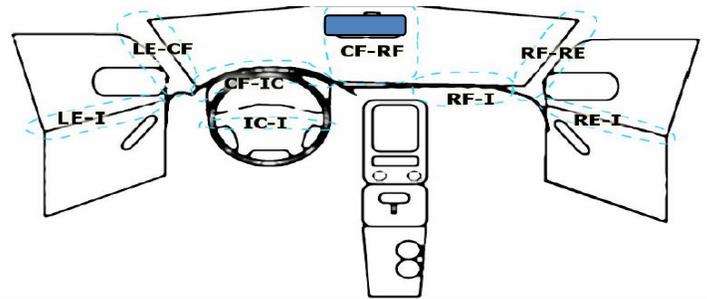


- Back to road (neutral position), You can move your head for next movements (looking at instruments)



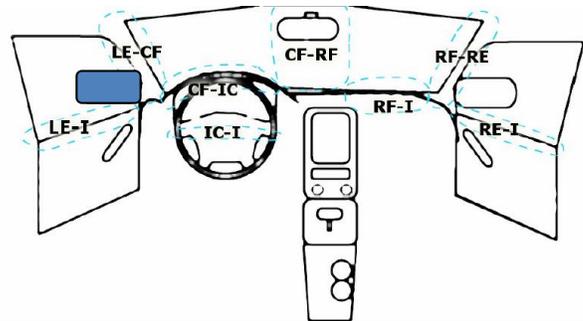
Step 4: Focus on central rear view mirror; Duration: 15s

- Look at the rear-view mirror
 - Close your eyes
 - Open your eyes
 - Short blink
- Back to road (neutral position)



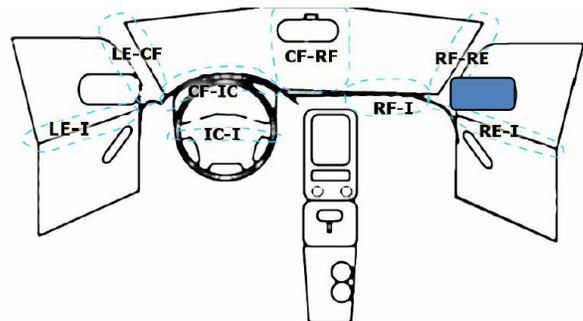
Step 5: Focus on left rear view mirror; Duration: 15s

- Look at the left-side mirror
 - Close your eyes
 - Open your eyes
 - Short blink
- Back to road (neutral position)



Step 6: Focus on right rear view mirror; Duration: 15s

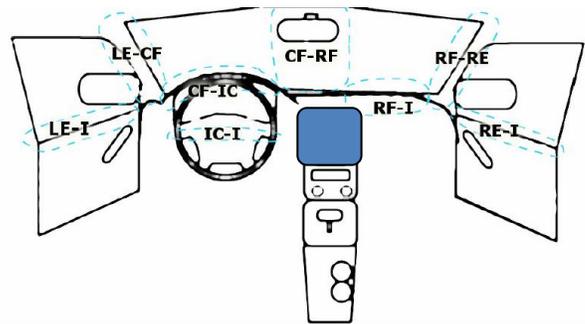
- Look at the right-side mirror
 - Close your eyes
 - Open your eyes
 - Short blink
- Back to road (neutral position)





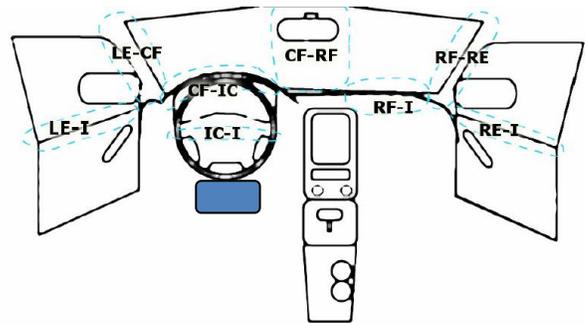
Step 7: Focus on central display; Duration: 15s

- Look at the left-side mirror
 - Close your eyes
 - Open your eyes
 - Short blink
- Back to road (neutral position)



Step 8: Focus on own lap; Duration: 15s

- Look at the left-side mirror
 - Close your eyes
 - Open your eyes
 - Short blink
- Back to road (neutral position)



4.1.2.2 Results

The accuracy analysis is still ongoing. Initial results are presented below.

The eye/head gaze out of the road (distraction) evaluation show results coherent with the driving ones:

- Smartphone recognition ratio: 90%;
- Navigation recognition ratio: 75%;
- Central mirror: 75%;
- Left mirror: 80%;
- Right mirror: 90%.

In conclusion all areas are detected with a detection rate above 75% which is already fully acceptable in regard to the automate requirements.



4.2 E1.2 – V2x communication

The aforementioned V2x messages are standardized, and will be used in different scenarios of the project. Therefore, the verification and validation of V2x component focus on these specific messages.

In this deliverable initial test cases are described, and their results are presented. In the next months, it is planned to perform field tests that model the Eva and Martha scenarios as close as possible from V2x aspect.

The initial test are split into simple test cases to build up and verify the V2x component step-by-step as described in D2.3 “Metrics and Experiments for V&V of the driver, vehicle and situation models in the 2nd cycle”, according to the requirements summarized in Table 9.

Table 9: Requirements and metrics used for the technical validation of E1.2

Requirement	Metric	Success criteria
Validation of DENM message reception	Packet loss rate	>90%
Validation of CAM message reception	Packet loss rate	>95%
Validation of CAM message reception	Jitter	<10%

The parameters of the performed laboratory test are the following:

- test duration: ~1 hour (~3600 sec);
- communicating ITS stations:
 - 1 RSU (Road Side Unit – Infrastructure);
 - 1 OBU (On-board Unit – Vehicle);
- generated and transmitted V2x messages:



- DENM (from RSU to OBU);
- CAM (from OBU to everyone, now to RSU);
- monitoring, logging method: captured packets/messages are logged in pcap files.

Table 10 contains the short test case descriptions and the results.

Table 10: Test case descriptions and related results

Test case	Description	Results
Verification of V2X communication	The communication is established between two V2x capable components. The transmitted information is received by the vehicle.	Yes
Verification of DENM message communication	The communication is established between two V2X capable components. The transmitted DENM messages from RSU is received by vehicle's OBU. The message have to be assembled properly.	Yes
Verification of CAM message communication	The communication is established between two V2X capable components. The transmitted CAM messages is received by the vehicle's OBU. The message have to be assembled properly.	Yes
Verification of DENM messages	The received DENM messages are relevant information about the traffic situation, i.e. they contain information about road works ahead.	Yes (see Figure 19)



Verification of CAM messages	The received CAM messages are relevant information about the vehicle status, i.e. they contain information about vehicle's current position, speed and heading.	Yes (see Figure 20)
Validation of DENM message reception	DENM messages are received properly during the test period. 99% percent of messages is received.	Number of transmitted and received packets: 3600 Packet loss rate: >99% Threshold values of metrics are fulfilled
Validation of CAM message reception	CAM messages are received properly during the test period. 99% percent of messages is received with low jitter.	Number of transmitted and received packets: 3600 Packet loss rate: >99% Jitter: <10% in ~95% of time Threshold values of metrics are fulfilled

The latter two test cases are related to the MARTHA and EVA scenarios, therefore, they will be extended taking into account the communication distances as requirements, and will be performed during field tests.

Figure 19 and Figure 20 present the captured DENM and CAM messages. As mentioned, the wireless transmission is logged in pcap files. These files are processed and decoded by Wireshark, which is a popular network protocol analyser tool. It also decodes messages into human readable form. The red boxes indicate the most important information contained by the DENM and



CAM messages, which are used by the TeamMate car in the scenarios. Since the tests were carried out indoor, therefore the speed values carried by the CAM messages are very low (practically it is never zero due to the inaccuracy of GPS).

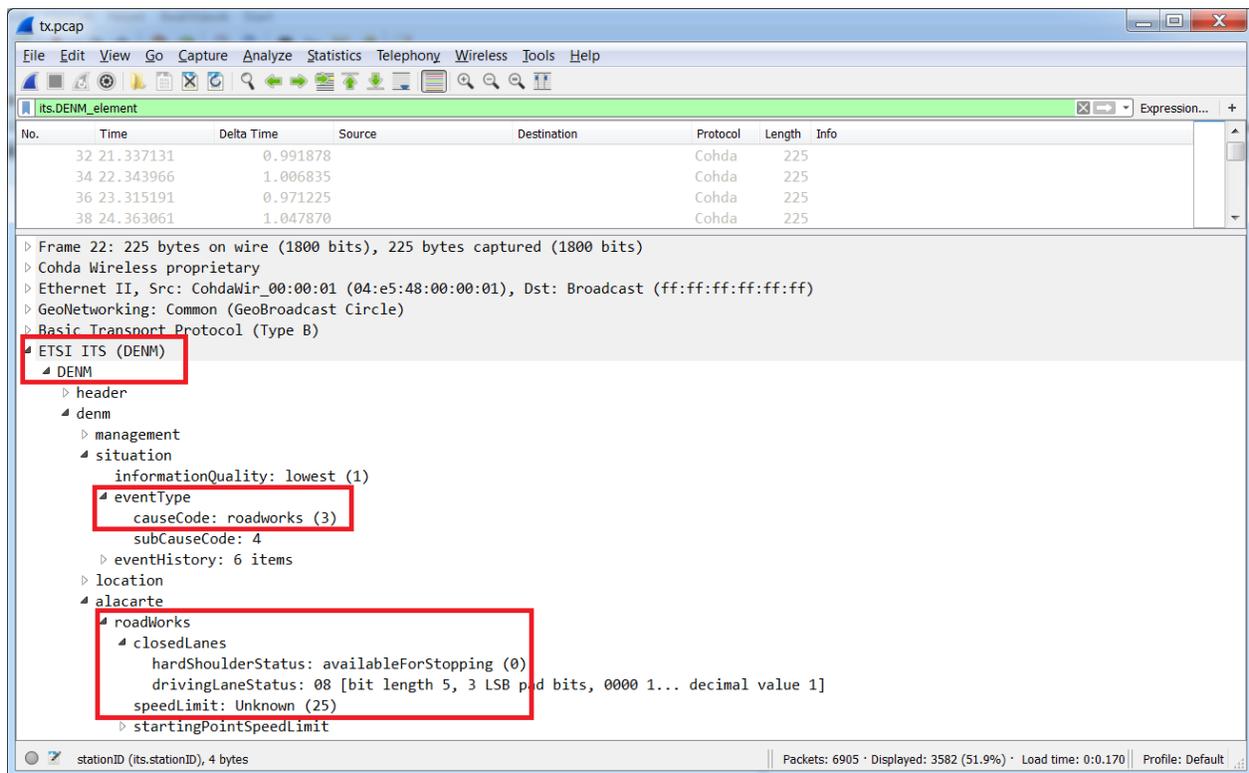
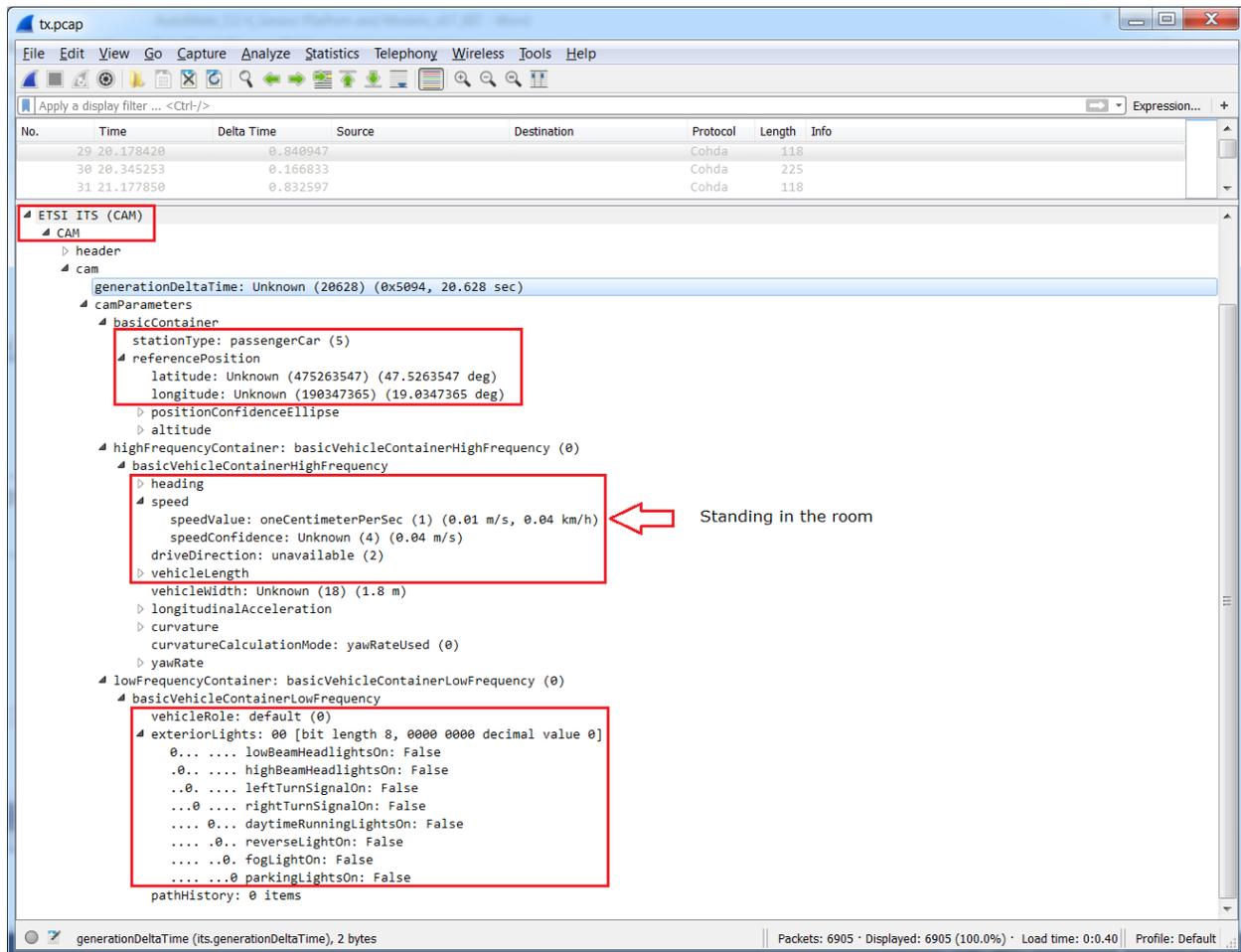


Figure 19: Captured DENM RWW message

tx.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression...

No.	Time	Delta Time	Source	Destination	Protocol	Length	Info
29	20.178420	0.848947			Cohda	118	
30	20.345253	0.166833			Cohda	225	
31	21.177850	0.832597			Cohda	118	

ETSI ITS (CAM)

- header
 - cam
 - generationDeltaTime: Unknown (20628) (0x5094, 20.628 sec)
 - camParameters
 - basicContainer
 - stationType: passengerCar (5)
 - referencePosition
 - latitude: Unknown (475263547) (47.5263547 deg)
 - longitude: Unknown (190347365) (19.0347365 deg)
 - positionConfidenceEllipse
 - altitude
 - highFrequencyContainer: basicVehicleContainerHighFrequency (0)
 - basicVehicleContainerHighFrequency
 - heading
 - speed
 - speedValue: oneCentimeterPerSec (1) (0.01 m/s, 0.04 km/h)
 - speedConfidence: Unknown (4) (0.04 m/s)
 - driveDirection: unavailable (2)
 - vehicleLength
 - vehicleWidth: Unknown (18) (1.8 m)
 - longitudinalAcceleration
 - curvature
 - curvatureCalculationMode: yawRateUsed (0)
 - yawRate
 - lowFrequencyContainer: basicVehicleContainerLowFrequency (0)
 - basicVehicleContainerLowFrequency
 - vehicleRole: default (0)
 - exteriorLights: 00 [bit length 8, 0000 0000 decimal value 0]
 - lowBeamHeadlightsOn: False
 - highBeamHeadlightsOn: False
 - leftTurnSignalOn: False
 - rightTurnSignalOn: False
 - daytimeRunningLightsOn: False
 - reverseLightOn: False
 - fogLightOn: False
 - parkingLightsOn: False
 - pathHistory: 0 items

generationDeltaTime (its.generationDeltaTime), 2 bytes

Packets: 6905 · Displayed: 6905 (100.0%) · Load time: 0:0.40 | Profile: Default

Figure 20: Captured CAM message



4.3 E2.1 – Driver Intention Recognition

Following the plans previously described in D2.3 “Metrics and Experiments for V&V of the driver, vehicle and situation models in the 2nd cycle”, validation of the driver model for intention and behaviour recognition was performed using a set of independent test data D_{Test} , representing annotated “ground-truth” time-series of manual driving on rural roads, akin to the Peter scenario.

Table 11 shows the requirements for the technical validation of E2.1.

Table 11: Requirements and metrics used for the technical validation of E2.1

| Requirement | Metric | Success criteria |
|-----------------|----------------------|------------------|
| R_EN2_model1.10 | Precision and Recall | >80% |
| R_EN2_model1.11 | Accuracy | >80% |

4.3.1 Experiments for data gathering

For gathering data for training and validation of the driver model for intention and behaviour recognition, the AutoMate partner OFFIS (OFF) conducted a data collection study for the Peter scenario in the OFF driving simulator.

4.3.1.1 Scenario

An initial rural road track (two lanes, one for each driving direction) was designed by the project partner ULM and provided to OFF. Because the objective of the study conducted by OFF was different, it was necessary to adapt this original track. The main objective here was to gather data about manual overtaking behaviour on rural roads. Therefore, it was necessary to include sufficient straight road sections which allowed subjects to accomplish



an overtaking manoeuvre. To reduce the chances of motion sickness two things were adjusted:

- The radii of most of the curves was increased to remove stronger braking actions. In these situations, drivers would expect a strong longitudinal force, which cannot be simulated in a fixed-based simulator. These situations should be avoided at least for subjects with less or no experience in driving simulators.
- The density of surrounding objects (trees, houses etc.) and also the appearance of higher mountains especially in curves was reduced. This reduces the effect that the visible environment seen on the 3 projection surfaces moves/turns around the driver which is a key problem regarding motion sickness. During the experiment only one subject reported symptoms of motion sickness.

The resulting total track length was 30.8 km with a general speed limit of 100 km/h and six curves with an 80 km/h speed limit. Participants were instructed to stop at a parking side at the end of the track. Traffic flow in both directions consisted of trucks and passenger cars with varying driving speed. The vehicles on the ego lanes were driving with an average speed of 72km/h which allowed subjects to overtake trucks as well as passenger cars. The oncoming traffic instead drove at different speeds: trucks between 65-75 km/h and passenger cars between 70-105 km/h.

Although it might not be realistic that passenger cars drive at the very same speed as trucks, the reason for this was the following: once a driver gets stuck behind a lead car, multiple vehicles are set up within a smaller area ahead which automatically leads to a small traffic jam, especially, if faster vehicles



approach a slower vehicle. Such situations sometimes lead to braking cascades and the driver might also no longer be able to overtake anymore.

4.3.1.2 Procedure

After filling the consent form and checking the validity of their driving license a short introduction to the driving simulator was given. Afterwards, each participant was trained for at least 10 minutes, but training was not stopped before they felt comfortable with the vehicle control and the overtaking scenario. No specific training scenario was implemented, instead the original experiment scenario was used. After the training session, an eye tracker was calibrated. After the training, a short verbal, scripted instruction was given.

After training and calibration, the actual experiment was started which was divided in three blocks: in each of them participants had to drive the 30.8 km track with different traffic conditions.

The following procedure was used to adapt the amount of traffic individually based on subject's overtaking behaviour: for passive drivers in the training we started with "very low traffic" ahead, whereas those drivers who easily overtook cars already in the training started with the "low traffic" condition. Based on the number of overtaking manoeuvres the traffic condition of the consecutive blocks was reduced or raised between "very low", "low", "medium", and "high" traffic conditions. In contrast, the amount of oncoming traffic was held constant in all conditions.

After the first and second block, participants had a 5-minute resting period and a re-calibration of the eye tracker was done before the next block started.



Before the last block, time pressure was imposed on the subject with the goal to foster more and probably risky overtaking manoeuvre. After the last block information about each subject's driving experience was captured using a paper form. For each subject the study took around two hours.

4.3.1.3 Materials and methods

The research-driving simulator at OFFIS is a fixed-based simulator platform, as shown in Figure 21.



Figure 21: Fixed-based simulator platform at OFF.

The basic simulator software is SILAB [¹³], which is used to create the road geometry, landscape and traffic scenario. Amongst others, it includes modules for vehicle dynamics, sensor models, data recording and also most of the scripting for empirical studies is done within the software or its extensions. The driving simulator implements a three beamer-based visualization, with a maximum field of view of 150 degree. Two displays with a resolution of 1024*768 are used for the simulation of left and right exterior mirror. Three

¹³ www.wivw.de/silab



Lexium Schneider CAN bus servo drives apply force feedback and vibration signals with adjustable amplitude and frequency on the steering wheel as well as accelerator and brake pedal. For recording the eye movements, we use an Ergoneers Dikablis Professional eye-tracker system (Full HD resolution, 60Hz tracking).

To be able to analyse the gaze behaviour of the subjects, eye-tracking was used during the study. The Dikablis eye-tracker offers the possibility to investigate areas of interest (AOI'S) within the recorded scene, based on the detection of visual markers. For this purpose, markers were mounted next to each of the three rear view mirrors (left, right and interior). Four additional markers were located at the corners of the frontal projection screen and one more marker was projected onto the back of each truck which had to be overtaken by the participants. Specialized procedures based on displaying sets of fixation points were applied for the calibration and measurement of the eye tracking quality.

4.3.1.4 Participants

Subjects were acquired by announcements at the local university and were required to be licensed drivers with at least 2000 km/year driving experience. All subjects received a compensation of 10 EUR/hour for their participation. In total 18 subjects participated the study, one of them experienced motion sickness in the very beginning. Thus, we ended up with valid data from 17 subjects. Subjects were 28 years old (SD=7.1) on average (9 male and 8 female). Participants were licensed on average for 10 years (SD=6.6), and drove 16941 km/year on average (SD=13961).



4.3.2 Data Preparation

During each trial, with a frequency of 50 Hz, data samples containing simulation and eye-tracking information was recorded, where the simulation data was comprised (among other information) of the state of the participant's vehicle (in the following denoted as the TeamMate vehicle), and the states of up to twelve vehicles in the vicinity of the TeamMate vehicle.

During post-processing, using an editor to visualize the recorded data, each data sample was first *manually* annotated with the shown driving behaviour (LCL, LCR, or LK) based on visual judgement of the traffic situation. Afterwards, each sample was *automatically* annotated with whether the driver intended to drive on the right or on the left lane, based on an arbitrary but conservative rule that *a change in the target lane intention is assumed to be formed at least up to one second prior to the annotated beginning of a lane change manoeuvre.*

During post-processing, the trials of a single participant had to be removed due to data inconsistencies. As such, the result of the annotation process was a set of 48 time-series of multivariate data for manual driving in the Peter scenario. From this annotated experimental data, the first and third trial of each participant were selected as a training set D_{Train} , consisting of 2234536 samples or approx. 620 minutes, while the remaining trials, consisting of 1218784 samples or approx. 338 minutes were reserved as a test set D_{Test} for validation.



4.3.3 Validation process

To recapitalize the overall validation process and metrics used for this validation cycle, let D_{Test} be composed by a number of m trials, where each trial is a time-series consisting of a number of $n_j, j = 1, \dots, m$ data samples $d_j^k = (l_j^k, b_j^k, l_j^k, \mathbf{o}_{I_j^k}, \mathbf{o}_{B_j^k}), k = 1, \dots, n_j$, annotated by experts with the assumed correct intention i_j^k and behaviour b_j^k (c.f., Section 4.3.2).

In this cycle, we focused on the component for intention recognition and neglected the influence of the component for behaviour recognition. As such, for each sample d_j^k , we used the different realizations of the driver model, M_{Gauss} , M_{GMM} , and M_{LR} , to infer a filtered belief state over the intentions $P(I_j^k | l_j^{1:k}, \mathbf{o}_{I_j^{1:k}})$, given all available sensory input in the resp. time-series up to the sample. The output of the model was then defined as the most probable target lane intention

$$i_{j,out}^k = \arg \max_i P(I_j^k = i | l_j^{1:k}, \mathbf{o}_{I_j^{1:k}}).$$

For the assessment of intention recognition, the (annotated) “true” and predicted target lane intentions i_j^k and $i_{j,out}^k$ were first mapped onto actual lane change intentions (in that a lane change intention is present if the current lane and the target lane intentions differ) by defining $\hat{l}_j^k = \mathbf{1}(l_j^k \neq i_j^k)$ and $\hat{l}_{j,out}^k = \mathbf{1}(l_j^k \neq i_{j,out}^k)$, where $\mathbf{1}$ denotes the indicator function. Interpreting the existence (we note that the ground truth is based on a manual annotation of the test data and therefore subject to error) of a lane change intention as positive and the absence as negative, we constructed a binary confusion matrix for each model, as shown in Figure 22.



| | | Predicted | |
|--------------|----------|-----------|----------|
| | | Positive | Negative |
| Ground truth | Positive | TP | FN |
| | Negative | FP | TN |

Figure 22: Binary confusion matrix.

The actual metric proposed in D2.3 “Metrics and Experiments for V&V of the driver, vehicle and situation models in the 2nd cycle” is called the accuracy and defined as:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

However, as the accuracy proved to be a poor metric for assessing the performance of the model, we additionally introduce the following, more meaningful, metrics:

- The *precision*, representing the fraction of correctly recognized intentions among all predicted intentions, defined as

$$Precision = \frac{TP}{TP + FP}$$

A high precision indicates that the model only recognizes intentions if there actually exists an intention.

- The *recall* (also known as sensitivity or true positive rate (TPR)), representing the fraction of correctly recognized intentions over the total amount of true intentions, defined as

$$Recall = \frac{TP}{TP + FN}$$

A high recall indicates that most of the intentions are recognized as such.



- The harmonic mean of precision and recall, the traditional F-measure or balanced *F-score*, defined as

$$F\text{-score} = 2 \frac{\textit{Precision} \cdot \textit{Recall}}{\textit{Precision} + \textit{Recall}}$$

- And, for the sake of completeness, the *False Positive Rate* (FPR), defined as

$$FPR = \frac{FP}{FP + TN}$$

4.3.4 Datasets used

The test set D_{Test} was obtained from the experimental data conducted for training and evaluation of the probabilistic driver models (c.f. Section 4.3.2). More specifically, the experimental data was split into a training set D_{Train} , including approx. 70% of the experimental data (2234536 samples or approx. 620 minutes), and a test set D_{Test} , including the remaining experimental data (1218784 samples or approx. 338 minutes).

The different realizations of the component for intention recognition M_{Gauss} , M_{GMM} , and M_{LR} have been learned exclusively using the training data D_{Train} . The resulting models were then subsequently validated on the test set D_{Test} . Being temporal models, intended to provide their output each 50ms, we only use every third sample for the actual validation, resulting in an effective test set D_{Test} , consisting of 406267 samples, covering 338 minutes of driving over each one trial for each of the 16 participants.

4.3.5 Results

Figure 23 shows the results in terms of a binary confusion matrix and corresponding metrics for the model M_{Gauss} , Figure 24 shows the results for the



model M_{GMM} , and Figure 25 shows the results for the model M_{LR} . In all cases, to allow a better interpretation of the results, values in brackets denote the corresponding values, if we limit the focus on the case, where the driver was located on the right lane, therefore allowing the interpretation of intentions purely as overtaking intentions.

| | | Predicted Intention | | Total | Metric | Value |
|---------------------|----------|-----------------------|-----------------------|----------------------|---------------|---------------|
| | | Positive | Negative | | | |
| „True“
Intention | Positive | TP=28248
(16816) | FN=3418
(800) | 31666
(17616) | Accuracy | 0.543 (0.524) |
| | Negative | FP=182198
(181692) | TN=192403
(183918) | 374601
(365610) | Precision | 0.134 (0.085) |
| Total | | 210446
(198508) | 195821
(184718) | N=406267
(383226) | Recall / TPR | 0.892 (0.955) |
| | | | | | F-score | 0.233 (0.156) |
| | | | | | FPR | 0.486 (0.497) |

Figure 23: Confusion matrix and corresponding metrics of interest for the model

M_{Gauss}

| | | Predicted Intention | | Total | Metric | Value |
|---------------------|----------|---------------------|-----------------------|----------------------|---------------|---------------|
| | | Positive | Negative | | | |
| „True“
Intention | Positive | TP=25567
(14343) | FN=6099
(3273) | 31666
(17616) | Accuracy | 0.787 (0.783) |
| | Negative | FP=80332
(79941) | TN=294269
(285669) | 374601
(365610) | Precision | 0.241 (0.152) |
| Total | | 105899
(94284) | 300368
(288942) | N=406267
(383226) | Recall / TPR | 0.807 (0.814) |
| | | | | | F-score | 0.372 (0.256) |
| | | | | | FPR | 0.214 (0.219) |

Figure 24: Confusion matrix and corresponding metrics of interest for the model

M_{GMM}

| | | Predicted Intention | | Total | Metric | Value |
|---------------------|----------|---------------------|-----------------------|----------------------|---------------|---------------|
| | | Positive | Negative | | | |
| „True“
Intention | Positive | TP=11666
(5) | FN=20000
(17611) | 31666
(17616) | Accuracy | 0.945 (0.951) |
| | Negative | FP=2290
(1247) | TN=372311
(364363) | 374601
(365610) | Precision | 0.836 (0.004) |
| Total | | 13956
(1252) | 392311
(381974) | N=406267
(383226) | Recall / TPR | 0.368 (0.000) |
| | | | | | F-score | 0.511 (0.000) |
| | | | | | FPR | 0.006 (0.003) |

Figure 25: Confusion matrix and corresponding metrics of interest for the model

M_{LR}



Focusing on the results for M_{Gauss} (Figure 23), we see that the model has a rather poor accuracy of 0.543. On a positive note, it was able to correctly recognize 28248 of 31666 samples annotated with a lane change intention, resulting in a high recall of 0.892. However, this comes with the drawback of 182198 wrongly recognizing intentions, resulting in an abysmal precision of only 0.134, and consequently an F-score of only 0.233.

As apparent for M_{GMM} (Figure 24), allowing for a more complex representation via the use of GMMs instead of simple Gaussian, we are able to improve the accuracy to 0.787. On a positive note, the model is able to improve the precision to (a still low) 0.241, while only reducing the recall to 0.807, resulting in a slightly improved F-score of 0.372.

For obvious reasons, these results leave much room for improvement. We note, however, that many of the false positives, resulting in low precisions and consequently F-scores, result from the driver “checking” for an opportunity to overtake (which would imply an intention to overtake in the classical sense) but not doing so, even if such an opportunity exists. We will tackle to incorporate this knowledge into the models for driver intention recognition during the 3rd cycle.

Lastly, when considering the overall results, M_{LR} has the highest accuracy (0.945) and precision (0.836), and, despite a low recall of 0.368, the highest F-score of 0.511. Using the accuracy as a metric for performance, M_{LR} would seem like the most successful candidate for driver intention recognition. However, when considering only the cases, where the TeamMate vehicle is located on the right lane, we see a dramatic drop in both precision and recall, and consequently, in the F-score. This can be explained by the fact, that M_{LR} basically classifies every situation on the right lane as resulting in an absence of an overtaking intention (For comparison, a model that simply classifies



every sample as negative would achieve an accuracy of 0.922). As such, despite the highest accuracy, M_{LR} must be seen as the worst candidate for intention recognition. For the moment, we will consider M_{GMM} as our starting point for the third cycle.

Enhancing the model incorporating the influence of the component for behaviour, i.e., using $P(I_j^k | l_j^{1:k}, \mathbf{o}_{I_j}^{1:k}, \mathbf{o}_{B_j}^{1:k}) = \sum_{b \in B} P(I_j^k, b^k | l_j^{1:k}, \mathbf{o}_{I_j}^{1:k}, \mathbf{o}_{B_j}^{1:k})$ instead of $P(I_j^k | l_j^{1:k}, \mathbf{o}_{I_j}^{1:k})$, we are able to slightly improve on the overall performance, as exemplary shown in Figure 26 for M_{GMM} . A preliminary inspection of the test data implies that the remaining false positives may result from a misclassification of the human behaviour to “peek out” behind the lead vehicle by performing half of a lane change as lane changes to the left lane.

| | | Predicted Intention | | Total | Metric | |
|---------------------|----------|---------------------|-----------------------|----------------------|--------------|---------------|
| | | Positive | Negative | | Value | |
| „True“
Intention | Positive | TP=26198
(14674) | FN=5468
(2942) | 31666
(17616) | Accuracy | 0.841 (0.840) |
| | Negative | FP=59057
(58527) | TN=315544
(307083) | 374601
(365610) | Precision | 0.307 (0.200) |
| Total | | 85255
(73201) | 321012
(310025) | N=406267
(383226) | Recall / TPR | 0.827 (0.833) |
| | | | | | F-score | 0.448 (0.323) |
| | | | | | FPR | 0.158 (0.160) |

Figure 26: Confusion matrix and corresponding metrics of interest for the model M_{GMM} .

In the next cycle, we will focus on enhancing the internal structure and provided input to improve on the precision of the models.



4.4 E3.1 – Situation and vehicle model

Table 12 shows the requirements for the technical validation of E3.1.

Table 12: Requirements and metrics used for the technical validation of E3.1

| Requirement | Metric | Success criteria |
|----------------|------------------------------|----------------------------|
| R_EN3_model1.6 | Functional test | Functional test successful |
| R_EN3_model1.7 | Ratio of correct predictions | >90% |

4.4.1 Semantic enrichment of the situation model

For validating this module, we used the F_1 -score $F_1 = \frac{2}{\frac{1}{recall} + \frac{1}{precision}}$, where $recall = \frac{TP}{TP+FN}$ and $precision = \frac{TP}{TP+FP}$. TP , FP and FN are the true positive, false positive and the false negative ratios. In Figure 27, the system predicts that the ego-vehicle (red vehicle) will stop in the near future because of the red traffic light. According to the ground truth the ego-vehicle stops in the next 2 seconds. The prediction is therefore a true positive for the stop maneuver and a true negative for all other maneuvers. The predicted maneuver for the blue vehicle is to enter the intersection, due to the green traffic light. According to the ground truth this vehicle stops in 2 seconds. Hence the predicted maneuver is a false positive for the stop maneuver, a false negative for the driving maneuver and a true negative for all other maneuver.

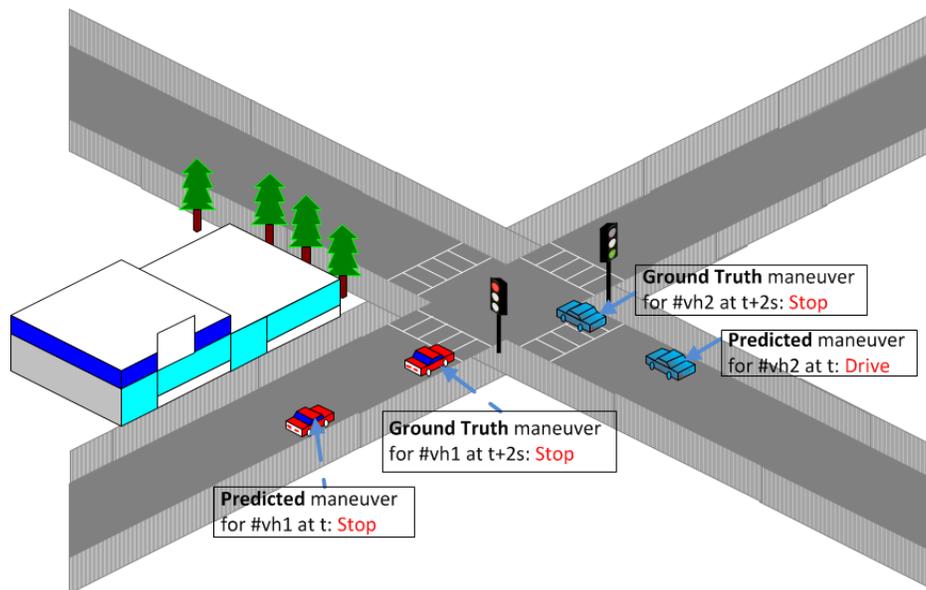


Figure 27: Example of predicted and ground truth vehicle maneuvers at an intersection

For estimating the F_1 -score, ground truth data from scene objects and manoeuvres must be generated. In this project we focus on the following scenarios:

1. Lane Following without intersection,
2. Intersection without traffic light and signal (priority to the right),
3. Intersection with traffic signal,
4. Intersection with traffic light,
5. Intersection with traffic light and signal.

For a first validation, the scenarios mentioned above will be first generated in the simulation. The final validation will be done on real traffic data. The experiment for generating the test data is described in detail in Deliverable D3.5 "Concepts and algorithms incl. V&V results from 2nd cycle". The final validation result will be annexed to D2.5.



4.4.2 Predicting the future evolution of the traffic scene

Following the plans previously described in D2.3 “Metrics and Experiments for V&V of the driver, vehicle and situation models in the 2nd cycle”, validation of the prediction of the temporal and spatial evolution of the traffic scene was performed using a set of independent test data D_{Test} , representing ground truth time-series of traffic situations.

To recapitalize and concretize the overall validation process and metrics used, let D_{Test} be composed by a number of m trials, where each trial $j, j = 1, \dots, m$ is a time-series consisting of a number of n_j , data samples $d_j^k = (\mathbf{x}_{TM}^k, \mathbf{x}_{v_1}^k, \dots, \mathbf{x}_{v_{n_j}}^k)$, $k = 1, \dots, n_j$ and a map M . For each sample d_j^k , and each object $v \in V$, we infer the most probable behaviour hypothesis h_{max} and predict a sequence of future states $p(\mathbf{s}_{j,v}^{k+i\Delta} | h_{max_{j,v}}^k)$, $i = 1, \dots, \eta_{max}$.

Concerning the validation of the prediction of the evolution of the traffic situation, it is most important that the predicted regions encompass the true future location of the vehicle. As a metric to validate the performance, we therefore choose the concept of a “correct classification rate” as the ratio of correct predictions and the number of total predictions. More specifically, let $0 < \delta < 1$ denote an arbitrary threshold, we can define a region that covers an area for which the probability of any state $\mathbf{s}_{j,v}^{k+i\Delta t}$ is above δ , i.e., $p(\mathbf{s}_{j,v}^{k+i\Delta t} | h_{max_{j,v}}^k) < \delta$. For each predicted state $p(\mathbf{s}_{j,v}^{k+i\Delta t} | h_{max_{j,v}}^k)$, $i = 1, \dots, n$, we then check whether the true state $\mathbf{s}_{j,v}^{k+i\Delta t}$ of object $v \in V$ has a probability $p(\mathbf{s}_{j,v}^{k+i\Delta t} | h_{max_{j,v}}^k) < \delta$. Denoting such an occurrence as a failure and resp. as a success otherwise, we used the metric

$$CR_{\delta}^i = \frac{\#_s}{\#_s + \#_f}$$



representing the *ratio of successes* $\#_s$ and the *sum of successes* $\#_s$ and *failures* $\#_f$ for a temporal prediction horizon $i \Delta$ and a specific level of δ for assessing the quality of the prediction of the temporal and spatial evolution of the traffic scene. For the actual validation, we abstract from the current velocity, acceleration, and yaw-rate, which are not used for online risk assessment, and instead focus on the valid prediction of the location and pose $p(x_{j,v}^{k+i\Delta t}, y_{j,v}^{k+i\Delta t}, \theta_{j,v}^{k+i\Delta t} | E_{j,v}^k = true, \mathbf{o}_j^{1:k}, h_{\max_{j,v}}^k)$ and, for the sake of comparison, on the valid prediction of the location $p(x_{j,v}^{k+i\Delta t}, y_{j,v}^{k+i\Delta t} | E_{j,v}^k = true, \mathbf{o}_j^{1:k}, h_{\max_{j,v}}^k)$.

The metric is used to assess the fulfilment of requirements R_EN3_model1.6 and R_EN3_model1.7, stating that the “integrated model must predict possible evolutions of the traffic situation in respect to potential interventions of the driver” (R_EN3_model1.6), resp. “[...] potential interventions of the automation” (R_EN3_model1.7) with a correct rate of the prediction above 90% to be fulfilled. For the second cycle, we abstracted from the potential interventions of the driver and automation. For a perfect prediction and a region that encloses $100(1.0 - \delta)\%$ of the probability mass, we would, in the perfect case, expect a failure-rate of $100\delta\%$. As such, we will treat the requirements as fulfilled, if the ratio of correct predictions is above $90(1.0 - \delta)\%$ for each prediction horizon $i \Delta$ and level δ independently.

4.4.2.1 Dataset for validation

The test set D_{Test} was obtained from the simulator study conducted for training and evaluation of the probabilistic driver models (see Section 4.3.2).

The scenario comprised approx. 30.8 km of a rural road track inspired by the Peter scenario with one lane for each direction and a primary speed limit of 100 km/h. Traffic flow in both directions consisted of trucks and passenger



cars with varying driving speed. The vehicle on the right lane were driving with an average speed of 72km/h. The oncoming traffic instead drove at different speeds: trucks between 65-75 km/h and passenger cars between 70-105 km/h.

Participants had to manually traverse the scenario by controlling a simulated vehicle until reaching a parking side at the end of the track. Participants were instructed to follow the traffic rules, but were free to overtake lead vehicles, if deemed necessary or desired. Each participant had to absolve a total of three trials, with the overall traffic conditions adapted between trials to encourage overtaking manoeuvres.

In total, 18 subjects participated the study, one of them experienced motion sickness in the very beginning. Thus, we ended up with valid data from 17 subjects. Subjects were 28 years old (SD= 7,1) on average (9 male and 8 female). Participants were licensed on average for 10 years (SD=6,6), and drove 16941km/year on average (SD= 13961). Post-analysis of the data resulted in the exclusion of the data of another participant, due to inconsistencies in the data recording.

The first and third trial of each participant was provided as experimental data for training of the models for intention recognition, while the second trial of each participant was reserved as a general test set D_{Test} for validation of the models for intention recognition, the prediction of the temporal and spatial evolution of the traffic scene, and online risk assessment.

The experimental data was recorded with a frequency of 60Hz. As the prediction of the temporal and spatial evolution of the traffic scene internally works with a frequency of 20Hz, we treat each time-series as information sequentially provided by the sensor and communication platform, and only use every third sample for the actual validation. As such, the test set D_{Test}



effectively consists of 406262 samples used for validation, covering approx. 338 minutes of simulated driving behaviour in the Peter scenario.

Due to the test set arising from a simulator study in which the traffic flow was automatically controlled by a traffic simulation, the resulting behaviour of traffic participants in the vicinity of the TeamMate vehicle is highly predictable and therefore potentially unrealistic. As a means to provide a more realistic assessment for humanly controlled traffic participants, we additionally perform our validation on the prediction of the temporal and spatial evolution of the humanly controlled "TeamMate" vehicle.

4.4.2.2 Results

We performed the validation for five different levels of δ , $\delta_{0.5} = 0.5$, $\delta_{0.25} = 0.25$, $\delta_{0.1} = 0.1$, $\delta_{0.05} = 0.05$, and $\delta_{0.01} = 0.01$. We report both the results focusing on location and pose, and focusing solely on location. To provide a more intuitive understanding of the results, we additionally report the average two-dimensional Euclidean distance (AED) between the predicted location and the ground truth.

As D_{Test} provided ground-truth data, we need to transform the ground-truth data into belief states. Let $s_{j,v}^k = (x_{j,v}^k, y_{j,v}^k, \theta_{j,v}^k, v_{j,v}^k, a_{j,v}^k, w_{j,v}^k)$ denote the ground truth of the state of a vehicle $v \in V$ in a sample d_j^k , we use the following belief state as a simulated provision of the sensor and communication platform:

$$p(S_{j,v}^k) = N \left(\boldsymbol{\mu} = \begin{pmatrix} x_{j,v}^k \\ y_{j,v}^k \\ \theta_{j,v}^k \\ v_{j,v}^k \\ a_{j,v}^k \\ w_{j,v}^k \end{pmatrix}, \boldsymbol{\Sigma} = \begin{pmatrix} 0.1 & & & & & \\ & 0.1 & & & & \\ & & 0.1 & & & \\ & & & 1.0 & & \\ & & & & 0.01 & \\ & & & & & 0.01 \end{pmatrix}^2 \mathbf{I}_6 \right).$$



Furthermore, to counterbalance our strong assumptions concerning the selected yaw-rate, after each prediction step, we inflated the resulting covariance matrix in the following way:

$$\Sigma^* = \Sigma + \Delta \begin{pmatrix} 0.1 \\ 0.1 \\ 0.01 \\ 1.0 \\ 0.1 \\ 0.00245 \end{pmatrix}^2 I_6$$

Table 13 shows the result for surrounding traffic participants. As apparent, the correct classification rate is for the most part above the corresponding level of δ , therefore fulfilling the requirements of being above $90(1.0 - \delta)\%$. Although a promising result, this mainly results from the behavior of the automatically controlled vehicles being highly predictable.

Table 13: Ratio of successes $\#_s$ and the sum of successes $\#_s$ and failures $\#_f$ for the prediction of the temporal and spatial evolution of the automatically controlled surrounding traffic participants, for different prediction horizons i (in seconds) and different levels of δ . Ratios limited to the location are shown in brackets, AED denotes the average Euclidean distance. Bold values denote that the result is above the required $90(1.0 - \delta)\%$.

| i | $\#_s + \#_f$ | $CR_{\delta_{0.5}}^i$ | $CR_{\delta_{0.25}}^i$ | $CR_{\delta_{0.1}}^i$ | $CR_{\delta_{0.05}}^i$ | $CR_{\delta_{0.05}}^i$ | AED |
|-----|---------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|--------|
| 1 | 974180 | 0.9226
(0.9033) | 0.9494
(0.9382) | 0.9630
(0.9577) | 0.9693
(0.9676) | 0.9776
(0.9846) | 0.0914 |
| 2 | 947303 | 0.9455
(0.9272) | 0.9690
(0.9602) | 0.9782
(0.9755) | 0.9804
(0.9795) | 0.9829
(0.9825) | 0.2626 |
| 3 | 920951 | 0.9405
(0.9181) | 0.9660
(0.9575) | 0.9761
(0.9732) | 0.9791
(0.9776) | 0.9823
(0.9817) | 0.5053 |
| 4 | 895087 | 0.9319
(0.9030) | 0.9641
(0.9525) | 0.9752
(0.9720) | 0.9785
(0.9768) | 0.9823
(0.9815) | 0.7872 |



| | | | | | | | |
|----|--------|----------------------------------|----------------------------------|----------------------------------|----------------------------------|----------------------------------|--------|
| 5 | 869836 | 0.9273
(0.8819) | 0.9611
(0.9469) | 0.9752
(0.9707) | 0.9788
(0.9770) | 0.9826
(0.9818) | 1.0749 |
| 6 | 845064 | 0.9224
(0.8651) | 0.9596
(0.9433) | 0.9766
(0.9704) | 0.9801
(0.9782) | 0.9840
(0.9830) | 1.3729 |
| 7 | 820691 | 0.9194
(0.8541) | 0.9611
(0.9411) | 0.9783
(0.9732) | 0.9819
(0.9797) | 0.9854
(0.9847) | 1.7065 |
| 8 | 796966 | 0.9181
(0.8490) | 0.9659
(0.9408) | 0.9795
(0.9761) | 0.9828
(0.9812) | 0.9863
(0.9857) | 2.0821 |
| 9 | 773688 | 0.9175
(0.8435) | 0.9679
(0.9441) | 0.9803
(0.9772) | 0.9833
(0.9822) | 0.9865
(0.9862) | 2.4833 |
| 10 | 750899 | 0.9186
(0.8385) | 0.9683
(0.9467) | 0.9805
(0.9776) | 0.9834
(0.9824) | 0.9864
(0.9863) | 2.8966 |

To provide a potentially more accurate picture, Table 14 shows the result for the humanly controlled TeamMate vehicle itself. In contrast to the automatically controlled vehicles, we see that the correct classification rate only completely satisfies for $\delta = 0.5$. As such, the requirements of $CR_{\delta}^i > 90(1.0 - \delta)\%$ is only fulfilled up to $\eta_{max} \Delta = 4s$ for $\delta_{0.01} = 0.01$ for location and orientation and only fulfilled up to $\eta_{max} \Delta = 3s$ for $\delta_{0.05} = 0.05$ when considering the location in isolation. An inspection of the *AED* implies the major issue being a poor prediction of the speed, resp. acceleration, resulting in a poor prediction of the travelled distance for higher prediction horizons. We will address this issue in the third cycle, by the incorporation of driver-models for longitudinal control.



Table 14: Ratio of successes $\#_s$ and the sum of successes $\#_s$ and failures $\#_f$.

| i | $\#_s + \#_f$ | $CR_{\delta_{0.5}}^i$ | $CR_{\delta_{0.25}}^i$ | $CR_{\delta_{0.1}}^i$ | $CR_{\delta_{0.05}}^i$ | $CR_{\delta_{0.01}}^i$ | AED |
|----------|---------------------------------|---|--|---|--|--|------------|
| 1 | 405942 | 0.8610
(0.8870) | 0.9146
(0.9515) | 0.9458
(0.9748) | 0.9594
(0.9835) | 0.9754
(0.9910) | 0.1002 |
| 2 | 405622 | 0.7907
(0.7622) | 0.8562
(0.8483) | 0.8946
(0.8950) | 0.9122
(0.9150) | 0.9361
(0.9422) | 0.4582 |
| 3 | 405302 | 0.7298
(0.6873) | 0.8126
(0.7910) | 0.8631
(0.8554) | 0.8840
(0.8808) | 0.9125
(0.9137) | 1.1032 |
| 4 | 404982 | 0.6909
(0.6299) | 0.7790
(0.7498) | 0.8360
(0.8218) | 0.8594
(0.8533) | 0.8928
(0.8916) | 2.0307 |
| 5 | 404662 | 0.6616
(0.5875) | 0.7564
(0.7229) | 0.8122
(0.7962) | 0.8381
(0.8283) | 0.8756
(0.8720) | 3.2034 |
| 6 | 404342 | 0.6410
(0.5572) | 0.7397
(0.7012) | 0.7960
(0.7791) | 0.8226
(0.8109) | 0.8610
(0.8559) | 4.5836 |
| 7 | 404022 | 0.6266
(0.5378) | 0.7298
(0.6879) | 0.7869
(0.7669) | 0.8121
(0.7995) | 0.8489
(0.8432) | 6.1402 |
| 8 | 403702 | 0.6193
(0.5261) | 0.7245
(0.6800) | 0.7811
(0.7599) | 0.8052
(0.7921) | 0.8397
(0.8341) | 7.8474 |
| 9 | 403382 | 0.6133
(0.5182) | 0.7193
(0.6722) | 0.7747
(0.7533) | 0.7985
(0.7844) | 0.8314
(0.8260) | 9.6818 |
| 10 | 403062 | 0.6050
(0.5107) | 0.7123
(0.6633) | 0.7675
(0.7455) | 0.7922
(0.7771) | 0.8243
(0.8190) | 11.6285 |



5 Conclusions and Outlook

This document presented all the results related to the sensor platform and models including the verification and validation results of task 2.2 to task 2.5 from the 2nd cycle.

During the first cycle initial models were developed, their performance were validated and limitations were collected. During this cycle the enablers were improved to extend the known limitations and be able to meet the requirements of the defined use cases. Then, these enhanced models were evaluated as well in different ways. The results show that all the enablers are able to provide useful data for other components of the TeamMate car allowing for the unique feature set of it.

In this document, we also highlighted the scenarios, in which the enablers play crucial roles. That is important to show how the enablers contribute to achieve the goals of the project.

In the next steps the emphasis will be on integration, i.e. to assemble the components into the demonstrators, and then to evaluate the performance of the demonstrators against their baseline to quantify the benefit and progresses of the TeamMate car approach.

Finally, the results of this V&V activity and the results of the next integration cycle allow us to draw the conclusions and find the proper ways to further improve the enablers during the 3rd cycle.