D3.4

Metrics and plan for V&V of the concepts and algorithms in the 2nd cycle

Project Number:	690705
Classification	Public
Deliverable No.:	D3.4
Work Package(s):	WP3
Document Version:	Vs. 1.0 (for submission)
Issue Date:	31.10.2017
Document Timescale:	Project Start Date: September 1, 2016
Start of the Document:	Month 12
Final version due:	Month 14
Deliverable Overview:	**Main document:** D3.4 **Annex I:** <name> <classification> **Annex II:** <name> <classification>
Compiled by:	Fabio Tango (CRF)
Authors:	David Käthner (DLR) Paulin Pekezou (DLR) Mark Eilers (OFF) Mohamed-Cherif Rahal (VED) Sébastien Glaser (VED) Stefan Suck (HMT)
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

RECORD OF REVISION		
Date	Status Description	Author
26.09.2017	Deliverable structure	Fabio Tango (CRF)
29.09.2017	Feedbacks about the structure	All interested partners
13.10.2017	Sections provided	All interested partners
18.10.2017	Harmonization of contributions	Fabio Tango (CRF)
25.10.2017	Comments from internal reviewers	TBD
27.10.2017	Integration of comments	Fabio Tango (CRF) and involved partners
30.10.2017	Document ready for submission	Fabio Tango (CRF)
31.10.2017	Document submitted	Andreas Lüdtke (OFF)

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

Table of Contents

List of figures

List of Tables

1 Introduction

The TeamMate car regards driver and system as members on one team, that understand and support each other in their collective goal of safe and comfortable driving. In order to realize this concept, the vehicle must be able to navigate through traffic on its own and therefore it requires the capacity to judge risks connected to certain manoeuvres, as well as to plan and follow concrete trajectories on the road.

As mentioned in deliverable D3.1, the goal of WP3 is to design and implement functionalities which allow the TeamMate car to show the desired behaviour, with specific focus on the adaptive and safe driving strategies. This will be done for the following aspects:

1. online risk assessment
2. algorithms for trajectory planning and execution
3. algorithms to do online and offline learning of the behaviour of a human driver.

After the cycle 1, the approach to verification and validation is refined and adapted now in cycle 2. Hence, the objective of D3.4 is to describe the results of T3.1 at the beginning of 2nd cycle.

For a quick reminder from D3.1, verification and validation should be understood both from a modelling and a software engineering perspective.

Thus, the view of software engineering can be summarized as follows:

- *Verification* is concerned with whether the system under development is well-engineered, error-free, etc.: *Are we building the system right?*
- *Validation* is concerned with whether the system under development will meet the posed requirements: *Are we building the right system?*

Since algorithms can be seen as functions, they take as input certain arguments, such as the situation or a criticality metric; then, functions compute the desired output. In the case of the TeamMate car, this can be a trajectory, a driver intention prediction, or a safety assessment of a certain situation.

When defining the functions, three properties must be addressed: *Verification* (function must always return an output for the given input, even an error), *Efficiency* (the time until an output is produced) and *Validation* (the usefulness of an output provided by the function). For more details, the interested readers can see the deliverable D3.1 and (Oberkampf and Barone, 2006). The degree to which these three properties have been addressed can be expressed by *metrics*. Note that the term *metric* refers to the definition from measurement theory: a numerical representation of an empirical matter that fulfils certain properties.

Finally, this deliverable is structured as following. Section 2 deals with online risk assessment module, while Chapter 3 is about the development of algorithms for trajectory planning and execution. Then, Section 4 illustrates the current situation in the development of the online & offline learning algorithms. Section 5 ends the document with the conclusions and the next steps.

2 Online risk assessment

The purpose of online risk assessment in AutoMate is the calculation of *safety corridors* that quantify the safety of the current and near-future traffic situation according to a metric of risk. These safety corridors will be used by the TeamMate car to assess and plan safe and feasible trajectories, leading to a set of algorithms that allow identifying safe and reasonable arrangements of the driving process.

2.1 Computation of the Safety Corridor

In the following, let Δ denote a temporal step width and h_{max} denote a maximal step width, resulting in a desired prediction horizon $h_{max}\Delta$, and $\boldsymbol{V} = \{v_1, \dots, v_{n_V}\}$ denote a set of $n_{\boldsymbol{V}}$ objects (usually traffic participants) detected by the sensor platform of the TeamMate vehicle at some time step t. As described in D3.3 "Concepts and algorithms incl. V&V results from 1st cycle", the output of the online risk assessment at each time step t is a set $\boldsymbol{c}^{t:t+h_{max}\Delta}$ of *safety corridors* $\boldsymbol{c}^{t:t+h_{max}\Delta} = \left(\boldsymbol{c}^{t:t+\Delta}, \boldsymbol{c}^{t+\Delta:t+2\Delta}, \dots, \boldsymbol{c}^{t+(h_{max}-1)\Delta:t+h_{max}\Delta}\right)$. For the sake of readability and as envisioned for online risk assessment, we will silently assume that $\Delta = 1s$ and omit mentioning Δ in the following.

Each safety corridor $\boldsymbol{c}^{i:i+1}, t \leq i < h_{max}$ defines a region over a temporal interval $[i, i+1]$ for which the probability of collision between the TeamMate vehicle and a *single* object $v \in \boldsymbol{V}$ *or* the road boundaries is upper-bounded by two of user-defined thresholds δ_R and δ_v. Formally, each safety corridor $\boldsymbol{c}^{i:i+1}$ is defined as a set of *polygonal lines* $\boldsymbol{c}^{i:i+1} = \{L_R^{t:t+h_{max}}, L_1^{i:i+1}, \dots, L_{n_V}^{i:i+1}\}$, where a polygonal line L should be understood as a *closed broken line*, i.e. a *polygon*, composed of a finite number of line segments, specified by a sequence of points $L = (A_1, \dots, A_k)$, where each $A_j \in L$ is defined as a pair $A_j = (x_j, y_j)$ denoting the x- and y-coordinates in a Cartesian coordinate system.

For a safety corridor $\boldsymbol{c}^{i:i+1} = \{L_R^{t:t+h_{max}}, L_1^{i:i+1}, \dots, L_{n_{\boldsymbol{V}}}^{i:i+1}\}$, $L_R^{t:t+h_{max}}$ denotes a polygonal line derived from the road boundaries, that *encloses* a region in which the probability of collision with the road boundaries is below the threshold δ_R. Each $L_j^{i:i+1}, j = 1, \dots, n_{\boldsymbol{V}}$ denotes a polygonal line that *excludes* a region for which the probability of collision with a corresponding object is below a threshold δ_V. As such, the joint set of the set of polylines $\{L_R^{t:t+h_{max}}, L_1^{i:i+1}, \dots, L_{n_V}^{i:i+1}\}$ imply a continuous "*safe area*" for the temporal interval $[i, i+1]$, in which the probability of collision with *any* object is upper-bounded by a probability

$$\delta = 1 - (1 - \delta_R)(1 - \delta_v)^{n_V}$$

that can be used by the path planning algorithm to plan current and future trajectories.

A visual example of a safety corridor is provided in Figure 1. We note that a safety corridor abstracts from the dimension of the TeamMate vehicle itself, which should instead be taken into account by the path planning algorithms.

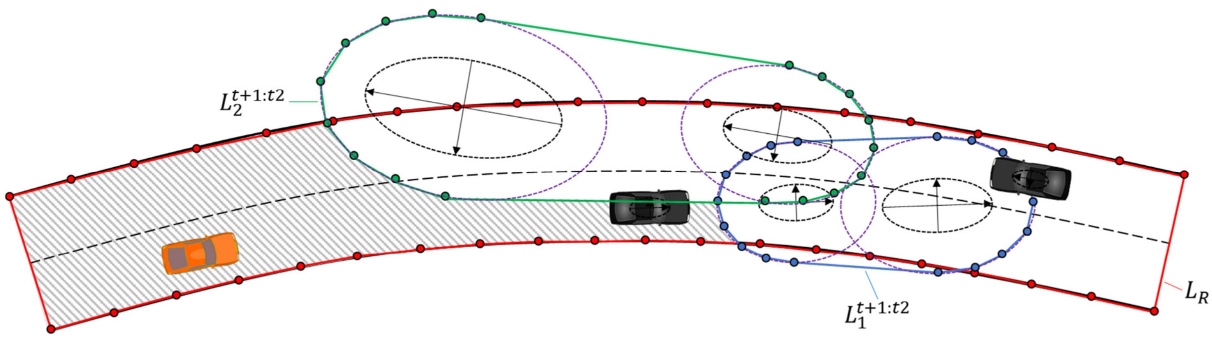

Figure 1: Exemplary visualization of a safety corridor for a temporal interval $[\boldsymbol{t+1}, \boldsymbol{t+2}]$, composed of a polyline $\boldsymbol{L_R^{t:t+n}}$ associated with the lane boundaries and two polylines $\boldsymbol{L_1^{t+1:t+2}}$ (blue) and $\boldsymbol{L_2^{t+1:t+2}}$ (green) associated with two traffic participants. The grey hachured area represents the area of collision-free travel.

To derive the safety corridors, the online risk assessment relies on a prediction of the temporal and spatial evolution of the traffic situation, provided by the Vehicle and Situation Modelling Module (for a description, please refer to D2.3 "Metrics and Experiments for V&V of the driver, vehicle and situation models in the 2nd cycle". For this, we assume that at each point in time t, the sensor platform provides a belief state $p(\boldsymbol{X}_v^t|\boldsymbol{o}^{1:t})$ for each $v \in \boldsymbol{V}$, estimated from the sensor observations received up to the current point in time $\boldsymbol{o}^{1:t}$, where $\boldsymbol{X}_v^t = \{X_v^t, Y_v^t, \Theta_v^t, V_v^t, A_v^t, W_v^t, S_{L_v}^t, S_{W_v}^t, E_v^t, C_v^t, L_v^t\}$, as described in Table 1.

Table 1: Description of variables for the representation of an object $v \in \boldsymbol{V}$ in the vicinity of the TeamMate vehicle considered for the first cycle.

Variable	Type	Unit	Description
X_v	Continuous	[m]	X-coordinate of the center of the object $v \in \boldsymbol{V}$ in a two-dimensional spatial

			coordinate system relative to the position of the TeamMate vehicle
Y_v	Continuous	[m]	Y-coordinate of the center of the object $v \in \boldsymbol{V}$ in a two-dimensional spatial coordinate system relative to the position of the TeamMate vehicle
Θ_v	Continuous	[rad]	Yaw-angle relative to a reference axis
V_v	Continuous	[m/s]	Longitudinal velocity along the objects heading
A_v	Continuous	[m/s²]	Longitudinal acceleration
W_v	Continuous	[rad/s]	Yaw-rate
S_{L_v}	Continuous	[m]	Length (along the x-axis)
S_{W_v}	Continuous	[m]	Width (along the y-axis)
E_v	Binary	$\{\text{true,false}\}$	Binary flag, whether the object $v \in \boldsymbol{V}$ exists in the current traffic scene.
C_v	Discrete	$\{0, \dots, \lfloor \mathrm{C}_v \rfloor\}$	Classification of the object $v \in \boldsymbol{V}$, e.g. PKW, LKW, VRU, etc.
L_v	Discrete	$\{0, \dots, \lfloor \mathrm{L}_v \rfloor\}$	The lane, the object $v \in \boldsymbol{V}$ is currently located in, e.g. fast or slow lane on a two-

For the actual prediction, let $\boldsymbol{S}_v = \{X_v, Y_v, \Theta_v, V_v, A_v, W_v\}$ denote a six-dimensional state for any $v \in \boldsymbol{V}$. At each point in time t, the Vehicle and Situation model is used to infer a sequence of future states $p(\boldsymbol{S}_v^{t+i} | E_v = true, \boldsymbol{o}^{1:t}), i = 1, \dots, h_{max}$. The online risk assessment then uses these predictions to derive a region that encompasses the probable future location of the object v, in respect to its position, dimension, and orientation, with a probability of $(1 - \delta_v)$, i.e., we aim that the probability that an object $v \in \boldsymbol{V}$ is located outside of the predicted region is upper-bounded by δ_v.

Concerning the validation of online risk assessment, it is most important that the resulting safe area is indeed safe, i.e. that no obstacle is located within or penetrating the safe area. Given an individual failure-rate δ_v, the probability that *all* objects $\boldsymbol{V}$ are located within the predictions is given by $(1 - \delta_v)^{n_V}$. To provide an upper bound δ on the probability that *no* object is located within or penetrating the safe area, we therefore need to choose δ_v such that

$$\delta_v = 1 - \sqrt[n_V]{(1 - \delta)}.$$

2.2 Metrics for Validation

Validation of online risk assessment will be performed on a set of independent test data D_{Test}, representing ground truth time-series of traffic situations. As the online risk assessment operates on the prediction of the temporal and spatial evolution of the traffic scene, provided by the Vehicle and Situation Modelling Module, the quality of online risk assessment is upper-bounded by the quality of the predicted evolution, to be evaluated in WP2. To provide an independent assessment of online risk assessment we will assume that the test data D_{Test} consists of samples for which the prediction of the spatial and temporal evolution of the traffic scene is correct.

Let such a test set D_{Test} be composed by a number of m trials, where each trial j, $j = 1, \dots, m$, is a time-series consisting of a number of n_j data samples $d_j^k = \left(\boldsymbol{x}_{v_1}^k, \dots, \boldsymbol{x}_{v_{n_V}}^k\right), k = 1, \dots, n_j$. For each sample d_j^k, and each object $v \in \boldsymbol{V}$, we will use the Vehicle and Situation Modelling Module to predict a sequence of future states $p\left(\boldsymbol{S}_{j,v}^{k+i\Delta} | E_{j,v}^k = true, \boldsymbol{o}^{1:k}\right), i = 1, \dots, h_{max}$, and derive the region that includes the expected position of the vehicle with a probability of $(1 - \delta_v)$. Based on this prediction, the online risk assessment component will be used to calculate a corresponding set of safety corridors $\boldsymbol{c}^{k:k+h_{max}\Delta} = \left(\boldsymbol{c}^{k:k+\Delta}, \boldsymbol{c}^{k+\Delta:k+2\Delta}, \dots, \boldsymbol{c}^{k+(h_{max}-1)\Delta:k+h_{max}\Delta}\right)$, choosing δ_v such that $\delta_v = 1 - \sqrt[n_V]{(1-\delta)}$. For each safety corridor $\boldsymbol{c}^{i:i+1}, k \le i < h_{max}$, we will then use subsequent samples corresponding to the resp. temporal interval $[i, i+1]$ and check for each such sample, whether *any* object $v \in \boldsymbol{V}$ penetrates the implied safety region defined by the conjunction of the polygons. Denoting such an occurrence as a failure and resp. as a success otherwise, we define the metric of validation for a prediction horizon i and a specific level of δ *as the ratio of successes* $\#_s$ *and the sum of successes* $\#_s$ *and failures* $\#_f$*:*

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

We will perform this validation process for different levels of $\delta = 0.5$, $\delta = 0.25$, $\delta = 0.1$, $\delta = 0.05$, and $\delta = 0.01$.

The metric will be used to assess the fulfillment of requirements R_EN5_model1.1 and R_EN_model1.5 stating that the "online risk assessment must be able to calculate a context-dependent safety corridor based on a set of pre-defined metrics" (R_EN5_model1.1) and that the "online risk assessment must determine the safety level of a planned trajectory based on a set of pre-defined metrics" with a correct rate of classification above 90% to be fulfilled. The new metrics will be used to update these requirements accordingly. As it should be expected that the quality of online risk assessment decreases with an increasing prediction horizon $h_{max}\Delta$, we will report the fulfillment of the requirements up to a highest achieved prediction horizon.

Test data for the evaluation of online risk assessment will be obtained from the experiments conducted in the first cycle for obtaining data for the driver models for intention recognition, experiments conducted in the second cycle for the semantic enrichment sub-module and where available, additional real data sets provided by the demonstrator owners (see D2.3 "Metrics and Experiments for V&V of the driver, vehicle and situation models in the 2nd cycle").

3 Trajectory planning and execution

The purpose of trajectory planning is to calculate reference values for the vehicle controller to execute. The trajectory planning module has to consider all relevant vehicle properties, for example the vehicle's shape, it's maximum physical velocity, acceleration etc. Additionally it must always be guaranteed, that the vehicle avoids collisions with other traffic participants and that the vehicle will not leave the road. The functionality of the trajectory planner is already described in deliverable 3.3. In this deliverable, there will be an explanation of how the trajectory planner used in automate will be verified and validated.

3.1 Verification

As explained in the introduction of this document, algorithms can be seen as functions which take certain arguments and deliver values as the result. In the case of the trajectory planner this will be the trajectory. To make sure, that there are no errors in the trajectory planning software, the amount of possible inputs will be varied. For each input the result will be regarded, if the result is not "satisfying" the software will be improved, respectively scanned for errors.

3.2 Validation

The goal of Automate is to develop the "teammate car". A big amount of work to develop the teammate car, is in programming and development of suitable software modules. In the end of the project the functionality of these modules shall be shown in the project demonstrators and simulators. The functionality of the demonstrators will be shown by means of 3 different scenarios, which all take place in so called “structured environments”. These are environments, where there is a predominant direction (along the lane), with speed limits road boundary lines etc. (parking areas are for example non structured areas). Regarding these aspects, the concept of the trajectory planning module was chosen in such a way, that it can handle the three scenarios in automate. To validate the trajectory planning module, before it is running the demonstrators, a simulation environment is built. On this environment, there will be a roundabout (scenario Eva) scenarios, 2 different lanes to be able of perform a lane change which is necessary for overtaking (scenario Peter). For the scenario Martha, in which a takeover of automation on a motorway shall be performed, the trajectory planner itself has no influence.

4 Online & offline learning algorithms to learn from the driver

The AutoMate system should be able to adapt to the driver's preferences and guarantee a human expert-like and safe driving behaviour. To meet this demand the AutoMate system includes the probabilistic driver model for intention recognition and behavior prediction from WP2, which is carried out as a Dynamic Bayesian Network. This model shall be learned offline from annotated driving data, as well as online from observations during the driving process. Where online learning should be understood as the online recalibration of the parameters of the initially offline learned model.

In the second cycle of AutoMate the Peter scenario, which contains driving and overtaking on rural roads, and the online learning for an initially offline trained intention recognition model from WP2 is focused.

In the Peter scenario the aforementioned model will provide the TeamMate vehicle with an online recognition of the current intentions of the driver. While intention should be understood as maneuvers like "keeping the lane" or "changing the lane".

At the beginning the initially offline learned model will rather be able to recognize the intentions of the average driver from all driver data which was used to train the model offline. The online learning should then over time adapt the model parameters to recognize the intentions of the individual driver more robust.

The validation of the online learning algorithms to learn from the driver is understood as the assessment of how well the recalibrated driver model recognizes the intentions of the current individual driver. Thus, the output of the recalibrated model will be compared with empirical data in form of multivariate time series. In particular, we evaluate the performance of the updated driver model on a test data set D_{Test} and in comparison to the initial model. The data set consists of m trials, where each trial is a sequence of recorded traffic situations consisting of a number of $n_j, j = \{1, \ldots, m\}$ data samples $d_j^k = \left(i_j^k, b_j^k, \boldsymbol{o}_j^k\right), k = \{1, \ldots, n_j\}$. Where $\boldsymbol{o}_j^k$ represents the available sensor input, i_j^k the driving intention, and b_j^k driving behaviour. The assumed correct driving intention and driving behaviour are annotated in advance by experts to gain a ground truth. For each sample d_j^k, the initial and the updated model is used to infer a probability distribution over the intentions $P\left(I_j^k | \boldsymbol{o}_j^{1:k}\right)$ and behaviours $P\left(B_j^k | \boldsymbol{o}_j^{1:k}\right)$ given all available sensory input in the corresponding time-series up to this sample. The output of the model is then defined as the most probable intention

$$i_{j,out}^k = \arg\max_{\mathrm{i}} P\left(I_j^k = i | \boldsymbol{o}_j^{1:k}\right)$$

and behaviour

$$b_{j,out}^k = \arg\max_{\mathrm{b}} P\left(B_j^k = b | \boldsymbol{o}_j^{1:k}\right).$$

Since the Driver Intention Recognition model delivers as intention the desired target lane the ground truth lane i_j^k and the predicted target lane from the model $i_{j,out}^k$ are mapped to actual lane changes. Thus, only if the current lane and the target lane intentions differ a lane change intention is preset. We match the lane change intentions from the model with the lane change intentions from the ground truth to construct a confusion matrix as shown Figure 2. The existence of a lane change intention is counted as a positive and the absence as a negative.

		Ground Truth	
		Positive	Negative
Predicted	Positive	TP	FP
	Negative	FN	TN

Figure 2: Binary confusion matrix to visualize model output vs annotated ground truth

Based on this table we can calculate accuracy metric which is defined as:

$$ACC = \frac{TP + TN}{TP + FP + FN + TN}.$$

Analogously, the approach is also applied to this the overall assessment of the behavior recognition. This is done for the initial model and the updated model. The ACC of both models can then be compared. Since the updated model shall be adapted to an individual driver it is crucial that the testing time-series comes from the same driver as the training data that was used to update the model.

The accuracy can be used to verify that the requirements R_EN4_model1.11 and R_EN4_model1.12 for intention recognition and for behavior recognition are fulfilled.

5 Conclusions, outlook

The goal of deliverable D3.4 is to describe the results of Task3.1 at the beginning of cycle 2. In particular, in this document, concepts for trajectory planning as well as the necessary risk assessment have been illustrated, with the further development after D3.3, where *trajectory planning*, *execution & learning* and *on-line risk assessment* modules were described after cycle 1. This means that here, we have described how the concept have been refined and showed that the programming of these concepts has gone on, with some preliminary results.

In addition, we have described the validation and verification process, with the associated metrics for each of the aforementioned modules.

In the next year, cycle 3, we will show the results for the final concepts and algorithms of these modules (deliverable D3.5) and the related metrics, with all possible changes and refinement (deliverable D3.6).

References

Damerow, F. and Eggert, J. (2014). Predictive risk maps. *Proceeding of the IEEE 17th International Conference on Intelligent Transportation Systems (ITSC 2014)*, 703-710.

Houénou, A., Bonnifait, P., and Cherfaoui, V. (2014). Risk assessment for collision avoidance systems. *Proceedings of the IEEE 17th International Conference on Intelligent Transportation Systems (ITSC 2014)*, 386-391.

Laugier, Ch., Paromtchik, I. E., Perrollaz, M., Yoder, J-D., Tay, Ch., Yong, M., Nègre, A., and Makhnacha, K. (2011). Probabilistic analysis of dynamic scenes and collision risks assessment to improve driving safety. *IEEE Intelligent Transportation Systems Magazine*, 3(4), 4-19

Lawitzky, A., Wollherr, D., and Buss, M. (2012). Manoeuvre-based risk assessment for high-speed automotive scenarios. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1-6.

Lefèvre, S., Vasquez, D., and Laugier, Ch. (2014). A survey on motion prediction and risk assessment for intelligent vehicles. *ROBOMECH Journal*, 1:1, 1-14.

Oberkampf, W. L. and Barone, M. F. (1985). Measures of agreement between computation and experiment: Validation metrics. *Journal of computational physics*, 217, 5-36.

Schreier, M., Willert, V., and Adamy, J. (2013). From Grid Maps to Parametric Free Space Maps – A Highly Compact, Generic Environment for ADAS. *Proceedings of the 2013 IEEE Intelligent Vehicles Symposium (IV)*, 938-944.

Weiherer, T., Bouzouraa, S., and Hofman, U. (2013). An interval based representation of occupancy information for driver assistance systems. *Proceedings of the 16th International IEEE Annual Conference on Intelligent* Transportation Systems (ITSC 2013), 21-27.

Soltani, A. R., Tawfik, H., Goulermas, J. Y., & Fernando, T. (2002). Path planning in construction sites: performance evaluation of the Dijkstra, A∗, and GA search algorithms. Advanced Engineering Informatics, 16(4), 291-303.

Raja, P., & Pugazhenthi, S. (2012). Optimal path planning of mobile robots: A review. International Journal of Physical Sciences, 7(9), 1314-1320.