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

| | |
|---|---|
| **Project Number:** | 690705 |
| **Classification**<br>**Deliverable No.:** | Public |
| **Work Package(s):**<br>**Document Version:**<br>**Issue Date:**<br>**Document Timescale:**<br>Start of the Document:<br>Final version due: | WP2<br>Vs. 1.0<br>30.06.2017<br>Project Start Date: September 1, 2016<br>Month 7<br>Month 10 |
| **Compiled by:** | A. Giralt , CAF |
| **Authors:** | A. Giralt, CAF<br>M. Eilers, HMT<br>P. P. Fouopi, DLR<br>F. Tango, CRF<br>M. Graf, ULM<br>D. Kaethner, DLR<br>M. Rahal, VED<br>S. Suck, OFF<br>S. Feuerstack, OFF<br>J. Pichen, ULM<br>A. Kassitzky, BroadBit |
| **Technical Approval:** | Fabio Tango |
| **Issue Authorisation:** | Andreas Lüdtke, OFF |

| 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 |
| 16.01.2017 | Preparation of initial document | A. Giralt (CAF) |
| 02.05.2017 | Freeze document structure | A Giralt (CAF) |
| 03.05.2017 to 22.06.217 | Contributions from partners | All |
| 23.06.2017 | Version to be reviewed | A.Giralt (CAF) |
| 30.06.2017 | Final version to be delivered | A.Giralt (CAF |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

# Table of Contents


**1 Introduction .......... 5**
**2 Automate Driver, vehicle and situation modelling concept .......... 5**
2.1 Enabler 1: Sensor and communication platform .......... 5
**2.1.1 Generic requirements of the Automate sensors** .......... 6
2.1.2 Global Positioning Sensing .......... 6
2.1.3 Digital map .......... 6
2.1.4 Environmental sensing .......... 7
2.1.5 Driver’s state sensor .......... 8
2.1.6 V2X (V2V and V2I) communication .......... 8
2.1.7 Vehicle data via in-vehicle buses .......... 9
2.1.8 Ego-Vehicle pose and motion .......... 9
2.1.9 Communication protocol .......... 9
2.2 Enabler 2: Driver Modelling and Learning .......... 10
2.2.1 DriveGOMS .......... 11
2.2.2 Driver Intention Recognition BadMob .......... 12
2.2.3 Driver state model .......... 18
2.2.4 Verification and Validation of Driver Models .......... 19
2.3 Enabler 3: Vehicle & situation models .......... 24
2.3.1 Joint Directors of Laboratories (JDL) fusion model .......... 24
2.3.2 Vehicle and Situation Models .......... 25
2.3.3 Semantic enrichment of the situation model .......... 28
2.3.4 Predicting the future evolution of the traffic scene .......... 31
2.3.5 V+V of Vehicle and Situation Models .......... 33
**3 Instantiation of the Automate platform .......... 36**
3.1 User scenario 1 (Peter) .......... 37
3.1.1 Sensors and communication platform of ULM demo car .......... 38
3.2 User Scenario 2 (Martha) .......... 39
3.2.1 Sensors and communication platform of VED demo car .......... 39
3.3 User Scenario 3 (Eva) .......... 40
3.3.1 Sensors and communication platform of the CRF test-car .......... 41
**4 Conclusion .......... 42**
**5 References .......... 44**

# 1 Introduction

The purpose of this document is to describe all the results of the sensor platform and models including the verification and validation results of task 2.2 to task 2.5 from the first cycle.
As described in the DOW the first cycle aims at developing a framework for later integration of all enablers by defining interfaces for data exchange and communication. This framework serves as an initial specification of the system architecture (Enabler 7). At the end of the initial requirement definition milestone M1 was reached. The technologies for Enabler 1–6 will be researched along the requirements and first versions of software components and will be implemented by taking into account the interface definitions. For the first cycle we perform exploration, validation and verification on a component level. For the current milestone M2 the current state of all exploration, validation and verification activities is documented with respect to the models in this document. This document will then be used as a starting point for the subsequent cycle. In parallel, we setup the baseline vehicles to be used for comparative evaluation in the upcoming cycles 2 and 3.

The document is divided in two main sections: In the following section "Automate driver, vehicle and situation modelling concept" we present the status of the WP2 development and tests performed during the first cycle as well as those planned for the two next cycles. The second main section "Instantiation of the Automate platform" is dedicated to the instantiation of the sensor and communication platform and the current state of development of the driver, situation and environment models that will then be later integrated into the different demonstration vehicles.

# 2 Automate Driver, vehicle and situation modelling concept

The Automate driver, vehicle and situation modelling concept is targeted to the implementation of the technical enablers 1, 2 and 3.

## 2.1 Enabler 1: Sensor and communication platform

As described in the DOW the objectives of enabler 1 are to use and advance existing sensor and V2X communication technology provided by the consortium partners as a technological basis to realize the objective 2 (Develop solutions to monitor, understand and anticipate the driver, the vehicle and the traffic situation).

### 2.1.1 Generic requirements of the Automate sensors

The three scenarios targeted within Automate require different sets of sensors based on the following ones:

- Global positioning Sensing (GPS)
- Digital map
- Environmental sensing
- Driver's state sensor
- Communication V2X, V2V
- Vehicle data
- Ego-vehicle pose and motion
- Communication protocol

All sensors must be calibrated to a global/local coordinate system and use the same clock. All provided sensor data must provide information about measurement uncertainties. Sensor data can be useful for visualization or evaluation of situation model. The situation model doesn't need a direct access to this layer since data needed for situation modeling are provided by the object layer. Therefore the object layer must be able to forward sensor data to the situation layer.

### 2.1.2 Global Positioning Sensing

This sensor must provide at least the ego-vehicle pose and motion as well as UTC time. The UTC time can be used for synchronization

### 2.1.3 Digital map

#### 2.1.3.1 Global Map

This map contains information about the road topology as well as the transient dynamic data. This information can be useful for situation prediction.

1. Global map topology: road-graph
2. Transient Dynamic Data (e.g. Traffic Jam, Construction, Blocking, average travel time)

#### 2.1.3.2 High accurate digital map

This map contains high-accurate information about road and infrastructure. This data can be recorded offline and/or detected during driving. A link to the global map must be available

1. Road/Lane (Marking, Curb, stop line, etc.)
2. Roadside infrastructure (Traffic sign, Traffic light, etc.)

### 2.1.4 Environmental sensing

The environmental data include the road, the objects (static and dynamics) and the driver state (internal scenario). These different classes are described in the following paragraphs.

#### 2.1.4.1 Road data

Road data is necessary to predict the future evolution of the traffic situation, as a necessary input for driver modelling (T2.3), vehicle and situation modelling (T2.4), and online risk assessment (T3.3). Road data can be detected or extracted from the high accurate digital map.

1. Road/Lane marking
2. Curb
3. Traffic light and signal

#### 2.1.4.2 Static Obstacles

All detected objects must be related to a unique/global coordinate system as well as have the same clock. Estimated data must provide uncertainties. This model contains static detected obstacles. Obstacles can be classified or not by three steps.

1. (Semantic) Occupancy Grid Maps
2. Stixels. A stixel is a vertical stick defined by its 3D position relative to the camera. Each stixel limits the free space and approximates the object boundaries.
3. Elevation/Drivable Maps

#### 2.1.4.3 Dynamic objects

Information about dynamic objects, assumed to represent other traffic participants, are necessary to predict the future evolution of the traffic situation, as a necessary input for driver modelling (T2.3), vehicle and situation modelling (T2.4), and online risk assessment (T3.3). This model contains a list of detected, tracked and fused dynamic objects with the attribute:

1. Position
2. Motion velocity and acceleration
3. Size
4. Semantic class

### 2.1.5 Driver's state sensor

The driver's state sensor is a vision based system which processes the video flow of the driver's face provided by one camera. From the image analysis the system detects and track facial features (eyelid, eye corners, mouth, etc.). The dynamics of these features are then analysed to determine the following driver's state models:

1. Drowsiness
2. Visual Inattention/Distraction
3. Cognitive distraction

In cycle 1 we have defined the sensor output according to the Automate requirements, ported and adapted the existing algorithms, a specific automate HW (camera, lights, processing unit) has been defined.

For cycle 2 the sensor HW will be finalized according to the requirements the different demonstration vehicle. The up-to-date models will then be integrated. More detailed sensor specifications are available in the document AutoMate_WP2_Driver_State_CAF_01.pdf

### 2.1.6 V2X (V2V and V2I) communication

In the AutoMate project, it is taken advantage of Vehicle-to-Vehicle and Vehicle-to-Infrastructure communication, which are called V2X together. In the first case, different V2V capable vehicles are communicating using wireless, RF connection. In the latter case, vehicles communicate with the infrastructure (e.g. traffic lights and signs, lamp- or utility poles, etc.). This communication between the entities is temporary, since the vehicles are in motion and often with high speeds. Therefore the connection between them is not sustainable. V2X is similar to a mobile ad hoc network; however, in this case the network elements are the vehicles and the road side elements (e.g. lamp post).

The benefit of V2X is to share and broadcast information between the vehicles. These information consist of frequently transmitted beacon messages (who I am, what is my current geo-position and speed, where I am heading etc.), warning messages (e.g. accident, oil spill on the road, traffic jam ahead etc.), environmental messages (e.g. heavy rain, frozen road, heavy cross-wind etc.).

In the EU the accepted standard for V2X is the GeoNetworking protocol (Ziya Cihan & Ali Gokhan Yavuz, 2013). It provides the above mentioned messages based different facilities and also it handles that these messages are broadcasted in the given geo-area, where they are relevant.

From V2X technology, the beacon messages will be mostly used in AutoMate project to allow the TeamMate car to sense, predict and react to other V2V capable cars in an extended area, and also to enhance the sensor fusion or decision making procedure. Further usage of V2X is required for the case that the infrastructure sends useful information about its current condition and events, e.g. a traffic sign that is able to transmit map information about the oncoming roundabout.

### 2.1.7 Vehicle data via in-vehicle buses

Vehicle data is required to compute the vehicle trajectory:

1. Speed
2. Yaw rate
3. Steering wheel Angle

### 2.1.8 Ego-Vehicle pose and motion

High-quality data of the ego-pose and –motion is necessary to derive the spatial relation between the ego-vehicle, other traffic participants, and the road data, as a necessary input for driver modelling (T2.3), vehicle and situation modelling (T2.4), and online risk assessment (T3.3). The ego-pose and motion consist of:

1. Position
2. Orientation
3. Motion (velocity and acceleration) according to a global coordinate system.

### 2.1.9 Communication protocol

The communication protocol between the AutoMate sensor systems is a set of libraries and tools for message passing and data marshalling, targeted at real-time systems where high-bandwidth and low latency are critical. It is specific to each demo car. Still in order to achieve a reliable and efficient exchange of data between the sensors, the vehicle and the models system the communication protocol should fulfil a minimal set of requirements:

- Real time
- Low-latency inter-process communication

- Efficient broadcast mechanism using UDP Multicast
- Type-safe message marshalling
- User-friendly logging and playback (lcm-logger and log-player)
- No centralized "database" or "hub" – peers communicate directly

## 2.2 Enabler 2: Driver Modelling and Learning

As described in the DOW the objective of enabler 2 is to build a probabilistic driver model. The model will describe the dynamic evolution and statistical relationships between the driver’s state, behaviour and environment and will enable to infer and predict the driver status, behaviour as well as intentions.

The general connection between situation-, vehicle-, and driver-models is depicted in Figure 1. The sensor and communication platform collects and enriches the available sensor information and passes this information to a component for situation understanding which maintains the situation-model as a current representation of the TeamMate vehicle’s belief about the world. The information is further enriched by the situation understanding to provide a semantic classification for the situation model. The resulting beliefs about the current state of the world are passed 1) to the situation prediction, which utilizes a set of vehicle models to estimate likely temporal and spatial evolution of the traffic scene, and 2) to the driver monitoring, which uses a set of driver models to estimate the current state of the driver.

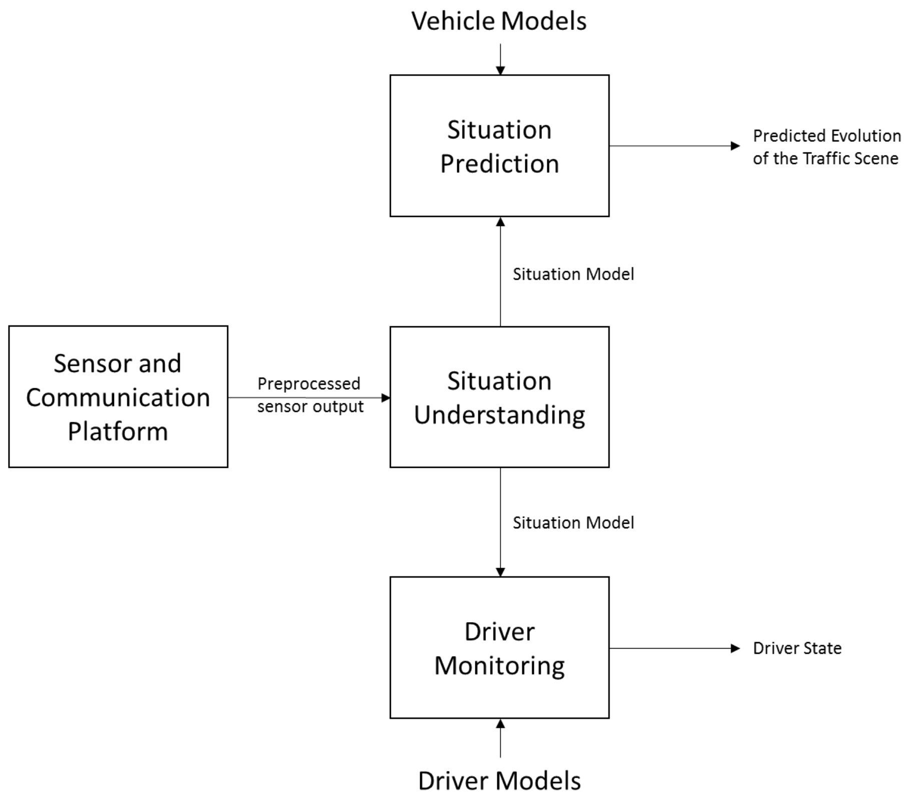



**Figure 1: Informal description of the information flow for situation-, vehicle- and driver-models.**

### 2.2.1 DriveGOMS

In the first cycle of AutoMate, we have continued the work on our driving task analysis framework, DriveGOMS. The original goal of the framework is make driver modelling for HMI design and evaluation easier. Within AutoMate, we plan to not only address these aspects by modelling the human-machine interaction, but also support the work on enabler 2 (Probabilistic Driver Modelling and Learning) by providing insight into the structure of human behaviour.

DriveGOMS applies the principles of the GOMS task analysis approach to the driving task (Card, Moran & Newell, 1983). This means a decomposition of the driving task into goals, operators, and methods. Goals are what structures the task, and can be derived e.g. from thinking aloud protocols. Operators can partially be measured, or derived from task necessities. Methods are a collection of operators and be viewed as driving manoeuvres. The resulting models can be used to predict execution times of driving activities, as a formal description of empirical driving data, or to define normative models of driving activities.

The purpose of the approach is to a) describe existing data, and to b) model unknown situations. The advantage of modelling existing data in this way is to have an un-ambiguous way to describe driving behaviour on all psychologically relevant levels. Since there are times attached to the operators, a model of a behavioural sequence (such as an interaction with the TeamMate car) predicts a time for this sequence.

In cycle 1, we specifically have worked to gain a better understanding of goals and operators based on empirical data from simulator driving studies. To this end, we used data acquired previously, and used this to hypothesize and validate goals and operators. This work included a lot of the necessary data pre-processing, such as data fusion (e.g. driving data and eye tracking data). We have produced lists of goals and operators that can now be used to model driver-vehicle-interactions.

In cycle 2, we will support the empirical analysis of the studies conducted within WP2 with our framework. That means that we will model the driver behaviour and driver interaction with an automation based on the empirical data, and use this knowledge to make suggestions regarding the design of the TeamMate car.

### 2.2.2 Driver Intention Recognition BadMob

In Deliverable D2.1 "Metrics and Experiments for V&V of the driver, vehicle, and situation models in the 1st cycle", we introduced Bayesian Autonomous Driver Mixture-of-Behaviors (BAD MoB) models and Driver Intention Recognition (DIR) models as a starting point for a coherent probabilistic architecture for intention recognition and behavior prediction.

In the following, random variables will be denoted by capital letters, such as$X$, $Y$, $Z$, and we will use corresponding lower-case letters $x$, $y$, $z$ to denote specific values taken by such variables. The set of values, a random variable $X$ may take, will be denoted by $\mathrm{Val}(X)$. Sets of variables will be denoted by bold capital letters, e.g., $\boldsymbol{X} = \{X_1, \ldots, X_n\}$, and we will use lower-case bold letters $\boldsymbol{x} = \{x_1, \ldots, x_n\}$ to denote specific values taken by such sets. Dealing with temporal models, the time line is assumed to be discretized into time slices with a constant time granularity of $\Delta t$. Time slices are indexed by non-negative integers, and we use $X^t$ to represent the instantiation of a variable $X$ at a time $t$. A sequence of variables $X$, resp. sets of variables $\boldsymbol{X}$, from time

$i$ to $j$ will be denoted by $X^{i:j}$, resp. $\boldsymbol{X}^{i:j}$. Lastly, probability density functions (PDFs) and probability distributions (CPDs) will uniformly be denoted by $p(.)$.

Let $\boldsymbol{A} = \{A_1, \ldots, A_{n_A}\}$ denote a set of continuous and/or discrete random variables that represent the control actions of the human driver; $\boldsymbol{B} = \{B_1, \ldots, B_{n_B}\}$ denotes a set of discrete variables that represent different behaviors and intentions on the maneuvering layer, $\boldsymbol{P} = \{P_1, \ldots, P_{n_P}\}$ denotes a set of continuous and/or discrete random variables that represent the hypothetical perceptual input of the human driver. Additionally, let $\Delta$ denote a delay between perception and action due to perception and reaction times. A BAD MoB model defines a (conditional) Dynamic Bayesian Network that specifies the (conditional) probability density function $p(\boldsymbol{A}^{1:T}, \boldsymbol{B}^{1:T} | \boldsymbol{p}^{1-\Delta:T-\Delta})$ for any number of $T$ time slices as

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

For the creation of the DIR model, the dependence of behaviors and action variables on the past perception was replaced by a more traditional sensor model. Let $\boldsymbol{O} = \{O_1, \ldots, O_{n_O}\}$ denote a set of continuous and/or discrete random variables that represent the observations of the current traffic situations, a DIR model would therefore model the joint density distributions over actions, behaviours and observations over an arbitrary length $T \geq 1$ as:

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

Following this short reminder, this deliverable will introduce a template for probabilistic driver models developed in AutoMate. This template is assumed to be adapted to the different scenarios resp. demonstrators based on dataset obtained in these scenarios.

For the probabilistic driver models in AutoMate, we start with the idea of BAD MoB models, where we replace the perception variables $\boldsymbol{P}$, by scenario-dependent subsets of variables $\boldsymbol{S}$ provided by the situation-model (c.f., Section 2.3.2), representing necessary information about the current situation, including e.g., the state of the ego-vehicle, surrounding vehicles, and the future course of the road. We furthermore assume knowledge about the mode, either manual or autonomous driving, of the TeamMate car, represented by a binary variable $M$, with $\mathrm{Val}(M) = \{m_M, m_A\}$. The resulting probabilistic driver models define a CPD $p(\boldsymbol{A}^{1:T}, \boldsymbol{B}^{1:T}, \boldsymbol{S}^{1:T} | M^{1:T})$ for any number of $T$ time slices as

$$p(\boldsymbol{A}^{1:T}, \boldsymbol{B}^{1:T}, \boldsymbol{S}^{1:T} | M^{1:T}) = p(\boldsymbol{A}^{1}, \boldsymbol{B}^{1}, \boldsymbol{S}^{1} | M^{1}) \prod_{t=2}^{T} p(\boldsymbol{A}^{t}, \boldsymbol{B}^{t}, \boldsymbol{S}^{t} | \boldsymbol{A}^{t-1}, \boldsymbol{B}^{t-1}, \boldsymbol{S}^{t-1}, M^{t}),$$

with

$$p(\boldsymbol{A}^{t}, \boldsymbol{B}^{t}, \boldsymbol{S}^{t} | \boldsymbol{A}^{t-1}, \boldsymbol{B}^{t-1}, \boldsymbol{S}^{t-1}, M^{t}) = p(\boldsymbol{B}^{t} | \boldsymbol{B}^{t-1}, \boldsymbol{S}^{t-1}) p(\boldsymbol{A}^{t} | \boldsymbol{A}^{t-1}, \boldsymbol{S}^{t-1}, \boldsymbol{B}^{t}) p(\boldsymbol{S}^{t} | \boldsymbol{S}^{t-1}, \boldsymbol{A}^{t}, M^{t}).$$

Here, the CPD $p(\boldsymbol{B}^{t} | \boldsymbol{B}^{t-1}, \boldsymbol{S}^{t-1})$ models the formation of intentions and driving maneuvers/behaviors based on the current state of the traffic situation, $p(\boldsymbol{A}^{t} | \boldsymbol{A}^{t-1}, \boldsymbol{S}^{t-1}, \boldsymbol{B}^{t})$ models the selection of control actions for different intentions and behaviors based on the current state of the traffic situation, and $p(\boldsymbol{S}^{t} | \boldsymbol{S}^{t-1}, \boldsymbol{A}^{t}, M^{t})$ models the evolution of the traffic situation dependent on the control inputs of the human driver if in a manual mode:

$$p(\boldsymbol{S}^{t} | \boldsymbol{S}^{t-1}, \boldsymbol{A}^{t}, M^{t}) = \begin{cases} p(\boldsymbol{S}^{t} | \boldsymbol{S}^{t-1}, \boldsymbol{A}^{t}, M^{t}), & M^{t} = m_{M}^{t} \\ p(\boldsymbol{S}^{t} | \boldsymbol{S}^{t-1}, M^{t}), & M^{t} = m_{A}^{t} \end{cases}.$$

This template is motivated by reasonable assumptions on the causal mechanisms. The formation of intentions and resulting selection of behaviors is governed on the traffic situation, as perceived by the human driver. Resp., the selection of control actions will be guided by the underlying intentions and selected maneuvers/behaviors guided by the perception of the traffic situation. Lastly, the evolution of the traffic scene itself will directly be influenced by the selected control actions, but not by the driver's intentions or behaviors, outside of the effects of control actions.

The CPDs $p(\boldsymbol{B}^{t} | \boldsymbol{B}^{t-1}, \boldsymbol{S}^{t-1})$ and $p(\boldsymbol{A}^{t} | \boldsymbol{A}^{t-1}, \boldsymbol{S}^{t-1}, \boldsymbol{B}^{t})$ will be estimated from multivariate time-series of human behavior traces obtained in the experiments planned for the first cycle (Section 2.2.4.1). The CPD $p(\boldsymbol{S}^{t} | \boldsymbol{S}^{t-1}, \boldsymbol{A}^{t}, M^{t})$ will reuse the algorithms developed for the prediction of the spatial and temporal evolution of the traffic scene.

As an alternative, we will consider replacing $p(\boldsymbol{B}^{t} | \boldsymbol{B}^{t-1}, \boldsymbol{S}^{t-1})$ by $p(\boldsymbol{B}^{t} | \boldsymbol{B}^{t-1})$ and $p(\boldsymbol{S}^{t} | \boldsymbol{S}^{t-1}, \boldsymbol{A}^{t}, M^{t})$ by $p(\boldsymbol{S}^{t} | \boldsymbol{S}^{t-1}, \boldsymbol{B}^{t}, \boldsymbol{A}^{t}, M^{t})$ akin to DIR models.

The goal of the probabilistic driver model in AutoMate is to maintain a belief state $p(\boldsymbol{B}^{t}, \boldsymbol{A}^{t}, \boldsymbol{S}^{t} | \boldsymbol{e}^{1:t}, M^{t} = m_{M}^{t})$ at each time step $t$ over intentions resp. behaviors, control actions and the world state, given all "evidence" $\boldsymbol{e}^{1:t}$ (the exact nature of evidence will be explained below) obtained by the TeamMate vehicle up to the current point in time $t$ and under the assumption that the TeamMate vehicle is controlled by the human driver. The belief state can then be further processed to derive information required for the different demonstrators. For the ULM demonstrator, the probabilistic driver model will provide a belief state over the current intentions and behaviors of the human driver as a means for selecting appropriate maneuvers to be performed autonomously by the TeamMate vehicle, e.g.:

$$p(\boldsymbol{B}^t|\boldsymbol{e}^{1:t}, M^t = m_M^t) = \int_{-\infty}^{\infty} p(\boldsymbol{B}^t, \boldsymbol{a}^t, \boldsymbol{s}^t|\boldsymbol{e}^{1:t}, M^t = m_M^t)\, d\boldsymbol{a}^t d\boldsymbol{s}^t.$$

For the VED and CRF demonstrator, under the assumption that the model captures the normative driving behavior of the human driver, the probabilistic driver model will provide an assessment of the current driving parameters and control actions of the human driver, e.g.:

$$p(\boldsymbol{a}^t, \boldsymbol{s}^t|\boldsymbol{e}^{1:t}, M^t = m_M^t) = \sum_{\boldsymbol{b}\in \mathrm{Val}(\boldsymbol{B})} p(\boldsymbol{b}^t, \boldsymbol{a}^t, \boldsymbol{s}^t|\boldsymbol{e}^{1:t}, M^t = m_M^t).$$

It remains to specify, what evidence is available at each time step to estimate the current belief state. We assume that the driver model has access to the current mode of the TeamMate vehicle $m^t$, the belief state over the state of the TeamMate vehicle $p(\boldsymbol{X}_{TM}^t|\boldsymbol{o}^{1:t})$, the belief states over other traffic participants $p\left(\boldsymbol{X}_{O_i}^t|\boldsymbol{o}^{1:t}\right), i = 1, \dots, m$ and the description of the environment around the TeamMate car provided e.g., by a map $M$, provided by the situation model.

An important challenge is the incorporation of uncertain information provided by the situation model in the driver model. A first possibility to incorporate the information provided by the situation model is to discard the knowledge about uncertainty and just use the modes of each belief state as evidence. As such, at each time step $t$, we'd have evidence about the current control actions $\boldsymbol{A}^t = \boldsymbol{a}^t$, the current world state $\boldsymbol{S}^t = \boldsymbol{s}^t$, and the current mode of operation $m^t$. We can use this information to recursively obtain a belief state $p(\boldsymbol{B}^t|\boldsymbol{s}^{1:t}, \boldsymbol{a}^{1:t}, m^{1:t})$ from a previously inferred belief state $p(\boldsymbol{B}^{t-1}|\boldsymbol{s}^{1:t-1}, \boldsymbol{a}^{1:t-1}, m^{1:t-1})$ in the following way:

$$
\begin{aligned}
&p(\boldsymbol{B}^t|\boldsymbol{s}^{1:t}, \boldsymbol{a}^{1:t}, m^{1:t}) = \frac{1}{Z} p(\boldsymbol{B}^t, \boldsymbol{a}^t, \boldsymbol{s}^t|\boldsymbol{s}^{1:t-1}, \boldsymbol{a}^{1:t-1}, m^{1:t}) \\
&\propto \sum_{\boldsymbol{b} \in \mathrm{Val}(\boldsymbol{B})} p(\boldsymbol{B}^t, \boldsymbol{a}^t, \boldsymbol{s}^t|\boldsymbol{b}^{t-1}, \boldsymbol{a}^{t-1}, \boldsymbol{s}^{t-1}, m^t) p(\boldsymbol{b}^{t-1}|\boldsymbol{s}^{1:t-1}, \boldsymbol{a}^{1:t-1}, m^{1:t-1}) \\
&= \sum_{\boldsymbol{b} \in \mathrm{Val}(\boldsymbol{B})} p(\boldsymbol{B}^t|\boldsymbol{b}^{t-1}, \boldsymbol{s}^{t-1}) p(\boldsymbol{a}^t|\boldsymbol{a}^{t-1}, \boldsymbol{s}^{t-1}, \boldsymbol{b}^t) p(\boldsymbol{s}^t|\boldsymbol{s}^{t-1}, \boldsymbol{a}^t, m^t) p(\boldsymbol{b}^{t-1}|\boldsymbol{s}^{1:t-1}, \boldsymbol{a}^{1:t-1}, m^{1:t-1}),
\end{aligned}
$$

with $Z$ being a normalization constant:

$$
Z = p(\boldsymbol{a}^t, \boldsymbol{s}^t|\boldsymbol{s}^{1:t-1}, \boldsymbol{a}^{1:t-1}, m^{1:t}) = \sum_{\boldsymbol{b} \in \mathrm{Val}(\boldsymbol{B})} p(\boldsymbol{b}^t, \boldsymbol{a}^t, \boldsymbol{s}^t|\boldsymbol{s}^{1:t-1}, \boldsymbol{a}^{1:t-1}, m^{1:t}).
$$

This method is very efficient, but may be unsatisfactory, as we're discarding any uncertainties about the information provided by the situation model.

A potential method to incorporate uncertainties in the evidence is uncertain or *soft evidence*. The basic idea is also known as Jeffrey's rule and follows the idea to first define a model conditioned on the evidence and then average over the distribution of the evidence (Barber, 2012). Let $\tilde{\boldsymbol{a}}$ and $\tilde{\boldsymbol{s}}$ denote the uncertain evidence for both the actions and states provided by the situation model. The inference scheme for the probabilistic driver model using soft evidence is given by:

$$
\begin{aligned}
p(\boldsymbol{B}^t|\tilde{\boldsymbol{a}}^{1:t}, \tilde{\boldsymbol{s}}^{1:t}, m^{1:t}) &\propto \int_{-\infty}^{\infty} p(\boldsymbol{B}^t, \boldsymbol{a}^t, \boldsymbol{s}^t|\tilde{\boldsymbol{a}}^{1:t}, \tilde{\boldsymbol{s}}^{1:t}, m^{1:t}) \; d\boldsymbol{a}^t d\boldsymbol{s}^t \\
&= \int_{-\infty}^{\infty} p(\boldsymbol{B}^t|\boldsymbol{a}^t, \boldsymbol{s}^t, \tilde{\boldsymbol{a}}^{1:t}, \tilde{\boldsymbol{s}}^{1:t}, m^{1:t}) \, p(\boldsymbol{a}^t, \boldsymbol{s}^t|\tilde{\boldsymbol{a}}^{1:t}, \tilde{\boldsymbol{s}}^{1:t}, m^{1:t}) \, d\boldsymbol{a}^t d\boldsymbol{s}^t,
\end{aligned}
$$

where we assume that

$$
\begin{aligned}
&p(\boldsymbol{B}^t|\boldsymbol{a}^t, \boldsymbol{s}^t, \tilde{\boldsymbol{a}}^{1:t}, \tilde{\boldsymbol{s}}^{1:t}, m^{1:t}) \\
&\propto \sum_{\boldsymbol{b} \in \mathrm{Val}(\boldsymbol{B})} \int_{-\infty}^{\infty} p(\boldsymbol{B}^t, \boldsymbol{a}^t, \boldsymbol{s}^t|\boldsymbol{a}^{t-1}, \boldsymbol{s}^{t-t}, m^t) p(\boldsymbol{b}^{t-1}, \boldsymbol{a}^{t-1}, \boldsymbol{s}^{t-1}|\tilde{\boldsymbol{a}}^{1:t-1}, \tilde{\boldsymbol{s}}^{1:t-1}, m^{1:t-1}) \; d\boldsymbol{a}^{t-1} d\boldsymbol{s}^{t-1}
\end{aligned}
$$

and

$$
p(\boldsymbol{A}^t, \boldsymbol{S}^t|\tilde{\boldsymbol{a}}^{1:t}, \tilde{\boldsymbol{s}}^{1:t}, m^{1:t}) = \prod_{A \in \boldsymbol{A}} p(A^t|\tilde{a}^t) \prod_{S \in \boldsymbol{S}} p(S^t|\tilde{s}^t),
$$

with $p(X^t|\tilde{x}^t)$ being given by the current belief in the situation model. Unfortunately, the resulting inference scheme requires the joint integration resp. summation over action variables $\boldsymbol{A}^t$ and state variables $\boldsymbol{S}^t$ that cannot be simplified due to the model structure, which contrasts with potential simplification for the integration and summation over past action and state variables. As such, inference using uncertain evidence may turn out to be too costly for the context of AutoMate, which must be tested during the project.

Another possibility, and the one we'll focus on first, is the use of unreliable or *likelihood evidence* (Barber, 2012). The basic idea is as follows. For each variable $X \in \{\boldsymbol{A}, \boldsymbol{S}\}$ we extend the model by a corresponding variable $O_X$ and CPD $p(O_X^t|X^t)$. During runtime, we assume $O_X^t$ to be observed and define the likelihood $p(o_X^t|X^t)$ to equal the corresponding belief in the situation model. Let $\boldsymbol{O}$ denote the set of all variables $O_X$, we can now recursively obtain a belief state $p(\boldsymbol{B}^t, \boldsymbol{A}^t, \boldsymbol{S}^t|\boldsymbol{o}^{1:t}, m^{1:t})$ from a previously inferred belief state $p(\boldsymbol{B}^{t-1}, \boldsymbol{A}^{t-1}, \boldsymbol{S}^{t-1}|\boldsymbol{o}^{1:t-1}, m^{1:t-1})$ in the following way:

$$p(\boldsymbol{B}^t, \boldsymbol{A}^t, \boldsymbol{S}^t|\boldsymbol{o}^{1:t}, m^{1:t})$$
$$\propto \sum_{\boldsymbol{b}\in\mathrm{Val}(\boldsymbol{B})} \int_{-\infty}^{\infty} p(\boldsymbol{B}^t, \boldsymbol{A}^t, \boldsymbol{S}^t, \boldsymbol{o}^t|\boldsymbol{b}^{t-1}, \boldsymbol{a}^{t-1}, \boldsymbol{s}^{t-1}, m^t) p(\boldsymbol{b}^{t-1}, \boldsymbol{a}^{t-1}, \boldsymbol{s}^{t-1}|\boldsymbol{o}^{1:t-1}, m^{1:t-1}) d\boldsymbol{a}^{\boldsymbol{t-1}} d\boldsymbol{s}^{\boldsymbol{t-1}}$$
$$= p(\boldsymbol{o}^t|\boldsymbol{A}^t, \boldsymbol{S}^t) \sum_{\boldsymbol{b}\in\mathrm{Val}(\boldsymbol{B})} \int_{-\infty}^{\infty} p(\boldsymbol{B}^t, \boldsymbol{A}^t, \boldsymbol{S}^t|\boldsymbol{b}^{t-1}, \boldsymbol{a}^{t-1}, \boldsymbol{s}^{t-1}, m^t) p(\boldsymbol{b}^{t-1}, \boldsymbol{a}^{t-1}, \boldsymbol{s}^{t-1}|\boldsymbol{o}^{1:t-1}, m^{1:t-1}) d\boldsymbol{a}^{\boldsymbol{t-1}} d\boldsymbol{s}^{\boldsymbol{t-1}},$$

where

$$p(\boldsymbol{o}^t|\boldsymbol{A}^t, \boldsymbol{S}^t) = \prod_{A\in\boldsymbol{A}} p(o_A^t|A^t) \prod_{S\in\boldsymbol{S}} p(o_S^t|S^t).$$

Compared to the notion of soft evidence, likelihood evidence has the advantage that $p(\boldsymbol{o}^t|\boldsymbol{A}^t, \boldsymbol{S}^t)$ can be evaluated without the need to jointly integrate resp. sum over the current state and actions, making it much more efficient for performing inferences in real-time scenarios. It is to note however, that likelihood evidence works on a fundamentally different principle than soft evidence. Using soft evidence, we'd assume that the beliefs provided by the situation model are *correct,* but uncertain. Likelihood evidence on the other hand, will be fused into the prior beliefs of the model itself, only shifting the driver model's a-priori beliefs towards the beliefs provided by the information model. For now, the effects of using likelihood evidence are not tested, but will be analyzed for the next cycle.

### 2.2.3 Driver state model

The driver state model aims to provide an indication about the physiological, behavioural and psychological state of the driver. The driver's state model is a SW module, which provides the following the models: Drowsiness, Visual attention and visual distraction, and Cognitive distraction (possibly, under investigation).

#### 2.2.3.1 Drowsiness

Drowsiness is a state of reduced consciousness (or, near-sleep) due to sleep pressure. The drive to sleep is primarily caused by increased activity of the sleep system, in combination with decreased activity of the arousal system. Somnolence or sleepiness can be caused by prior lack of sleep and/or circadian disturbance, and might be exacerbated by long periods of inactivity/boredom.

Note that for definition (and measurement) sake, there is a difference between drowsiness/sleepiness (due to sleep need) and fatigue (due to excessive exertion of mental effort).

There are many problems related with drowsiness: lowered acuity in perception of driving events, reduced tasks performance, impaired judgement abilities, lower reaction, delay, etc. Moreover the risk of error is increased by the fact that drivers are mostly not able to make a reliable evaluation or acknowledgement of their sleepiness level.

Drowsiness is characterized by many physiological symptoms. The most mentioned in the literature are: an increase of the blink duration, yawning, head leaning forward, reduced eyelid opening, and eye gaze staring. The driving behaviour is also affected. The driver shows difficulties to maintain an accurate trajectory. Vehicle drifting and swaying in the lane are symptoms of a significantly degraded drowsiness.

The developed drowsiness model is mainly based on the increase of the blink duration. The model output 4 drowsiness levels correlated with the Karolinska Sleepiness Scale (KSS) ranging from alert to falling asleep (Boverie & Giralt, 2008). Within Automate we will reinforce this diagnostic using facial or head behaviour. In cycle 1 the worked focused on head specific movement like leaning forward or backward to the head rest in a specific way. The achieved performance will guideline future works toward this head movement approach or toward specific facial behaviour like talking or yawning.

These developments are supported by a drowsiness labelled data base of 15 subjects driving in simulator conditions.

#### 2.2.3.2 Visual attention/distraction

A driver is visually distracted when s/he is not looking ahead at the road; his eye gaze is off the road. The underlying idea is that a driver who is visually distracted cannot be fully aware of the situation. But the contrary case is not true: a driver who is looking ahead is not necessarily aware of the situation; he could be drowsy or cognitively distracted.

A visual attention level is computed from the visual attention distribution (Visual Time Sharing, VTS) of the different areas of interest located inside and outside of the vehicle according to the driving situation (Boverie & Cour, 2011).

In cycle 1 basic visual distraction models based on the proportion of time the driver spends looking at the road have been integrated in the Automate algorithmic frame work. In cycle 2 they will be further improved by tuning the timings, and weights of the different areas observed by the driver. Still the main line of improvement will be to improve the accuracy and the robustness of the eye/head gaze provided by the face tracker. This work is carried on within the development of the driver's state sensor.

#### 2.2.3.3 Cognitive distraction

Cognitive distraction can occur when attention is withdrawn from the processing of information necessary for the safe operation of a motor vehicle, when an individual's focus is not directly on the act of driving and his/her mind "wanders". Many non-related driving tasks like speech to text system, talking on a cell phone or talking to passenger may generate a certain level of cognitive distraction.

Estimate the driver's cognitive distraction is a very challenging tasks. The developed model will be based on indirect observations of non-related driving activities combined with a decreased visual scanning of the driving environment. The investigations will start in cycle 2.

### 2.2.4 Verification and Validation of Driver Models

#### 2.2.4.1 Driver Intention Recognition

The first cycle for verification and validation on a component level is currently being performed by a joined study with the partners from ULM and

OFF. The study is being performed with 48 subjects driving the Peter scenario on two driving simulators: one in ULM and the other one located at OFF. In the Peter scenario, the TeamMate car is driving on a rural road. A big vehicle, which in the validation scenario is a truck, reduces the capturing capabilities of different sensors of the car and therefor it is up to the driver to support the TeamMate car to decide about the correct point in time for that it is safe to initiate the overtaking manoeuvre. This interaction should be designed, following the TeamMate approach. The driver and the vehicle work together as two teammates, where the vehicle still supports the driver as much as possible so the driver only needs to initiate the overtaking manoeuvre.

For this first study the driver can initiate the overtaking based on taping the indicator or by pressing a button located on the touch screen of a centre touch panel. One of the goals is to figure out if there is a difference between these two interaction designs.

Figure 2 and figure 3 show the experiment setups for both simulators. For the studies both simulators shared the same software setup and also the same road and traffic structure. During the slot the road (which is a secondary road with only one lane in each direction) changes between sections with flat parts and good sight to more curvy ones with forest sections that limit the frontal view of the driver to identify potential obstacles early. Finally, basic inner city sections are also part of the road track, but for this first study that focuses on the Peter scenario intersections are not relevant.

**Figure 2: OFFIS driving simulator**

**Figure 3: ULM driving simulator**

Each subject had to drive three slots, taking around 30 minutes to drive. For two of the three slots the AutoMate car is driving in automated driving mode and therefore the driver is mainly concerned with identifying the correct time to initiate an overtaking manoeuvre for those situations in that the AutoMate car cannot automatically overtake.

In one slot each subject is driving completely without any automation involved. The data collected in this slot is on the one hand, the baseline for the AutoMate car and on the other hand, the training and validation data for the first version of the driver intention recognition component. Several variables will be recorded from the simulator. Besides all driving variables from the ego vehicle (for example lateral and longitudinal acceleration) of all vehicles are recorded. All these data are needed to validate the driver intention model.

#### 2.2.4.2 Driver Situation Awareness Assessment

Also the situation awareness of the driver is explored as part of the same study. The subject’s eyes are tracked based on remote eye-tracking cameras in the UML driving simulator and a head-mounted eye tracker that is used in the OFF driving simulator. While the subjects are driving without interruptions for each slot, they pass certain identification points (i.e. flow points) that trigger specific situations (e.g. the slow vehicle to appear) or identify phases of an overtaking manoeuvre (e.g. left lane merge, overtake, right lane merge). The eye tracking data is recorded for all subjects and all three slots and the gained data from the study is being explored on changes of subjects' visual situation awareness construction procedures and hedging behaviour between manual driving and automated driving situations in that the AutoMate car requests driver support to initiate the automated overtaking manoeuvre.

#### 2.2.4.3 Driver state estimation

In the current state of the project, a final decision about how to test the driver monitoring system has not been taken yet and it depends on where this system will be implemented (driving simulator, real car or both) and what it is possible to evaluate (distraction or drowsiness).

**Distraction.**

In this section, we sketch some ideas for distraction[2]. All in all, driver's distraction – and inattention – is an important safety concern and not a new problem in road safety: we may say that it has been around for as long as people have been driving cars. It is moreover likely that the problem will increase as more wireless or mobile technologies find their way into vehicles. Being distracted can make drivers less aware of other road users such as pedestrians, cyclists and road workers and less observant of road rules such as speed limits and junction controls.

Drivers do much more than control the vehicle when driving (such as: adjusting an entertainment system or climate control, consulting maps, eating / drinking / smoking, interacting with passengers, and so on). Driver distraction occurs when a driver diverts their attention away from the activities needed for safe driving. Distracted driving is the state that occurs when attention is given to a non-driving related activity, typically to the detriment of driving performance.

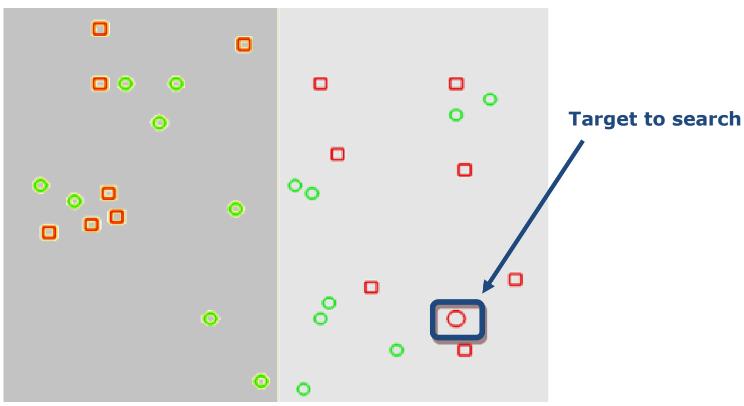
Here, we focus on a particular type of distraction the *visual* one, which occurs when a driver takes their eyes off the road. Typically this is cause when the driver looks away from the road to engage in a secondary activity either inside (e.g. radio, telephone) or outside (e.g. signs, advertisements) of the vehicle.

As aforementioned, driving is a complex task: a person must engage almost all of their mental faculties (in other words, it is not simply about physically

[2] Drowsiness is not easy to test, because of safety constraints and complexity of the tests. A possibility is to have a complete evaluation for distraction and to focus only on detection of false positives for drowsiness.

controlling the car) so it is not surprising that attention-grabbing distractions can interfere with successful and safe completion of the driving task. The brain never actually focuses on two tasks at the same time, it switches back and forth between them – true 'multi-tasking' is a myth. Your performance suffers as you struggle to divide your attention (detrimental in accuracy).

In order to evaluate driver's distraction, dedicated tests have to be carried out. For example, a certain number of participants can be asked to drive on the dedicated test-site in real-traffic situations, while completing a secondary task session. Distraction (visual and manual) can be induced by means of a secondary visual research task, called SuRT, reproduced on an in-vehicle touch screen (7'' TFT touch screen installed on the right-hand side of the car cabin). SURT was chosen to simulate an IVIS (In Vehicle Information System). It requires visual perception and manual response, possibly causing a degradation of driving task performances. The situation is depicted in the following figures:



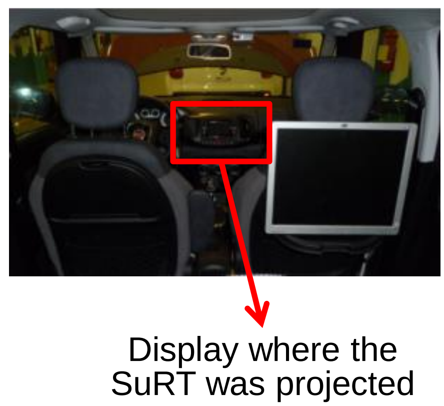




**Figure 4: sketch of how the SuRT works and possible location inside the vehicle cockpit.**

Participants are presented with a set of stimuli on a touch screen (e.g. a tablet or a smart phone) which can be mounted on the right side of the steering wheel in reach of the driver's right arm. The time interval between two consecutive screens was pseudo-randomized between 3 and 9 seconds. The output data are the reaction times and the error rates.

At the moment the use of this methodology (represented by SuRT) is still under discussion. Alternatively, it can be used a secondary task based on reading aloud a sequence of random letters, with a predefined duration.

## 2.3 Enabler 3: Vehicle & situation models

As described in the DOW the objectives of enabler 3 are to infer an integrated probabilistic vehicle and situation model from the data provided by Enabler 1 (incl. information from other TeamMate Cars). The model will integrate and represent all traffic participants in the surroundings of the TeamMate Car as well as the dynamic characteristics of the own vehicle. This will be done in a way, which is consistent to human situation understanding e.g. by applying scene understanding/classification techniques to put the recognized objects in relation with each other.

### 2.3.1 Joint Directors of Laboratories (JDL) fusion model

To realize the vehicle and situation modelling, we propose using the *Joint Directors of Laboratories* (JDL) fusion model as it provides an established and time proven approach to handling complex environments. It was initially developed for military applications and later adapted to the use in an automotive context (Polychronopoulos & al., 2006). For the purpose of AutoMate, we will employ tailored version of the JDL model (see Figure 5). The perception and the decision/situation layer of the proposed model are explained in the next subsections.

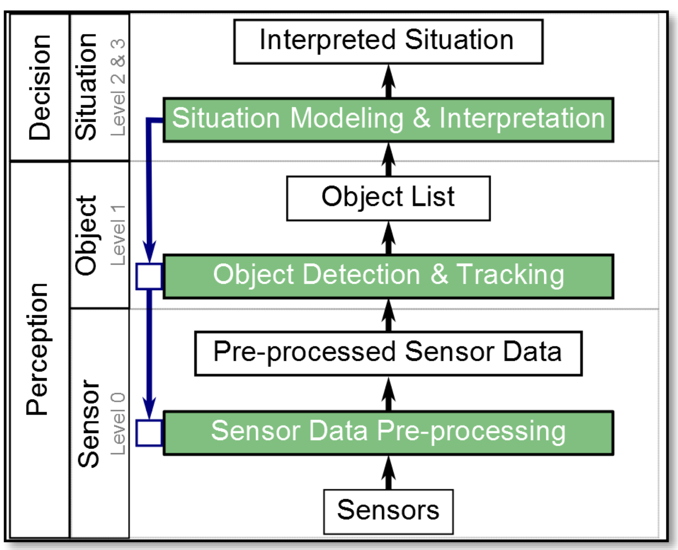



**Figure 5: simplified version of the JDL model for sensor data fusion**

The *perception layer* consists of the sensor and object level. In the first step, sensor data are pre-processed on the sensor level. GPS, (stereo) camera, RADAR and laser scanner are widely used sensors. A digital map server is also a potential sensor. All sensors must be calibrated with respect to a common coordinate system and synchronized. The situation layer doesn't need a direct access to the sensor level since data needed for the situation modelling und interpretation are provided by the object level. The object level must be able to forward sensor data to the situation layer. The sensor data (e.g. camera images) can be used on the situation level for visualization as well as verification and validation.

In the second step, pre-processed sensor data from the sensor level are used for object detection. Detected objects are fused and tracked over the time. Moreover, highly-accurate information about the road (road markings, curb, etc.), traffic infrastructure (traffic light, traffic signal, etc.) and free space is extracted. These data can be recorded offline as a digital map and/or detected during driving. The ego-pose and –motion are also estimated on this layer. The data generated on the perception layer are inputs for the situation modelling and understanding. More details on the perception layer can be found in the subsection 2.1.
On the *decision layer*, inputs from the perception layer are integrated into a situation model. The situation model is enriched with semantic information and used to predict the evolution on the situation. Another part on the Decision layer is the threat assessment where the situation criticality is estimated. Such issues are addressed in WP3.

### 2.3.2 Vehicle and Situation Models

For the driver models developed for intention recognition and online risk assessment, the situation model is intended as an intermediate layer between the sensor and communication platform and the subsequent driver-, and vehicle-models and online risk assessment. More specifically, the situation model represents a subset of the TeamMate vehicle's current belief about the world based on sensor observations and it is assumed that the information of the situation model is updated via the sensor and communication platform in constant tie intervals$\Delta t$.

In general, the situation model is assumed to maintain information about the current state of the TeamMate vehicle, the current states of a number of objects recognized in the vicinity of the TeamMate vehicle and a description of the environment. More specifically, we assume the existence of a map $M$ centered at the current position of the TeamMate vehicle that allows reasonable reconstruct the course of the road in the vicinity of the TeamMate vehicle. For now, we don't specify the exact format.

Concerning the TeamMate vehicle, the situation model is assumed to maintain information about the current state of the TeamMate vehicle, represented by a set of variables as described in Table1:

$\boldsymbol{X}_{TM}^{t} = \{X_{TM}^{t}, Y_{TM}^{t}, \Theta_{R_{TM}}^{t}, \Theta_{A_{TM}}^{t}, D_{TM}^{t}, L_{TM}^{t}, V_{TM}^{t}, A_{TM}^{t}, W_{TM}^{t}, S_{L_{TM}}^{t}, S_{W_{TM}}^{t}, \mathrm{A}_{A_{TM}}^{t}, \mathrm{A}_{B_{TM}}^{t}, \mathrm{A}_{S_{TM}}^{t}, G_{TM}^{t}\}.$

**Table 1: Description of variables for the representation of the TeamMate vehicle considered for the first cycle.**

| Variable | Type | Unit | Description |
|---|---|---|---|
| $X_{TM}$ | Continuous | [m] | X-coordinate of the center 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 center of the TeamMate vehicle in a two-dimensional spatial coordinate system relative to an origin synchronized with the map $M$ |
| $\Theta_{\mathrm{R}_{TM}}$ | Continuous | [rad] | Yaw-angle relative to a global x-axis synchronized with the map $M$ |
| $\Theta_{\mathrm{A}_{TM}}$ | 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 centerline on a two-lane road |
| $L_{TM}$ | Discrete | $\{0, \dots, \lfloor L_{TM} \rfloor\}$ | 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/s²] | Longitudinal acceleration |
| $W_{TM}$ | Continuous | [rad/s] | Yaw-rate |
| $S_{L_{TM}}$ | Continuous | [m] | Length (along the x-axis) |
| $S_{W_{TM}}$ | Continuous | [m] | Width (along the y-axis) |
| $A_{A_{TM}}$ | Continuous | [%] | Activation of the acceleration pedal |
| $A_{B_{TM}}$ | Continuous | [%] | Activation of the braking pedal |
| $A_{S_{TM}}$ | Continuous | [rad] | Steering wheel angle |
| $G_{TM}$ | Discrete | $\{0, \dots, \lfloor G_{TM} \rfloor\}$ | Selected gear |

Within the situation model, the TeamMate state is expected to be provided as a probability density function (pdf) $p(\boldsymbol{X}_{TM}^{t} | \boldsymbol{o}^{1:t})$, the belief state about the state of the TeamMate vehicle given all sensor information up to the current point in time $t$. For the first cycle, $p(\boldsymbol{X}_{TM}^{t} | \boldsymbol{o}^{1:t})$ is assumed to be provided in factorized form,

$$p(\boldsymbol{X}_{TM}^{t}|\boldsymbol{o}^{1:t}) = \prod_{X \in \boldsymbol{X}_{TM}} p(X^{t}|\boldsymbol{o}^{1:t}),$$

where each pdf $p(X^t|\boldsymbol{o}^{1:t})$ over a continuous variable $X \in \boldsymbol{X}_{TM}^{t}$ is provided as a Normal distribution $p(X^t|\boldsymbol{o}^{1:t}) = N\left(X^t|\mu_X^t, {\sigma_X^t}^2\right)$, with mean and variance provided by the sensor and communication platform, while each probability mass function over a discrete variable $Y \in \boldsymbol{X}_{TM}^{t}$ is provided as a vector denoting the probabilities for each $y \in Y$. Where such information cannot be provided directly by the sensor and communication platform, it is expected to be derived by a semantic enrichment of the situation model.

Concerning other traffic participants and objects, let $m$ denote the number of recognized objects in the vicinity of the TeamMate vehicle $\boldsymbol{O} = (O_1, \dots, O_m)$, each object $O \in \boldsymbol{O}$ is assumed to be represented by a set of variables $\boldsymbol{X}_O^t = \{X_O^t, Y_O^t, \Theta_O^t, V_O^t, A_O^t, W_O^t, S_{L_O}^t, S_{W_O}^t, E_O^t, C_O^t, L_O^t\}$, described in Table 2 and provided as a belief state $p(\boldsymbol{X}_O^t|\boldsymbol{o}^{1:t})$. As for the TeamMate vehicle, for the first cycle, $p(\boldsymbol{X}_O^t|\boldsymbol{o}^{1:t})$ is assumed to be provided in factorized form. More specifically, let $\boldsymbol{X} \subset \boldsymbol{X}_O$ denote the set of continuous variables and $\boldsymbol{Y} \subset \boldsymbol{X}_O$ it is assumed that $p(\boldsymbol{X}_O^t|\boldsymbol{o}^{1:t})$ is given by

$$p(\boldsymbol{X}_O^t|\boldsymbol{o}^{1:t}) = \prod_{X \in \boldsymbol{X}} p(X^t|E_O^t = \text{true}, \boldsymbol{o}^{1:t}) \prod_{Y \in \boldsymbol{Y}} p(Y^t|\boldsymbol{o}^{1:t}),$$

where each pdf $p(X^t|E_O^t = \text{true}, \boldsymbol{o}^{1:t})$ over a continuous variable $X \in \boldsymbol{X}$ is provided as a Normal distribution $p(X^t|E_O^t = \text{true}, \boldsymbol{o}^{1:t}) = N\left(X^t|\mu_X^t, {\sigma_X^t}^2\right)$, with mean and variance provided by the sensor and communication platform, while each probability mass function over a discrete variable $Y \in \boldsymbol{Y}$ is provided as a vector denoting the probabilities for each $y \in \text{Val}(Y)$. For the most part, the information represented by $\boldsymbol{X}_O^t$ should be considered standard for current LIDAR sensors. Where such information cannot be provided directly by the sensor and communication platform, it is expected to be derived by a semantic enrichment of the situation model.

**Table 2: Description of variables for the representation of an object $\boldsymbol{O} \in \boldsymbol{O}$ in the vicinity of the TeamMate vehicle considered for the first cycle.**

| Variable | Type | Unit | Description |
|---|---|---|---|
| $X_O$ | Continuous | [m] | X-coordinate of the center of the object $O \in \boldsymbol{O}$ in a two-dimensional spatial coordinate system relative to the position of the TeamMate vehicle |
| $Y_O$ | Continuous | [m] | Y-coordinate of the center of the object $O \in \boldsymbol{O}$ in a two-dimensional spatial coordinate system relative to the position of the TeamMate vehicle |
| $\Theta_O$ | Continuous | [rad] | Yaw-angle relative to a reference axis |

| | | | |
|---|---|---|---|
| $V_O$ | Continuous | [m/s] | Longitudinal velocity along the objects heading |
| $A_O$ | Continuous | [m/s²] | Longitudinal acceleration |
| $W_O$ | Continuous | [rad/s] | Yaw-rate |
| $S_{L_O}$ | Continuous | [m] | Length (along the x-axis) |
| $S_{W_O}$ | Continuous | [m] | Width (along the y-axis) |
| $E_O$ | Binary | $\{\text{true,false}\}$ | Binary flag, whether the object $O \in \boldsymbol{O}$ exists in the current traffic scene. |
| $C_O$ | Discrete | $\{0, \dots, \lfloor C_O \rfloor\}$ | Classification of the object $O \in \boldsymbol{O}$, e.g. PKW, LKW, VRU, etc. |
| $L_O$ | Discrete | $\{0, \dots, \lfloor L_O \rfloor\}$ | The lane, the object $O \in \boldsymbol{O}$ is currently located in, e.g. fast or slow lane on a two-lane road |

### 2.3.3 Semantic enrichment of the situation model

The goal of the semantic enrichment is to extend the inputs from the perception layer with semantic information. For this purpose, we propose ontology extended with logical rules in the first cycle of this project. An ontology is a semantic model that represents domain knowledge using concepts and relations. A modelled ontology can be used to reason about new complex relations and facts. For this work, we used the *Web Ontology Language* (OWL) 2 (Motik & al., 2009) and the *Semantic Web Rule Language* (SWRL) (Horrocks & al., 2004) to model the ontology and logical rules. Figure 6 shows an overview of the taxonomy ("*has subclass*") and the relations we modelled in the ontology for this cycle using Protégé (Musen & al, 2015). Scene objects as pedestrian, vehicle, traffic light, traffic signal and road are concepts of this ontology. The relations between those scene objects are spatial, temporal and semantic.

These relations are divided into three main classes:

1. Assignment of roads/lanes to traffic participants ("*is_on*") as well as traffic lights and signals ("*street_has_trafficlight*", "*street_has_sign*") using map matching,
2. Assignment of traffic lights and signals to allowed maneuvers ("signal_maneuver") based on traffic rules,
3. Assignement of traffic participants allowed maneuvers ("allowed_maneuver") and maximal velocity ("has_max_speed_value") according to the traffic rules.

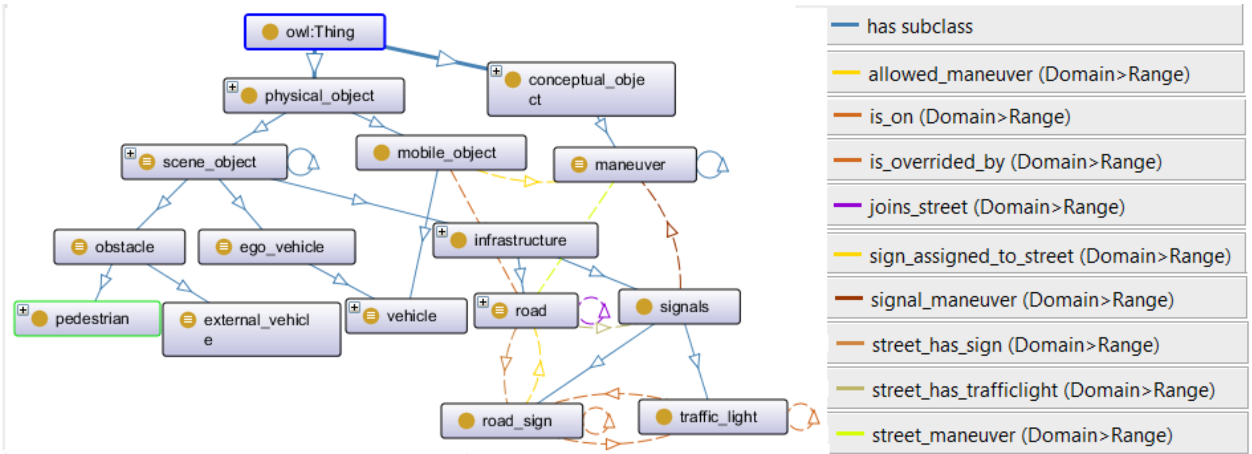



**Figure 6: Overview of the proposed ontology taxonomy (left) and the relations legend (right). See color version of the image for more details.**

Based on the relations and concepts of the ontology presented above, SWRL rules describing basics traffic rules are modelled. These SWRL rules cover following part of the traffic rules (see Table 3):

1. Definition of manoeuvres associated with traffic lights and signals (rules R1 to R3)
2. Definition of allowed manoeuvres on a road depending on the assigned traffic lights and signals (rules R4 and R5)
3. Definition of traffic participant allowed manoeuvres and maximal velocity depending on the road assigned to the traffic participant (rules R6 and R7)

**Table 3: examples of SWRL rules for basic traffic rules**

| Name | Rule | Meaning |
|---|---|---|
| R1 | $stop\_sign(?s) \wedge maneuver(?m) \wedge signal\_maneuver(?s,?m) \rightarrow stop(?m)$ | Stop sign allows stop maneuver |
| R2 | $give\_way\_sign(?s) \wedge maneuver(?m) \wedge signal\_maneuver(?s,?m) \rightarrow slow\_down(?m)$ | Give way sign allows slow maneuver |
| R3 | $traffic\_light(?l) \wedge has\_tl\_state(?l,?s) \wedge red\_light(?s) \wedge signal\_maneuver(?l,?m) \rightarrow stop(?m)$ | Red traffic light allows stop maneuver |
| R4 | $traffic\_light(?l) \wedge road(?r) \wedge road\_sign(?s) \wedge trafficlight\_assigned\_to\_street(?l,?r) \wedge sign\_assigned\_to\_street(?s,?r) \wedge signal\_maneuver(?s,?m) \wedge signal\_maneuver(?l,?m2) \rightarrow street\_maneuver(?r,?m2)$ | Traffic lights has high priority comparing to traffic signs, if both are assigned to the same road |
| R5 | $road(?r) \wedge road\_sign(?s) \wedge sign\_assigned\_to\_street(?s,?r) \wedge no\_trafficlight\_assigned\_to\_street(?r,true) \wedge signal\_maneuver(?s,?m) \rightarrow street\_maneuver(?r,?m)$ | maneuver allowed on that road depend on the assigned traffic sign where there is |

| | | |
|---|---|---|
| | | no traffic light |
| R6 | $mobile\_object(?o) \wedge road(?r) \wedge street\_maneuver(?r, ?m) \wedge is\_on(?o, ?r) \rightarrow allowed\_maneuver(?o, ?m)$ | Traffic participants allowed maneuvers depend on the road there are on |
| R7 | $road(?r) \wedge road\_vehicle(?o) \wedge is\_on(?o, ?r) \wedge has\_max\_speed\_value(?r, ?v) \rightarrow has\_max\_speed\_value(?o, ?v)$ | Vehicles allowed maximal velocity depend on the road there are on |

For testing the modelled ontology and traffic rules, we generated the scene in Figure 7. In this scene, the ego-vehicle (red) is approaching an intersection, where the traffic light is red. Another vehicle is approaching the intersection on a lane with green light. Based on this scene, individuals are generated for:

- the vehicles ("*Vehicle(?vh1)*" and "*Vehicle(?vh1)*" ),
- the traffic lights ("*Traffic_light(?tl1)*" and "*Traffic_light(?tl2)*"),
- the traffic lights states ("*Red_light(?tlst1)*"and "*Green_light(?tlst2)*"),
- the maneuvers ("*Maneuver(?mn1)*"and "*Maneuver(?mn2)*") which will be inferred, and
- the roads ("*Road(?street1)*", "*Road(?street2)*")

Furthermore traffic lights and vehicles are matched to the corresponding roads using the relations "*Is_on(?vh1,?street1)*", "*Is_on(?vh2,?street2)*", "*Trafficlight_assigned_to_street(?tl1,?street1)*" and "*Trafficlight_assigned_to_street(?tl2,?street2)*". Traffic lights states are set using the relations "*Has_tl_state(?tl1,?tlst1)*" and "*Has_tl_state(?tl2,?tlst2)*". The "*Pellet*" reasoner available on Protégé infers the allowed maneuvers for each vehicle individual according to the traffic rules. The allowed maneuver "*Allowed_maneuver(?vh1,?mn1)*" inferred for "*Vehicle(?vh1)*" is "*Stop(?mn1)*", meaning that this vehicle must stop due to the red light. For "*Vehicle(?vh2)*" the inference result "*Drive(?mn2)*" allows this vehicle to drive since the traffic light assigned to the road this vehicle is driving on is green. Based on the inference results, we can conclude that the ontology and logical rules can be used to infer traffic participants allowed maneuvers according to the traffic rules.

The allowed manoeuvres inferred by the reasoner will be used in the second project cycle for predicting the traffic evolution. For that, we will develop an interface allowing us to integrate the ontology, the logical rules and the reasoner results into the situation interpretation module working on real traffic data.

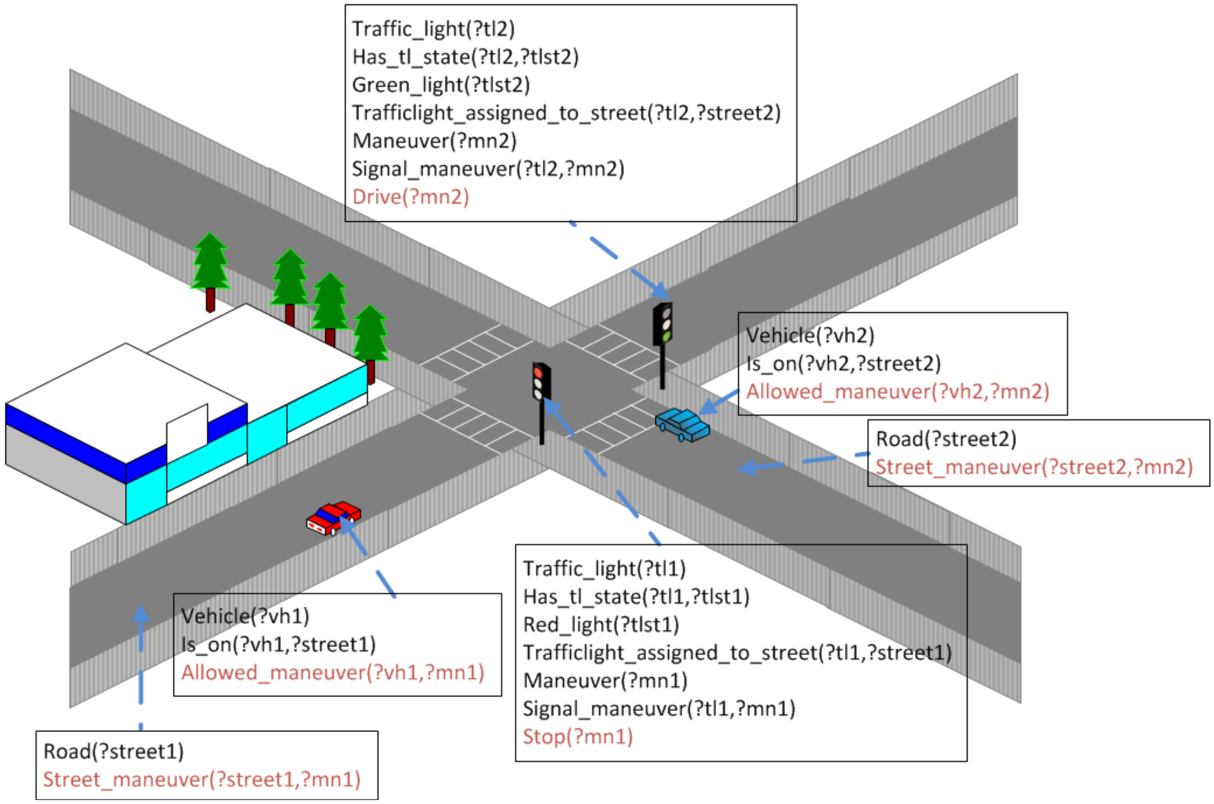



**Figure 7: illustration of the semantic erichment at an intersection based on the ontology and logical rules.**

### 2.3.4 Predicting the future evolution of the traffic scene

The purpose of vehicle models is to predict the future evolution of the traffic scene based on the information represented by the situation-model and the use of vehicle-models as a necessary input for online risk assessment (for more information on online risk assessment, we refer to the deliverable D3.3 "Concepts and algorithms incl. V&V results from 1st cycle").

#### 2.3.4.1 Vehicle-Models

In this context, vehicle models should be understood as motion models. Based on a comparison and evaluation of motion models for vehicle tracking (Schubert et al., 2008) and their successful use for risk assessment for collision avoidance systems (Houenou et al., 2013, 2014), we use the so-called *Constant Turn Rate and Acceleration* (CTRA) (Schubert et al., 2008), resp. *Constant Yaw-Rate and Acceleration* (CYRA) motion model (Houenou et al., 2013, 2014). The CYRA model is based on a state space

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

where $x$ and $y$ (in $m$) denote the spatial coordinates of the center of the vehicle, $\theta$ (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. The state transition equation for this model is given by

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

with

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

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

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

and

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

#### 2.3.4.2 Unscented Transformation

In the following, let $\boldsymbol{S}_O^t = \{X_O^t, Y_O^t, \Theta_O^t, V_O^t, A_O^t, W_O^t\}$ denote a reduced set of state variables for an object $O \in \boldsymbol{O}$ and let $p(\boldsymbol{S}_O^t | E_O^t = \text{true}, \boldsymbol{o}^{1:t})$ denote our beliefs about $O$, given all observed sensor values up to the current point in time, and given that said object $O$ is actually existing. We can obtain a prediction for a future time step $p(\boldsymbol{S}_O^{t+\Delta t} | E_O^t = \text{true}, \boldsymbol{o}^{1:t})$ via unscented transformation (Wan and Van der Merwe, 2000, Murphy, 2012). The basic idea is as follows: Under the assumption that $p(\boldsymbol{S}_O^t | E_O^t = \text{true}, \boldsymbol{o}^{1:t})$ is a multivariate Gaussian $N(\boldsymbol{S}_O^t | \boldsymbol{\mu}_O^t, \Sigma_O^t)$, we'd like to estimate $p(\boldsymbol{S}_O^{t+\Delta t} | E_O^t = \text{true}, \boldsymbol{o}^{1:t})$ as a multivariate Gaussian $N(\boldsymbol{S}_O^{t+\Delta} | \boldsymbol{\mu}_O^{t+\Delta}, \Sigma_O^{t+\Delta})$, where $\boldsymbol{S}_O^{t+\Delta} = f_{CYRA}(\boldsymbol{S}_O^t)$, with $f_{CYRA}$ being the nonlinear function given by the CYRA motion model. Following Murphy (2012), let $d = 6$ denote the dimension of the multivariate Gaussian, we create a set of $2d+1$ *sigma vectors* $\boldsymbol{s}_i^t$, where

$$\boldsymbol{s}_0^t = \boldsymbol{\mu}_O^t,$$

$$\boldsymbol{s}_i^t = \boldsymbol{\mu}_O^t + \left(\sqrt{(d+\lambda)\Sigma_O^t}\right)_{:i}, i = 1, \ldots, d,$$

$$\boldsymbol{s}_i^t = \boldsymbol{\mu}_O^t - \left(\sqrt{(d+\lambda)\Sigma_O^{\mathrm{t}}}\right)_{:i-d}, i = d+1, \dots, 2d,$$

and a corresponding set of $2d+1$ sigma weights for both mean $w_{m,i}$ and covariance $w_{c,i}$, where

$$w_{m,0} = \frac{\lambda}{d+\lambda},$$
$$w_{c,0} = \frac{\lambda}{d+\lambda} + (1 - \alpha^2 + \beta),$$
$$w_{m,i} = w_{c,i} = \frac{1}{2(d+\lambda)}, i = 1, \dots, 2d.$$

Here, $\left(\sqrt{(d+\lambda)\Sigma_O^{\mathrm{t}}}\right)_{:i}$ denotes the $i$th column of the (scaled) square-root matrix of $\Sigma_O^{\mathrm{t}}$, $\lambda = \alpha^2(d+k) - d$ is a scaling parameter, with $\alpha$ and $k$ being corresponding parameters that determine the spread of sigma vectors around the mean, while $\beta$ can be used to incorporate prior information on (non-Gaussian) distributions. For $d = 1$, Murphy (2012) states optimal values as $\alpha = 1$, $\beta = 0$, and $k = 2$, which we adopt for unscented transformation in AutoMate for the time being. We propagate these sigma vectors through the nonlinear function to obtain a transformed set of sigma vectors $\boldsymbol{s}_i^{t+\Delta}$:

$$\boldsymbol{s}_i^{t+\Delta t} = f_{CYRA}(\boldsymbol{s}_i^t).$$

The mean $\boldsymbol{\mu}_O^{t+\Delta}$ for $N(\boldsymbol{S}_O^{t+\Delta} | \boldsymbol{\mu}_O^{t+\Delta}, \Sigma_O^{\mathrm{t}+\Delta})$ is then computed from this transformed sigma vectors as

$$\boldsymbol{\mu}_O^{t+\Delta} = \sum_{i=0}^{2d} w_m^i \boldsymbol{s}_i^{t+\Delta},$$

and its covariance $\Sigma_O^{\mathrm{t}+\Delta}$ is given by

$$\Sigma_O^{\mathrm{t}+\Delta} = \sum_{i=0}^{2d} w_c^i \left(\boldsymbol{s}_i^{t+\Delta} - \boldsymbol{\mu}_O^{t+\Delta}\right)\left(\boldsymbol{s}_i^{t+\Delta} - \boldsymbol{\mu}_O^{t+\Delta}\right)^T.$$

Given this, let $n$ denote the desired prediction horizon, we predict the future evolution of the traffic scene, by estimating $p\left(\boldsymbol{S}_O^{t+i\Delta t} | E_O^t = \mathrm{true}, \boldsymbol{o}^{1:t}\right), i = 1, \dots, n$ for each object $O \in \boldsymbol{O}$ known to the TeamMate vehicle.

For the first cycle, the prediction of the future evolution of the traffic scene makes strong simplifications. The vehicle models are based on the assumptions of constant accelerations and yaw-rates. By now, the algorithm does not incorporate knowledge about the map $M$ into our predictions of the future state of other objects, i.e., the algorithm assumes that everything evolves statically. These limitations will be addressed in future cycles.

### 2.3.5 V+V of Vehicle and Situation Models

As of now, the prediction of the temporal and spatial evolution of the traffic scene via the CYRA vehicle model (Section 2.3.4.1) has been implemented for surrounding traffic participant under the assumption that the necessary input can be provided (Figure 8). The corresponding functionality has been tested using inputs provided by the SILAB simulation environment used at the OFFIS Institute for Information Technology.


**Figure 8: Screenshot of an exemplary visualization of the 95% prediction ellipses of the position of the lead vehicle (red). The coloured rectangles represent vehicles in the vicinity of the TeamMate vehicle (white rectangle). Blue lines indicate heading vectors, the purple line represents the centreline of a two-lane motorway.**

For validation purposes, we tested the preliminary "correctness" of the vehicle models on data sets obtained in simulator driving studies in the SILAB simulation environment. The data set comprises a time-series of 295123 training samples, recorded with a frequency of 60Hz, with each sample containing the data representing the necessary input of up to eight vehicles in the vicinity of the TeamMate vehicle, up to two vehicles on the current and adjacent lanes, both in front and behind the TeamMate vehicle. Let $\boldsymbol{s}_v^t = \{x_v^t, y_v^t, \theta_v^t, v_v^t, a_v^t, w_v^t\}$ denote the ground truth of the state of a vehicle $v$ in the vicinity of the TeamMate vehicle at a time $t$ in the data set, and $c^t$ denote the current curvature of the road, we used the following estimate for our initial belief state $p(\boldsymbol{S}_v^t | E_v^t = \text{true}, \boldsymbol{o}^{1:t})$:

$$p(\boldsymbol{S}_v^t | E_v^t = \text{true}, \boldsymbol{o}^{1:t}) = N\left(\boldsymbol{\mu} = \begin{pmatrix} x_v^t \\ y_v^t \\ \theta_v^t \\ v_v^t \\ 0 \\ v_v^t * c^t \end{pmatrix}, \boldsymbol{\Sigma} = \begin{pmatrix} 0.05^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.05^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.0^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1^2 \end{pmatrix}\right).$$

Due to the nature of simulated traffic realized by a kind of bang-bang controller, we replaced the true acceleration by zero and the true yaw rate the required yaw rate to follow the course of the road (if aligned with the road), but added a high uncertainty on the actual estimate.

At each time step $t$ and each vehicle $v$ within the sensor range of the TeamMate vehicle, we estimated the belief states of the future state of the vehicle $p(\boldsymbol{S}_v^{t+i} | E_v^t = \text{true}, \boldsymbol{o}^{1:t})$ for a set of prediction horizons of $i = 1, \ldots, 10$

seconds, from which we derived marginalized two-dimensional belief states $p\left(X_v^{t+\mathrm{i}}, Y_v^{t+i} | E_v^t = \mathrm{true}, \boldsymbol{o}^{1:t}\right)$ and marginalized three-dimensional belief states $p\left(X_v^{t+\mathrm{i}}, Y_v^{t+i}, \Theta_v^{t+i} | E_v^t = \mathrm{true}, \boldsymbol{o}^{1:t}\right)$.

At each subsequent time step $t+i$, we then checked, whether the true marginalized state of the vehicle $\left(x_v^{t+i}, y_v^{t+i}, \theta_v^{t+i}\right)$, resp. $\left(x_v^{t+i}, y_v^{t+i}\right)$ was located within the 50%, 90%, 95%, and 99% prediction ellipses derived from marginalized two-dimensional belief states $p\left(X_v^{t+\mathrm{i}}, Y_v^{t+i} | E_v^t = \mathrm{true}, \boldsymbol{o}^{1:t}\right)$ and marginalized three-dimensional belief states $p\left(X_v^{t+\mathrm{i}}, Y_v^{t+i}, \Theta_v^{t+i} | E_v^t = \mathrm{true}, \boldsymbol{o}^{1:t}\right)$. For the two-dimensional belief states, we furthermore calculated the mean cartesian distance between the actual position $\left(x_v^{t+i}, y_v^{t+i}\right)$ and the expected position $E\left[X_v^{t+\mathrm{i}}, Y_v^{t+i} | E_v^t = \mathrm{true}, \boldsymbol{o}^{1:t}\right]$. The resulting data was aggregated over all different vehicles to derive the percentage of vehicles outside the corresponding prediction ellipse for each temporal prediction horizon. The results are summarized in Table 4. We note that limited (simulated) sensor range of ∓200m for the detection of surrounding vehicles and a detection based on the spatial relation between the different vehicles, make it possible that a vehicle was outside the sensor range prior to entering temporal intervals, leading to a reduction of counts as apparent in Table 4.

**Table 4: Validation results for the use of implemented vehicle models for predicting the spatial and temporal evolution of the traffic scenes at different future time steps. Bracketed percentages outside prediction ellipses were obtained by comparing the three-dimensional states.**

| Prediction Horizon | Number of samples | Percentage Outside 50% Prediction Ellipse | Percentage Outside 90% Prediction Ellipse | Percentage Outside 95% Prediction Ellipse | Percentage Outside 99% Prediction Ellipse | Mean Cartesian Distance |
|---|---|---|---|---|---|---|
| 1s | 1239231 | 5.664 (8.230) | 3.616 (6.056) | 3.206 (5.568) | 2.529 (4.667) | 2.075 |
| 2s | 1213792 | 5.957 (12.347) | 2.709 (8.814) | 2.237 (8.316) | 1.849 (7.584) | 4.730 |
| 3s | 1190865 | 4.165 (15.630) | 2.204 (11.964) | 1.891 (11.391) | 1.445 (10.526) | 8.022 |
| 4s | 1170298 | 3.423 (19.942) | 1.643 (14.881) | 1.329 (14.050) | 0.933 (12.885) | 12.157 |
| 5s | 1151489 | 2.974 (26.991) | 1.167 (20.054) | 0.958 (18.858) | 0.638 (17.157) | 17.209 |
| 6s | 1133906 | 2.457 (37.342) | 0.890 (28.780) | 0.689 (27.125) | 0.421 (24.615) | 24.650 |
| 7s | 1116738 | 1.990 (48.402) | 0.670 (39.689) | 0.501 (37.814) | 0.334 (34.770) | 33.403 |
| 8s | 1099544 | 1.659 (55.335) | 0.522 (47.066) | 0.403 (45.217) | 0.252 (42.186) | 42.430 |
| 9s | 1082839 | 1.441 (56.384) | 0.436 (47.056) | 0.318 (45.014) | 0.247 (41.721) | 51.093 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 10s | 1066314 | 1.257 (57.506) | 0.347 (45.965) | 0.291 (43.497) | 0.273 (39.643) | 60.405 |

As apparent, the percentage outside the dedicated prediction ellipse for the three-dimensional states quickly exceeds the expected percentage with extended prediction horizons. Furthermore, although the percentages for the two-dimensional states are mostly below the expected percentages and fall with exceeded prediction horizons, this result is only achieved by a corresponding inflation of the prediction ellipses, as indicated by the mean Cartesian distance (c.f. Figure 8).

For future cycles, we will test to improve these preliminary results by incorporating knowledge and expectations of the future course of the road, potential manoeuvres, and potential interactions between vehicles.
The tests protocols, the vehicles or simulator and the scenarios are still to be defined in discussion with the partners.
It is already planned that the situation model and the driver state will be integrated and test in the VED demonstrator, these two modules will enrich the feasibility of the system in the case of the project and will help to ensure good transitions between the driver and the teammate system. VEDECOM will focus on the Martha scenario, this scenario will be tested, first on the Satory High speed track with an emulation of real situations and on an open road (the A86 highway) under constraint of having the legal authorizations from the local authorities. The scenario will be tested on 40 participants. In a second hand, if time permits it, we are interested to test the urban use cases.

# 3 Instantiation of the Automate platform

The three following scenarios instance the automate platform. Here a general description of the scenarios is provided; more details are available on D1.1 "Definition of framework, scenarios and requirements" and, for a refinement of Cycle 2, in D1.3 "Definition of framework, scenarios and requirements" (currently in progress):

- User scenario 1

| **User Scenario 1: Peter** | **Driver out of the loop, manoeuvre becomes necessary** | **Rural Road** |
|---|---|---|
| On a rural road, a driver is reading in full automation when a large vehicle makes an evasive manoeuvre necessary. | | |
| **TeamMate Car Functionality and Added Value** | | |
| (1) Situation understanding; (2) Anticipation of unsafe system predictions; (3) Decision | | |

| |
|---|
| that support from driver is needed; (4) "Explaining" situation to driver; (5) Generating optional interventions; (6) Detecting driver state; (7) Check driver decision for overtaking; (8) Planning and monitoring of overtaking manoeuvre |

- User scenario 2:

| **User Scenario 2: Martha** | **Take-over of automation after driver distraction** | **Motorway** |
|---|---|---|
| While driving manually, a driver suddenly receives a distracting message and the system takes over. | | |
| **TeamMate Car Functionality and Added Value** | | |
| (1) Driver monitoring with attention detection; (2) Driver recognition (distracted driver); (3) Anticipation of unsafe predictions due to distraction; (4) Decision that driver needs help; (5) Communication about situation to driver; (6) Adaptive communication and hand-over strategy; (7) Hand-over from manual driving to fully automated; (8) Escalating hand-over strategy with driver monitoring. | | |

- User scenario 3:

| **User Scenario 3: Eva** | **Learning to efficiently manage a roundabout** | **City Traffic** |
|---|---|---|
| A TeamMate Car is driving through a complex roundabout with different traffic and driving status conditions (i.e. risky driving situation (i.e. hidden pedestrian crossing), high/low driver workload).<br>By driving through a complex roundabout several times, the system learns from the driver how to deal with it efficiently and how to manage hand-over situation between human and automated system efficiently. | | |
| **TeamMate Car Functionality and Added Value** | | |
| (1) Driver monitoring; (2) Situation recognition; (3) Manoeuvre planning under uncertainty; (4) Safety assessment & decision that help from driver is needed; (5) Communicate situation to driver; (6) Handover from automated to manual driving; (7) Solution recording; (8) Deduction of general solution; (9) Learning of new solution | | |

## 3.1 User scenario 1 (Peter)

The ULM demo car will be used to implement, verify and validate the user scenario 1. A part of the verification and validation will be conducted in the ULM driving simulator. It is a static driving simulator which runs the SILAB driving simulation software. This software makes the simulation of any degree of driving automation, needed to conduct verification studies, possible. Each automation feature (e.g. ACC, lateral control etc.) can be turned off if needed. The sensors can also be simulated in accordance with experimental needs. Different driving variables can be recorded while driving in the simulator. These include lateral and longitudinal control, the interaction with the simulator, environmental and traffic parameters. The driver can interact with the car via the standard in vehicle instruments (steering wheel, indicators and pedals) or via a touch screen, which is integrated in the central stack of the vehicle mock-up.

### 3.1.1 Sensors and communication platform of ULM demo car

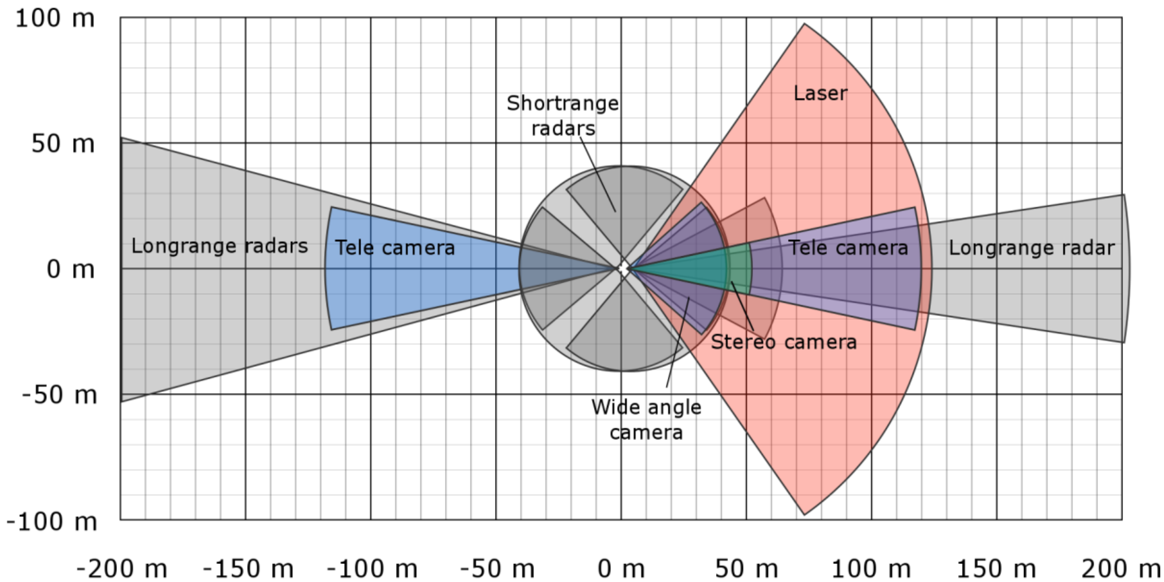
The road structure of the test-route is stored into a pre-recorded digital map using UTM coordinates. To extract relevant data from this map, the cars position must be localised. Free-space is located in between the boundary lines of the current lane.

Stopping due to dynamic obstacles is performed by closing the boundary lines to restrict the free-space. For a more detailed description see (Kunz & al, 2015).

Environment representation:
- Pre-recorded data:
  - Reference line as input for trajectory planning, e.g. center of lane (polygonal line, UTM coordinates).
  - Speed limits assigned to each lane
  - Static stop points, for example stop signs (stop must be performed!).

- Online calculated data
  - Boundary lines (2 polygonal lines, UTM coordinates).
  - Volatile stop points (due to static obstacles on the road)
  - Current position and predicted trajectories of other vehicles and pedestrians (spatial data: UTM coordinates and object dynamics, e.g., velocity and acceleration)
  - State of traffic lights

In Addition, the sensor setup of the demonstrator is shown in the figure below. This consists of two long-range radars, as well as cameras for the view near front and rear, as well as an additional stereo-camera, a laser-scanner and four short-range radars.




## 3.2 User Scenario 2 (Martha)

VED will use both its car and its simulator to implement, integrate and validate the user scenario Martha. VED already provides a demonstrator vehicle that is capable of autonomous driving in urban area; however, the Martha scenario requires a high-speed vehicle capable to deal with highway situations. We are actually equipping a new vehicle to implement our algorithms and the algorithms of the project.

### 3.2.1 Sensors and communication platform of VED demo car

The VED Automate sensor platform is composed of the following sensors:

- Monocular cameras
- 5 Lidars + a fusion system
- 1 Long Range Radar
- 1 Global Navigation Satellite System (GNSS)
- 1 Inertial Measurement Unit (IMU)
- 1 multi-bandwidth communication platform (wifi, 4G, internet, 802.11p)

The following table summarizes the situation of the most important sensors for the actual VEDECOM test-car, which are in accordance with the project:

| Function | Tech-nology | Supplier / Model | Op. Freq. / Wavel. | Range | Hor. field of view | Update rate | Interface |
|---|---|---|---|---|---|---|---|
| Front objects | Radar | Continental ARS 408 | 77 GHz | 200 m | 17° | 16 Hz | CAN |
| Surrounding objects | 5 Lidars | IbeoLux | | 100 m | 360° | 25 Hz | Ethernet TCP/IP |
| Vehicle Position | GNSS | IXEA + Septentrio | L1/L2 | | | 25 Hz | RS232+PPS |
| Vehicle Position – lateral (lane marking) | Camera | VEDECOM | // | 60m | 60° | 25 Hz | Ethernet |

In addition to the sensors we have several algorithms running:

- Obstacle detection tracking and fusion: In this part each surrounding obstacle is detected and tracked over time and its state vector is returned at each step of computation.
- Lane marking detection: based on a monocular camera, we have developed a robust algorithm of lane marking which detect the lines and their typology (dashed, continuous ...etc.)
- Path following and control algorithms: our car is able to follow a path, this path can be either computed from the results of the perception step or replay a recorded path coming from the IMU/GPS or a SLAM algorithm (in urban areas.).

To deal with high vehicle speed a new demo car is under construction for Automate. The equipment will have the same environmental sensing capabilities. The Vedecom Demo Car has actually a V2X interface based on a multiband architecture (802.11p, 3G, 4G, Internet). This interface is compatible with the latest ETSI norms. In addition to that, a network between the different systems, ECU's and computers ensures the communication inside the car. Tests will be performed to check the operability between proposed sensors and the existing platform. The current sensor platform will be then enhanced with operable sensors and communication layer
The Vedecom sensor platform will also include a driver's state sensor required for detecting the driver's distraction.

## 3.3 User Scenario 3 (Eva)

The CRF and REL will use a driving simulator to implement, verify and validate the user scenario 3. Three main reasons for that:

- Possibility to consider also high-critical situations and scenarios – such as roundabout in urban scenarios – without safety concerns.

- Experiments involving real users, with the advantage to explore several HMI solutions, as well as to investigate acceptability and usability issues (in a more flexible and effective way).
- Possibility to focus the attention on the interaction between human-agent and machine-agent, thus on the decision aspects, not affecting by perception problems.

In addition, a test-vehicle (from AdaptIVe EU project) is available to collect data from real-worlds and to test single and specific modules/components (e.g. driver monitoring system from CAF, on-line risk assessment module, driver model, and so on). It is not foreseen to experiment the Eva scenario with this vehicle. Main reasons are related with HMI integration issues and the fact that we cannot test this vehicle in real-roads and to make exhaustive tests on a dedicated test-track is not fully representative. Besides on the driving simulator we are free to investigate also high critical situations (fully exploiting the potential benefit of the team-mate car) not possible with a real-vehicle.

In the next paragraphs, we will focus on the sensorial system available on the test-vehicle, which is built on the basis of a Jeep Renegade, with robotised gearbox. This is because this model already offers some components and functions that are useful for the automatic system developed in the project. Following the approach of the layered architecture, the automatic system uses as much as possible of production vehicle components, adding redundancies, extra information sources and driver interaction channels to what is already available in production.

### 3.3.1 Sensors and communication platform of the CRF test-car

The following table shows the situation for the CRF test-car (Bisoffi et al., 2015):

| Function | Tech-nology | Supplier / Model | Op. Freq. / Wavel. | Range | Hor. field of view | Update rate | Interface |
|---|---|---|---|---|---|---|---|
| Front objects and Lane | Radar + Camera | Delphi RACam 1.0 | 77 GHz | 100 m | 100° | 20 Hz | Ethernet UDP |
| Front objects | Lidar | Valeo ScaLa | 905 nm | 150 m | 145° | 25 Hz | Ethernet TCP/IP |
| Vehicle Position | GNSS | NovAtel Flex6 | L1/L2 | | | 20 Hz | RS232+PPS |
| Side/rear objects | Radar | Autoliv SR radar | 24 GHz | 14 m | 100° | 20 Hz | CAN |
| Side/rear objects | Ultrasound | Series production | ultrasound | 5 m | 60° | 100 Hz | CAN |

It is worth to noting here that the “range” parameter is referred to the target constituted by a car.

For what concerning the actuators used to control vehicle motion in automatic mode (that is brake, steering, and the engine), they are the same as available on the production car model, but with specific SW update in the control ECUs, in order to be able to accept possible control requests from the autonomous system.

To conclude, a final remark: what is described before is true for the test-vehicle, not necessarily for the driving simulator, where the scenario is specifically built and thus ADAS sensors are “simulated” as well.

# 4 Conclusion

In cycle1, we have defined the demonstrator sensor platforms according to the scenarios requirements. These are either new demonstrators or existing demonstrators, which will be updated with the additional required sensors. This concerns mainly the driver’s state sensor and the V2V, V2X sensors.

Regarding the driver modelling we have produced lists of goals and operators that can now be used to model driver-vehicle-interactions. We defined and implemented a template probabilistic driver model for intention recognition and behaviour assessment, whose fine-level structure and parameter can be learned from time-series of human driving data.

The work on situation modelling focused on the semantic enrichment of the situation model and the prediction of the situation evolution. The semantic enrichment of the situation model based on the data provided by the perception layer was implemented using an ontology and logical rules. First vehicle models based on the CYRA motion model have been implemented and can be used for the prediction of the temporal and spatial evolution of the traffic scenes required by online risk assessment.

In this first cycle a study for learning, verification and validation of the driver’s intention recognition and driver’s situation awareness has been performed. It includes 48 subjects driving the Peter scenario on two driving simulators equipped with an eye-tracking system. Although focussing on the Peter scenario for the moment, the resulting model is then planned to be adapted to provide behaviour assessment in the Martha and Eva scenarios. Potential required collection of additional experimental data for this adaptation will be discussed with the related teams.

In the next cycle, we will start with integrating the sensors in the demonstrator platforms and further verification and validation test will then be performed at system level. The driver's state sensor hardware will be finalized and improved models will be integrated.

We will model the driver behaviour and driver interaction with an automation based on the empirical data, and use this knowledge to make suggestions regarding the design of the TeamMate car. Based on the experimental data obtained in the first cycle, the probabilistic driver models for intention recognition and behaviour assessment will be trained and validated. Furthermore, the semantic enriched situation model will be extended with an interface to provide inputs to the module for predicting the situation evolution, and the vehicle models will be extended to incorporate knowledge about the future course of the road and potential future manoeuvres.

Verification and Validation of the situation model and driver's model will be done in the different demonstrator vehicles with the related scenarios. Experiments with the Martha scenario are planned to test both models including driver's state sensor. The experiments will be performed with the VED vehicle first in a high speed track with an emulation of real situations and on an open road (the A86 highway) under constraint of having the legal authorizations from the local authorities. The next cycle experiments for the Eva and Peter scenarios are planned and still under discussion with the related teams.

# 5 References


Andrew Ng. Machine Learning - (free online course on Coursera). Available at https://fr.coursera.org/learn/machine-learning/lecture/QGKbr/model-selection-and-train-validation-test-sets

Barber, S. (2012). Bayesian Reasoning and Machine Learning. Cambridge University Press. Online version 02.2017: http://web4.cs.ucl.ac.uk/staff/D.Barber/textbook/020217.pdf

Bisoffi, A. (UNITN), Biral, F. (UNITN), Da Lio, M. (UNITN), Zaccarian, L. (UNITN): Longitudinal Jerk Estimation for Identification of Driver Intention in Proceedings of the 18th International IEEE Conference on Intelligent Transportation Systems - ITSC 2015, Date Added to IEEE Xplore: 02 November 2015.

Boverie S., Cour M. (2011) Adapted Human Machine Interaction concept for Driver Assistance Systems, *IFAC Proceedings Volumes, 44*(1), 2242–2247.

Boverie S., Giralt A. (2008) Driver vigilance diagnostic based on eyelid movement observation. *IFAC Proceedings Volumes, 41*(2), 12831–12836.

Card, Moran & newel). The Psychology of Human Computer Interaction. Lawrence Erlbaum Associates (1983)

Horrocks, I., F. Patel-Schneider, P., Boley, H., Tabet, S., Grosof, B., Dean, M.: SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Member submission 21 (2004)

Horberry, T., Anderson, J., Regan, M. A., Triggs, T. J., and Brown, J., "Driver distraction: the effects of concurrent in-vehicle tasks, road environment complexity and age on driving performance," Accident; analysis and prevention, vol. 38, no. 1, pp. 185–191, 2006.

Houenou, A., Bonnifait, P., Cherfaoui, V., and Wen, Y. (2013). Vehicle Trajectory Prediction based on Motion Model and Maneuver Recognition. Proceedings of the 2013 IEEE/RSJ International Conference of Intelligent Robots and Systems, pp. 4363-4369.

Houenou, A., Bonnifait, P., and Cherfaoui, V. (2014). Risk Assessment for Collision Avoidance Systems. Proceedings of the IEEE 17th International Conference on Intelligent Transportation Systems, pp. 386-391.

Dr Neale Kinnear and Dr Alan Stevens, TRL. "The battle for attention Driver distraction – a review of recent research and knowledge". IAM Driver Distraction Report, 2015

Kunz, F.; Nuss, D.; Wiest, J.; Deusch, H.; Reuter, S.; Gritschneder, F.; Scheel, A.; Stuebler, M.; Bach, M.; Hatzelmann, P.; Wild, C. & Dietmayer, K. (2015), Autonomous driving at Ulm University: A modular, robust, and sensor-independent fusion approach., in 'Intelligent Vehicles Symposium' , IEEE, , pp. 666-673 .

Liang Y., Reyes M.L. and Lee J.D.: Real-time Detection of Driver Cognitive Distraction Using Support Vector Machines. In Proc. IEEE Transactions on Intelligent Transportation 2007

Motik, B., Patel-Schneider, P.F., Parsia, B., Bock, C., Fokoue, A., Haase, P., Smith, M.: OWL 2 web ontology language: Structural specification and functional-style syntax. W3C recommendation 27.65 (2009)

Murphy, K. P. (2012). Machine Learning – A Probabilistic Perspective. The MIT Press, Cambridge, Massachusetts, London, England.

Musen, M.A., the Protégé Team: The Protégé Project: A Look Back and a Look Forward. AI matters 1, 4–12 (2015)

Polychronopoulos, A., Amditis, A., Scheunert, U., Tatschke, T. (eds.): Revisiting JDL model for automotive safety applications: the PF2 functional model. 2006 9th International Conference on Information Fusion (2006)

Schubert, R., Richter, E., and Wanielik, G. (2008). Comparison and Evaluation of Advanced Motion Models for Vehicle Tracking. Proceedings of the 11th International Conference on Information Fusion, pp. 730-735.

Wan, E. and Van der Merwe, R. (2000). The Unscented Kalman Filter for Nonlinear Estimation. Proceedings of the IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000, pp. 153-158.

Ziya Cihan & Ali Gokhan Yavuz (2013) ETSI compliant GeoNetworking protocol layer implementation for IVC simulations *Human-centric Computing and Information Sciences Springer*