D2.1 – Metrics and Experiments for V & V of the driver, vehicle and situation models in the 1st cycle

Project Number:	690705
Classification	Public
Deliverable No.:	D2.1
Work Package(s):	WP1
Document Version:	Vs. 1.0
Issue Date:	31.01.2017
Document Timescale:	Project Start Date: September 1, 2016
Start of the Document:	Month 2
Final version due:	Month 5
Compiled by:	N. Fricke, ULM
Authors:	N. Fricke, ULM J. Pichen, ULM M. Graf, ULM M. Eilers, OFF A. Giralt, CAF P. Pekezou Fouopi, DLR D. Käthner, DLR Fabio, Tango, CRF
Technical Approval:	Fabio Tango, CRF
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
18.11.2016	Deliverable structure	N.Fricke (ULM)
05.12.2016	Initial contribution of OFF	M. Eilers (OFF)
06.12.2016	Initial contribution of CAF	A. Giralt (CAF)
07.12. 2016	Version 0.2	N. Fricke (ULM)
12.12.2016	Initial contribution of DLR	P. Pekezou Fouopi (DLR)
16.12.2016	Contribution concerning task model	D. Käthner (DLR)
19.12.2016	Initial contribution of CRF	F. Tango (CRF)
20.12.2016	Version 0.3	N. Fricke (ULM)
18.01.2017	Version 0.4: Integration of partners revisions for internal review	N. Fricke (ULM)
25.01.2017	Version 0.5: Integration of reviews	N. Fricke (ULM)
26.01.2017	Minor revision of the introduction	M. Eilers (OFF)
20.01.2017	Final version	N. Fricke (ULM)

Table of Contents

1 Introduction

The concept of the TeamMate car requires solutions to monitor, understand, assess, and anticipate the driver, the vehicle, and the overall traffic situation. In AutoMate these solutions will be provided by *driver*-, *situation*- and *vehicle* models.

Driver models provide estimations of the hidden state, intentions and behavior of the human driver. As such, they can be used to provide the TeamMate car with information about the current state of the driver, e.g. whether he is distracted or tired and predict the current intentions and future behavior in the case of manual control, or predictions of potential driver intentions in the case of autonomous control.

Situation models serve the purpose to derive and represent a coherent snapshot of the current state of the world, i.e. the traffic environment, from the (usually noisy) sensor measurements provided by the sensor and communication platform in terms of LIDAR signals, GPS coordinates, video images, digital maps, etc. Such a world state should be understood as the observable physical traffic situation, including e.g. the future course of the road, and the position, velocities and accelerations of all surrounding traffic participants.

Whereas the situation models provide a static snapshot of the current traffic situation, vehicle models will be used to predict the future state and behavior of the different traffic participants. This can include both abstract models to predict the behavior of surrounding traffic participants, e.g. treating them as "lifeless" objects in space that retain their current direction and velocities, and more elaborate physical models of the TeamMate car as a means to predict the physical effects of potential control actions, needed e.g. for the planning of safe trajectories in the case of autonomous control or for assessing the risk of predicted behavior of the human driver.

Together driver-, situation- and vehicle models will enable the TeamMate car to reconstruct a coherent representation of the overall state of the world, including the ability to estimate likely spatial and temporal evolutions necessary for safe planning and control of autonomous control actions. The driver-, vehicle- and situation models will be developed using various modelling techniques. In the following sections the different models will be described as a basis for the definition of required metrics and measures for model validation.

1.1 Driver Models

To design and evaluate functionalities for automated driving which involve the human driver, a deep understanding of the driving task itself is urgently needed. Human models, especially cognitive models, can give fine-grained insight into specific situations, but are very costly to build and require a large scale effort to achieve a usable outcome. In AutoMate, the partners will therefore focus on other modelling techniques, which are based on Dynamic Bayesian Networks and Task Analysis.

1.1.1 Probabilistic Driver Models for Intention Recognition and Behavior Prediction

In AutoMate, a probabilistic driver model will be developed that enables the TeamMate car to create estimations about the intentions and future behavior of the human driver. The model will be based on previously developed hierarchical and modular probabilistic architectures for probabilistic driver models for behavior generation and prediction and intention recognition: *Bayesian Autonomous Driver Mixture-of-Behaviors* (BAD MoB) models and *Driver Intention Recognition* (DIR) models. In AutoMate, these models will be combined in a unified architecture and integrated with vehicle- and situation models to allow the recognition and prediction of driver states, behavior and intentions.

BAD MoB models (Eilers and Möbus, 2014) attempt to describe the statistical relations between observable control actions of human drivers[2] and the perceptual input available via ambient and foveal vision proposed in the psychological literature, like e.g. bearing and splay angles, and information derived from the optical flow for lateral control (Land and Tatler, 2009; Li and Chen 2010; Li and Cheng, 2011), or time-to-x and tau measures for car-following (Lee, 1976; Van Winsum, 1999). The general driving task has been described as a hierarchical structured task with three levels of skills and control (Michon, 1985): the *strategical* (or planning), *manoeuvring* (or tactical), and *control* (or operational) levels. At the strategic level the general planning of a journey is handled, e.g. the driver chooses the route and evaluates resulting costs and time consumption. At the manoeuvring level, the driver has to identify and select appropriate manoeuvres based on his/her current perception of the traffic situation, e.g. turning at an intersection or initiating a lane change. Lastly - at the control level - the driver has to execute simple (and for experienced drivers mostly

[2] By now steering wheel angles and combined acceleration-braking pedal positions as control actions are considered. For AutoMate, it may be necessary to replace these by more abstract signals, e.g. the yaw rate and longitudinal acceleration.

autonomous) sensor-motor programs or perception-action patterns, which taken together form a manoeuvre or specific behavior on the next level. An example is turning the wheel to remain in the middle of the lane. BAD MoB models cover the manoeuvring and control level of Michon's three-layered architecture (Michon, 1985). They rely on the assumption that complex human driving behavior can be hierarchically decomposed into simpler behaviors and - vice versa - that complex human driving behavior can be generated by a sequence and/or mixture of simpler behaviors.

The basic idea of BAD MoB models is best described by a simple example for lateral control on motorways, where it is assumed that the overall lateral control can be decomposed into a set of four simpler behaviors: lane-following, car-following and lane-changes to the adjacent left resp. right lanes. Let A denote a discrete random variable that represents the lateral control actions of the human driver, i.e. the steering wheel angle, and $\boldsymbol{P} = \{P_1, \dots, P_{n_P}\}$ denote a set of discrete random variables that represent the hypothetical perceptual input required for lateral control. Representing the set of simpler behaviors by a discrete random variable B with $\mathrm{Val}(B) = \{\text{lane following, car following, lane change left, lane change right}\}$, it is assumed that for a given set of perceptual evidence $\boldsymbol{P} = \boldsymbol{p}$, the resulting input-dependent conditional probability distribution (CPD) over steering wheel angles $P(A|\boldsymbol{p})$ can be described in terms of a mixture model:

$$P(A|\boldsymbol{p}) = \sum_{b_i \in \mathrm{Val}(B)} P(B = b_i|\boldsymbol{p})\, P(A|B = b_i, \boldsymbol{p}).$$

Here, each probability $P(B = b_i|\boldsymbol{p})$ can be interpreted as the likelihood to perform a corresponding simpler behavior $B = b_i$ for the given situation as encoded by the perceptual evidence $\boldsymbol{P} = \boldsymbol{p}$, while each CPD $P(A|B = b_i, \boldsymbol{p})$ can be understood as an *expert* that provides the likelihood of different control actions in order to realize the corresponding behavior. As such, a BAD MoB model combines aspects of both the manoeuvring level (manoeuvre, resp. behavior selection) and control level (realization of manoeuvres resp. behaviors in terms of motor programs) of Michon's hierarchy (Michon, 1985) in a unified probabilistic architecture.

For a more formal description, BAD MoB models can be seen as variants of *Hierarchical Markov Decision Trees* (Jordan et al., 1997), an extension of Hierarchical Mixture-of-Experts (Jordan and Jacobs, 1994) for modelling temporal processes, over both discrete and continuous random variables. Let $\boldsymbol{A} = \{A_1, \dots, A_{n_A}\}$ denote a set of continuous and/or discrete random variables that represent the control actions of the human driver, $\boldsymbol{B} = \{B_1, \dots, B_{n_B}\}$

denotes a set of discrete variables that represent different behaviors and intentions on the manoeuvring 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 Δ denote a delay between perception and action due to perception and reaction times, usually in the range of 0.5 to 1.0s. A BAD MoB model is then realized as a (conditional) Dynamic Bayesian Network (DBN), defined as a pair $\langle \mathcal{B}_1, \mathcal{B}_{\rightarrow} \rangle$, where $\mathcal{B}_1$ is a (conditional) Bayesian Network that represents an initial conditional probability density $p(\boldsymbol{A}^1, \boldsymbol{B}^1 | \boldsymbol{p}^{1-\Delta})$, and under the assumption of first order Markov and time invariance, $\mathcal{B}_{\rightarrow}$ is a 2-time-slice Bayesian network that represents a transition model $p(\boldsymbol{A}^t, \boldsymbol{B}^t | \boldsymbol{A}^{t-1}, \boldsymbol{B}^{t-1}, \boldsymbol{p}^{t-\Delta})$. For any desired time span $T \geq 1$, the conditional joint density distribution over $p(\boldsymbol{A}^{1:T}, \boldsymbol{B}^{1:T} | \boldsymbol{p}^{1-\Delta:T-\Delta})$ is defined as an unrolled (conditional) Bayesian Network, where, for any $X_i \in \boldsymbol{A} \cup \boldsymbol{B}$, the structure and distributions of X_i^1 are the same as those for X_i^1 in $\mathcal{B}_1$, and the structure and distributions of X_i^t for $t > 1$ are the same as those for X_i^t in $\mathcal{B}_{\rightarrow}$:

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

As illustrated in Figure 1, each CPD in $\mathcal{B}_1$ and $\mathcal{B}_{\rightarrow}$ of the BAD MoB model is associated with a tree-like structure that defines the hierarchical decomposition of the CPD into a set of experts. Each CPD in the context tree represents an expert for the associated region of the input space. Lastly, each CPD in the tree-structure is realized as an internal model that approximates the CPD by a (conditional) Bayesian network, called a component-model. The parameters and structure of component-models can be learned from multivariate time-series of behavior traces via machine-learning methods. By now, the learning algorithms require complete datasets, i.e. datasets without missing values, which makes it necessary to annotate datasets with the (usually hidden) labels of the underlying behaviors.

In the recently finished project "Holistic Human Factors and System Design of Adaptive Cooperative Human-Machine Systems" (HoliDes[3]), BAD MoB models have been used as a starting point for the development of Driver Intention Recognition (DIR) models. DIR models rely on the assumption that the intentions and manoeuvres of the driver can be seen as a hidden process that "emits" observable effects on the traffic situations.

[3] http://www.holides.eu/

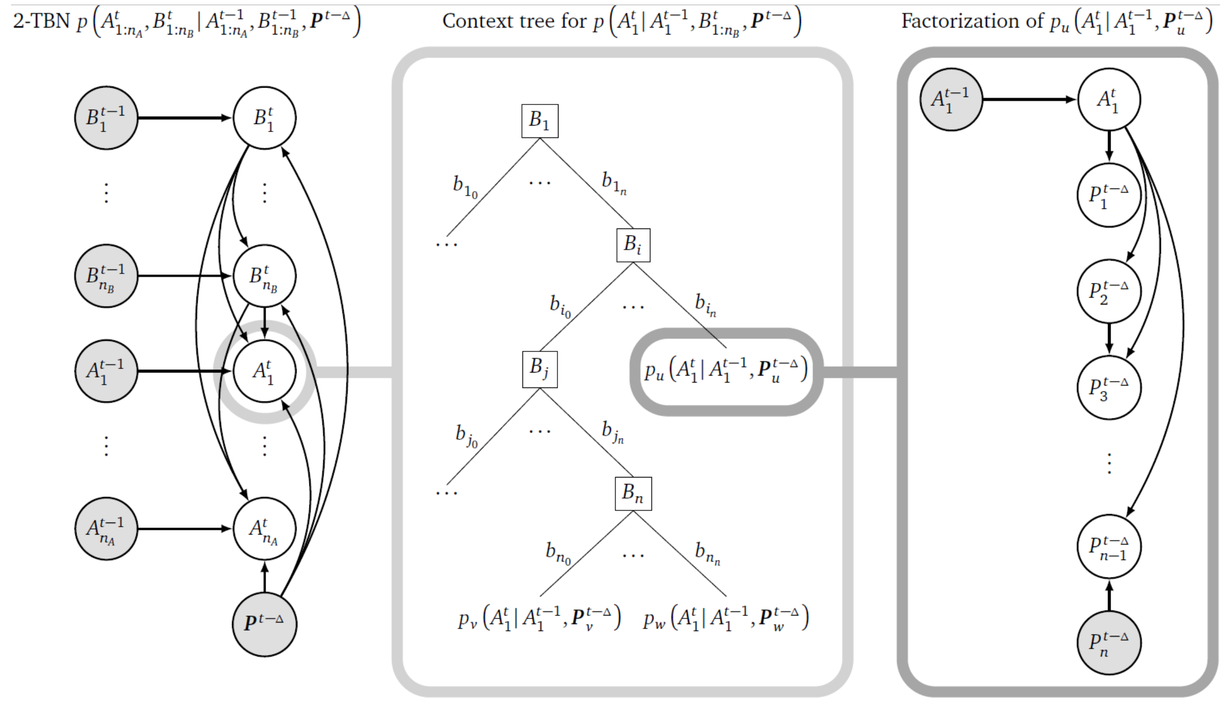

Figure 1: Exemplary overview of the hierarchical and modular architecture for Bayesian Autonomous Driver Mixture-of-Behaviors (BAD MoB) models. Dark nodes indicate that the model does not provide any distributions for the associated variable.

Examples for such observable effects are control actions of the driver, the position of his/her vehicle, physical relations to surrounding traffic participants, etc. 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 $\boldsymbol{O} = \{O_1, \dots, O_{n_O}\}$ 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, behaviors and observations over an arbitrary length $T \geq 1$ as:

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

In AutoMate, the aim is combining BAD MoB and DIR models in a coherent probabilistic architecture for probabilistic driver models that can be used for intention recognition and behavior prediction. Such models will be constructed for specific use cases and be used to recognize and predict the intentions and behaviors of drivers in these use cases. For this, a mechanism to predict the future evolution of the traffic scene is required. Within AutoMate this will be provided by the vehicle- and situation models. If such a situation representation is detailed enough to derive the potential perceptual input available to vehicles in the vicinity of the TeamMate car, it should be possible to use BAD MoB models representing yet to be defined groups of drivers (e.g. trucks, sporty drivers) to predict the future driving behavior of the own and surrounding vehicles that are controlled by human drivers.

1.1.2 Driver State Model

In AutoMate a driver state model shall be developed which can be further integrated within other driver models (e.g. model in previous section 1.1.1). The driver state model aims to provide an indication about the physiological, behavioral and psychological state of the driver. Such information will be used as input by the overall driver model. More specifically, the AutoMate concept of the driver state model will focus on providing the indication about the attention level, the visual distraction state and the fatigue level of the driver.

In order to estimate these indicators the concept will combine facial data, driving data and environmental data. Facial data are provided by a vision based application which processes a video stream provided by a camera installed behind the steering wheel looking at the driver's face through the steering wheel. The application detects and tracks a set of facial features (eye corners, pupils, eyelids, eyebrow, etc.) in real time (see Figure 2). From these features the application measures the eye gaze, the head gaze and the eye opening. The application output makes use of these measures to determine which area the driver is looking at: the road ahead, not to the road ahead, side mirrors, rear view mirror, etc.

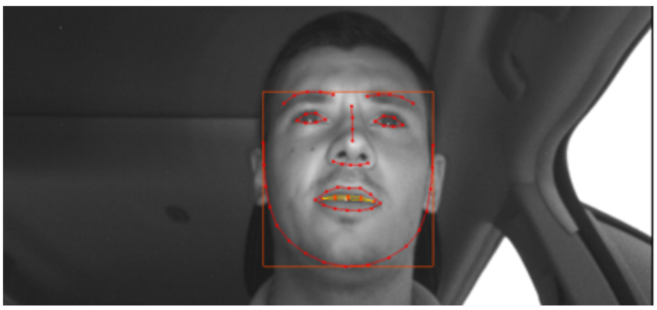

Figure 2: Tracking of facial features in real-time

From the area observed by the driver, the application classifies the visual distraction states of the driver. The visual distraction output is composed of one of two classes: "not visually distracted" (also called "on road") if the driver's visual attention is focused on the road ahead and "visually distracted" (also called "off road") if the driver is not looking ahead at the road (see Figure 3). In order to reduce the false alarm rate the application decides after 120 ms that a driver not looking ahead is visually distracted.

Figure 3: Classification of driver distraction states by the application based on the observed area

The underlying idea is that a driver who is visually distracted cannot be fully aware of the situation. But the contrary case does not need to be true: a driver who is looking ahead is not necessarily aware of the situation, he could be drowsy or cognitively distracted.

Table 1: Classification scheme for visual distraction based on visual attention distribution (VTS = Visual Time Sharing)

Levels of Visual attention	**Rule of classification**
High	VTS > 66%; The driver looks ahead more than 4 s during the last 6 s.
Medium	33% < VTS ≤ 66%
Low	8% < VTS ≤ 33%
Critical	VTS ≤ 8%

The application additionally computes the visual attention distribution (Visual Time Sharing, VTS) concerning the different areas of interest located inside and outside of the vehicle. The application computes an attention level based on the similarity between the expected VTS according to the driving situation and the computed one, which is the actual existing attention distribution (Boverie & Cour, 2011). The visual attention is computed on a time window of 6 seconds and classified into 4 levels according to VTS thresholds (see Table 1).

Furthermore, four drowsiness levels are classified by the application: Alert, Slightly drowsy, Drowsy and Sleepy (see Table 2). The classification is based on eye opening patterns and on the number of blinks and durations of the detected blinks. The drowsiness level diagnostic is based on the fact that people who are getting drowsy will show a modification of their blinking behavior and eyelid opening. For most of them the blink duration will increase along with an increase in sleepiness level (Boverie & Giralt, 2008). The output drowsiness levels are correlated with the Karolinska Sleepiness Scale (see Table 2).

Table 2: Classification scheme of drowsiness levels (related to the subjective rating in the Karolinska Sleepiness Scale, KSS; Akerstedt & Gillberg, 1990)

Levels	**Rule of detection**	**Description** (rating in KSS)
Alert	The driver has very few long blinks and very few very long blinks.	Driver is alert; No sign of drowsiness (KSS: 1 to 5).
Slightly drowsy	The driver has few long blinks and very few very long blinks.	First signs of drowsiness; Driver should only be informed (KSS: 6 to 7).
Drowsy	The driver could have some long blinks and few very long blinks or simply some very long blinks.	Driver is drowsy; Fighting sleep; Degradation of his/her driving performances; Driver must stop and take a rest (KSS: 8).
Sleepy	The driver has some sleepy blinks.	Driver is almost falling asleep; Critical state; Driver must stop urgently (KSS: 9).

Summarizing, the set of data output by the vision based application provides indications to the driver's model about the level of situational awareness of the driver.

In order to make the classification of the driver's state more robust, the inputs coming from the internal camera can be combined with data from vehicle dynamics (speed, yaw-rate, steering angle, etc.) and from the environment (position in the lane, presence and features of surrounding objects, etc.) for the creation of a classifier using a machine-learning (ML) approach (e.g. a Deep Neural Network, such as Co-evolutionary Neural Network, or CNN in short). In fact, ML is the technique of searching large volumes of data for unknown patterns. It has been successfully applied in business, health care and other domains (Baldi & Brunak, 2001; Tan, Steinbach & Kumar 2005). The ML techniques combined with data mining can be able to provide the right algorithms to cope with such a challenge such as driver state classification.

In particular, examples of possible inputs for the classifier are the following:

- Speed [m/s]
- Time To Collision [s]
- Time To Lane Crossing [s]
- Steering Angle [deg]
- Lateral Position [m]
- Lane Width [m]
- Road Curvature [%]
- Heading Angle [deg]
- Position of the accelerator pedal [%]
- Position of the brake pedal [%]
- Turn indicator [on/off]
- X,Y coordinates of car in front (if any)
- Speed of car in front (if any)

1.1.3 Task Model for Driving

In AutoMate a task based approach to driver modelling will be used, focusing on constraints of possible driver actions. The existing task analysis approach CPM-GOMS (Cognitive, Procedural, Motor - Goals, Operators, Methods and Selection Rules, e.g. John & Kieras, 1996) will be adapted and extended to achieve this goal. CPM-GOMS describes human machine-interaction as a set of operators applied by the user to achieve specific goals. A transfer to the domain of automotive driving requires an entirely new set of operators, which have not been described in the academic literature before.

The approach is well suited to describe existing data, such as recordings from vehicles, video data or eye-tracking data. Furthermore it is also possible to model situations that have not occurred yet. Modelling existing data abstracts from data of a high granularity towards more high-level representations of driver behavior. Specifically, the data on a very low level such as driving data, eye tracking or other physiological data can be summarized from the numerical representation onto a semantically richer level. This in turn makes it easier for analysts to interpret both driving behavior of specific persons as well as understand the requirements a situation imposes on executing the driving task. This also helps designing automation technology as it becomes clearer what information must be available, and which kind of computations need to be carried out to solve the driving task in complex environments.

The aim of this kind of driver modelling is twofold. Firstly, to understand the time course of drivers' actions, and secondly, to gain insight into the requirements the driving tasks put forward in order to be carried out successfully. To investigate drivers' actions, their behavior can be recorded in a driving simulator or in an instrumented vehicle. Drivers will move their arms, their heads, they will turn the steering wheel or press pedals. If that behavior is recorded with a high frequency, low-level data (i.e. data on a physical level) with a high granularity (i.e. a high time resolution) can be obtained. But this data is not informative yet, as it is just a large block of numbers. What is needed is a meaningful description of the behavior, such as "the driver looked for an open gap to switch to the left lane" or "the driver was confused about the traffic situation". This gap is addressed by combining the low-level data into operators, applied in specific situations in order to achieve a specific goal. This also tells us a lot about the requirements of the specific situation. Driving on a highway with a low traffic density typically does have few requirements on information uptake and the decision processes on a tactical level are of low complexity. Negotiating a complex intersection in an unknown city as well as searching for the intended route, on the other hand, is much more taxing on the driver's cognition and on a technical system attempting to carry out the driving task as well. Curiously, little is known about those requirements, both regarding the information requirements and the necessary cognition involved (Kircher & Ahlstrom, 2016). Using task analysis and empirical data it might be possible to define those requirements for prototypical situations and manoeuvres.

Finally, it is of great value to be able to construct driver models not only for a specific human-machine interaction based on empirical data, but also to use it for modelling new and unseen situations. To this end, operators are created, which serve to transfer a current state into a goal state. Such a goal state could be to change lanes, for which a sub goal may be to activate the indicator. Models will be constructed for specific use cases. They will be used to predict possible courses of actions in a given situation as well as execution times for driver actions. The models can therefore be validated by comparing model predictions with empirical data.

1.2 Situation and Vehicle Model

In this project the situation model consists of scene objects surrounding the ego-vehicle. These objects can be traffic participants (e.g. other vehicles, pedestrian, cyclist) as well as traffic light signal, road-lanes and other obstacles (tree, pole, building, etc.). Objects will be modelled as 3D bounding boxes with semantic classes (e.g. vehicle, pedestrian, traffic light signal), position, orientation, motion and intention. This information will be estimated using ego-vehicle sensors (e.g. radar, laser scanner, camera, digital map) as input for a sensor fusion system. Uncertainties will be modelled using a probabilistic approach. Detected objects will be tracked over the time.

In order to drive a car autonomously from point A to point B, a module is required that plans concrete actions. The output of this module is a trajectory, containing the vehicle's states parameterized by time. A state contains the position of the vehicle's reference point, orientation, velocity, acceleration etc. As soon as a suitable trajectory is found, it can be sent to the controlling unit for execution. Since this trajectory will be tracked by the vehicle, safety and comfort, as well as efficiency must be guaranteed.

Trajectory planning algorithms require a situation model - including relevant static and dynamic environmental data - as input. For example a path (sequence of positions included in a trajectory) which gets too close to one of the lane's boundary lines, violates the constraints and a new path must be found. Since the trajectory is parameterized by time, future actions of dynamic obstacles (vehicles, pedestrians etc.) must be predicted. Therefore the situation model must provide information concerning the states of dynamic obstacles. These obstacles have to be accurately classified, to know which models are appropriate for prediction. As a matter of fact - and due to sensor uncertainties and other effects - it is impossible to provide a situation model, which exactly matches reality. Therefore, probabilistic information concerning uncertainties, which will be considered during trajectory planning, is incorporated. Another important aspect is that the choice of the used trajectory planning algorithm heavily depends on the provided situation model. In AutoMate, safety corridors will be provided. These corridors contain spatial, as well as temporal data about the environment. This is an approved, well working method (Ziegler et al., 2014).

The vehicle model will be described using the position, the orientation and the motion (velocity and acceleration) of the vehicle with respect to a fixed global coordinate system. For the ego-vehicle this information can be estimated combining environment perception data (optical flow, depth map) and vehicle internal motion information (e.g. angular rate, steering angle, yaw rate, brake pressure, longitudinal/lateral acceleration, GPS data). The estimated vehicle model will be tracked over the time using specific algorithms such as Kalman filters.

2 Process of Model Validation and Metrics Specification

In AutoMate, several models are developed, which have been described in section 1 of this document. The descriptions are necessary for the definition of metrics for the validation of the models. At this point in time it is not possible to specify all details of the metrics for the model validation as they will mature during the project duration. Therefore, in AutoMate a four-step process of the model validation is used:

1. Step: Describe what the goal of the model is (see section 1).
2. Step: Define general metrics, which are related to the goals of the models (see section 3).
3. Step: Description of the purpose of the experiments with regards to the model validation.
4. Step: Specify criteria for deciding whether the validation was successful or not.

Currently, steps 1 and 2 are covered within this document laying the ground work for steps 3 and 4, which will be targeted during the next project cycle. This means that the content of this deliverable will be updated and specified in more detail within the following versions of this deliverable.

3 Preliminary General Metrics

In this section general metrics for the validation of the driver, situation- & vehicle model are described which will be further detailed at a later point in the project.

3.1 Metrics for Validation of Driver Models

In general, every model which is developed in AutoMate shall be validated with empirical or generated data which are compared to the data predicted by the model. Based on the goals of the model and the data which are used, the validation metrics can be defined.

In AutoMate a probabilistic driver model will be developed that can be used for intention recognition and behavior prediction in specific use cases. Validation of this model will rely on comparisons of model predictions and estimations with empirical data in form of multivariate time series of traffic situations for each use case treated as ground truth, i.e. as the objectively correct state of the world that should be estimated. For covert aspects of driving behavior that cannot be observed and therefore won't be included in the empirical data (e.g. intentions) the ground truth will be created by manual annotation of the empirical data by experts, or by the definition of objective measures, e.g. a lane change intention is defined to be present up to x seconds prior to the actual crossing of the lane. For intention recognition the model will be treated as a discrete classifier to be tested with resp. metrics, e.g. confusion matrices and corresponding metrics (see also section 3.2). For behavior prediction, the model will be used to generate estimates of the future traffic situation in different prediction horizons, which will then be compared with the ground truth, by means of time series comparison and the use of appropriate distance metrics.

The driver state model will provide indications about the attention level, the visual distraction state and the fatigue level of the driver. Driver distraction and inattention are important safety concerns (e.g. Regan, Hallett & Gordon, 2011). Deriving knowledge about the human operator can be very valuable in the system validation phase. While interacting with a prototype or some modules of the adaptive cooperative human-machine system, the operator's degree of visual distraction can be evaluated. The purpose of this system is to classify driver distraction, predicting the visual attention location of the driver (i.e. if the driver is looking at the road or not) providing a signal to the driver model. When the signal is "off-road" it means that the driver is not focusing his attention on the road. Thus, if for example a vehicle brakes in front of the driver, he will probably not react in time and the AutoMate System will have to take over the driving situation to avoid the critical situation.

Metrics should compare an expert reference of the model (ground truth) to the processed output of the model. In the frame of AutoMate, a ground truth for each driver state model must be defined. Sets of expert reference data must be generated to train, check and test the models. The data generation must take into account the required amount of data to be statistically significant but also the ground truth accuracy.

Following the ordinary procedure for supervised machine learning, each data set will be split in three different subsets:

- Training data (around 60% of the whole dataset), which are presented to the network during training and the network is adjusted according to its error.
- Checking data (around 15% of the whole dataset), which are used to measure network generalization and to halt training when generalization stops improving.
- Testing data (around 25% of the whole dataset), which have no effect on training and so provide an independent measure of network performance during and after training. The testing data shall be used for the validation.

It is worth noting here that “Supervised Learning” (SL) is the machine learning task of inferring a function from labeled training data. The training data consist of a set of training examples. In SL, each example is a pair consisting of an input object (typically a vector) and a desired output value.

Relevant metrics for the validation test of the driver state model are:

- Mean Squared Error (MSE), which is a measure of the difference between the estimator and what is estimated (namely, it measures the average of the squares of the errors or deviations).
- Correct Rate (CR), which is the percentage of the instances correctly classified by the system.
- Sensitivity, which is the correctly classified positive samples (or True Positive Samples).
- Specificity, which is the correctly classified negative samples (or True Negative Samples),
- Number of false detections per hour.

Other possible metrics can be considered, such as “Percent Error” (it indicates the fraction of samples which are misclassified: a value of 0 means no misclassifications, 100 indicates maximum misclassifications), “Cross-Entropy” (it can be used to define the loss function in machine learning and optimization: lower values are better, zero means no error), etc. Also specific plots, such as confusion matrix and Receiver Operating Characteristic (ROC) curves will be used to graphically represent the results of the classifiers.

3.2 Metrics for Validation of Situation- & Vehicle models

Since situation and vehicle modelling are estimations mostly based on machine learning approaches, metrics available in machine learning will be used to validate the estimation. The main idea is to measure the difference between the estimation and the ground truth. For that, success criteria are defined based on a specific threshold. If the difference between the estimation and the ground truth is greater than a given threshold, the estimation is accepted. Otherwise it will be rejected. Ground truth data can be collected using high accurate reference systems (e.g. laser scanner, digital map, inertial measurement unit). However, these systems are not always available. In most cases, human experts must generate the ground truth manually.

Considering the estimation of discrete values as an object semantic class or intention, the estimation can be validated using a confusion matrix. The confusion matrix in case of a 2-class estimation, Positive (P) and Negative (N), is a 2X2 matrix with the following outcomes:

1. True positive (TP): the estimation accepts a positive instance.
2. False positive (FP): the estimation accepts a negative instance.
3. False negative (FN): the estimation rejects a positive instance.
4. True negative (TN): the estimation rejects a negative instance.

Based on the outcomes mentioned above, several metrics will be computed:

Accuracy (ACC)	$ACC = \frac{\sum TP + TN}{\sum P + N}$
Precision (PR)	$PR = \frac{\sum TP}{\sum TP + FP}$
Recall (RC)	$RC = \frac{\sum TP}{\sum TP + FN}$

Curves such as ROC and Recall Precision Curve (RPC) will also be generated based on the confusion matrix. Some examples of success criteria based on these metrics and curves can be:

1. The classifier must have accuracy greater than 90% and a precision greater than 95%.
2. The classifier ROC must have an Area Under the Curve (UAC) greater than 80%.

4 Conclusion

Within this document the process of the definition of the validation metrics in AutoMate is described. As stated earlier, the first two steps which are necessary within this process are the description of the to be developed models (i.e. probabilistic driver model, driver state model, task model, situation model, vehicle model) and the definition of preliminary, high-level metrics for the model validation. These two steps are covered within this document and will be further refined and updated during the duration of the project. The latter two steps, i.e. the description of experiments and the specification of success-criteria, will also be covered in later versions of this deliverable when more information on the upcoming experiments and necessary aspects of the model validations is available.

5 References

Akerstedt, T. & Gillberg, M. (1990). Subjective and Objective Sleepiness in the Active Individual. *International Journal of Neuroscience 52*, 29-37

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

Baldi, P. & Brunak, S. (2001). *Bioinformatics: The Machine Learning Approach* (2nd edition). Cambridge: MIT Press.

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.

Eilers, M. & Möbus, C. (2014): Discriminative Learning of Relevant Percepts for a Bayesian Autonomous Driver Model. *Proceedings of the 6th International Conference on Advanced Cognitive Technologies and Applications*, 19-25.

John, B. E. & Kieras, D. E. (1996). The GOMS family of analysis techniques: Comparison and contrast. *ACM Transactions on Computer-Human Interaction, 3*(4), 320–351.

Jordan, M. I. & Jacobs, R. A. (1994): Hierarchical Mixtures of Experts and the EM Algorithm. *Neural Computation, 6*, 181-214.

Jordan, M. I., Ghahramani, Z., & Saul, L. K. (1997): Hidden Markov decision trees. *Advances in neural information processing systems*, 501-507.

Kircher, K., & Ahlstrom, C. (2016). Minimum required attention: a human-centered approach to driver inattention. *Human factors, 1-14.* Article first published online: October 13, 2016, https://doi.org/10.1177/0018720816672756.

Land, M. F. and Tatler, B. (2009). *Looking and Acting: Vision and eye movements in natural behavior.* Oxford University Press.

Lee, D. S. (1976). A theory of visual control based on information about time to collision. *Perception, 5*, 437-459.

Li, L. and Chen, J. (2010). Relative contribution of optic flow, bearing, and splay angle information to lane keeping. *Journal of Vision, 10*, 1-14.

Li, L. and Cheng, J. C. K. (2011). Heading but not path or the tau-equalization strategy is used in the visual control of steering towards a goal. *Journal of Vision*, *11*, 1-12.

Michon, J. A. (1985). A critical view of driver behavior models: what do we know, what should we do? Evans, L. and Schwing, R. C. (Eds.), *Human behavior and traffic safety* (pp. 485-520). New York, NY: Plenum Press.

Regan, M. A., Hallett, C. and Gordon, C. P. (2011). Driver distraction and driver inattention: Definition, relationship and taxonomy. *Accident Analysis & Prevention, 43*(5), 1771–1781.

Tan, P.-N., Steinbach, M., & Kumar, V.(2005). *Introduction to Data Mining.* Boston: Addison Wesley.

Van Winsum, W. (1999). The human element in car following models. *Transportation Research Part F, 2(*4), 207-211.

Ziegler,J., Bender,P., Dang, T. and Stiller, C. (2014).Trajectory planning for Bertha — A local, continuous method. *2014 IEEE Intelligent Vehicles Symposium Proceedings* (pp. 450-457). June 8-11, 2014, Dearborn, MI, USA.