D3.6

Metrics and plan for V&V of the concepts and algorithms in the 3rd cycle

Project Number:	690705
Classification **Deliverable No.:**	Public D3.6
Work Package(s): **Document Version:** **Issue Date:** **Document Timescale:** Start of the Document: Final version due:	WP3 Vs. 1.0 31.10.2018 Project Start Date: September 1, 2016 Month 25 Month 26
Deliverable Overview:	**Main document:** D3.6
Compiled by:	Stefan Suck, OFF
Authors:	Maximilian Graf, ULM Mark Eilers, HMT Stefan Suck, OFF David Käthner, DLR Gurucharan Srinivas, DLR
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
03.09.2018	Deliverable structure	Stefan Suck
19.10.2018	Integrated content	Maximilian Graf
19.10.2018	Integrated content	Stefan Suck
22.10.2018	Integrated content	Mark Eilers
24.10.2018	Integrated content	David Käthner
24.10.2018	Added executive summary and conclusion	Stefan Suck
29.10.2018	Clarifying content	Mark Eilers
30.10.2018	Clarifying content	Maximilian Graf
31.10.2018	Finalization of the document	Stefan Suck

Table of Contents

List of Abbreviations

DIR .... *Driver Intention Recognition*
GMM .... *Gaussian Mixture Model*
V&V .... *Verification & Validation*
WP .... *Work Package*

List of Figures

List of Tables

1 Executive Summary

This document D3.6 is an update of the previous deliverables D3.1 and D3.4 focussing on the results of Task 3.1, the verification and validation plans and metrics based on the requirements from D1.5 for the evaluation of the WP3 enablers.

Section 3 is about the module for planning and execution of safe manoeuvre. Compared to D3.3 the ability of stopping, overtaking, and reacting to leading vehicles behaviour are new features which will be verified and validated. The verification process stayed the same. Besides the monitoring of the optimization costs while keeping the constraints (used in cycle two), the Karush-Kuhn-Tucker condition, and the measuring of the planning time are considered for the validation process.

Section 4 deals with the learning of intention from driver enabler. The new features cover the updating of Gaussian Mixture Models (GMM) contained in some distributions of the Driver Intention Recognition model from WP2, as well as the usage for further scenarios (Martha and Eva). The verification process stayed, while the validation process from cycle two is modified to also evaluate the updating quality for GMMs without the effect of the automatic sample generation. Additionally the performance is validated in terms of the time that is required to update the model.

Section 5 is about the online risk assessment. New for the third cycle is mainly the adaptation to the motorway scenarios (Martha). Furthermore focussed is the assessment of trajectories provided by the enabler for the planning and execution of safe manoeuvres and the optimization of the computational efficiency of online risk assessment. The verification process mainly refers to checks for correct implementation of necessary modifications for the third cycle. The validation follows the approach presented in D3.4 which was utilized in D3.5 considering a safe area that is not penetrated by any object for a specific prediction.

2 Introduction

The TeamMate car regards driver and system as members of one team, who 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 the previous deliverables of WP3 the goal of this work package 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. planning and execution of safe manoeuvre
2. learning of intention from driver
3. online risk assessment.

For cycle 3 the approach to verification and validation is again slightly refined and adapted during Task T3.1. The results of this task at the beginning of 3rd cycle are described in this deliverable.

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?*

When defining the functions or implementing algorithms, three properties should 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.

This deliverable is structured as following. Section 3 deals with the module for planning and execution of safe manoeuvre, while Chapter 4 is about the development of algorithms for learning of intention from the driver. Then, Section 5 illustrates the current situation in the development of the online risk assessment. Section 6 ends the document with the conclusions.

3 Planning and execution of safe manoeuvre

Trajectory planning is required to provide reference values to the vehicle controller, to guide the vehicle safe through the environment while making sure that vehicle passengers feel comfortable at the same time.

Compared to the concept described in D3.3, there are some changes. To enable the ability of stopping and reacting accordingly to leading vehicles behaviour, model knowledge was inserted. To be able to overtake a leading vehicle a new algorithm was implemented within the planner.

3.1 Verification

To make sure that the trajectory planning software is working well and that the system is built right, several verification procedures need to be done. One major part of the trajectory planner is the solving of the optimization problem that consists of a cost functional and constraints as well. The cost functional itself is composed by multiple single terms. Each of these terms has a necessary influence to the global optimal solution. The desired effect of each single term is known in prior and can be checked by disabling all the other terms (by setting their weights to 0), solve the optimization problem and check if the solution looks like as expected. This is a mighty verification opportunity, since disabling all terms, except for the one to verify, the implementation of the system can be checked step by step. In this way, one can make sure, that each term is implemented correctly. To make sure, that the whole system works correctly, various special cases are regarded. For example, if the vehicle drivers on a straight road at the target speed with no leading vehicle, the trajectory planner should calculate acceleration values equal to 0. The effect of implemented constraints can be checked by provoking constraint violations and check if the solver satisfies the constraints nevertheless. It is to mention that in a real application it is advantageous to define the global optimum in a way, that it is located at a point where the constraints are satisfied anyway and no violation of them are provoked. Furthermore, the software is embedded within a simulation environment where it keeps repeating planning cycles as long as the simulation runs. This provides an insight whether the software is really able to react to all conditions of the simulation environment while running stable over an arbitrary temporal horizon. To guarantee that the software if free of memory leaks special software is used to verify the C++ code.

3.2 Validation

The validation of the trajectory planner is done in 2 ways. On the one hand a simulation environment is built for pretesting if all values such as positions, velocities, accelerations etc. are calculated smooth and within kinematic acceptable constraints. On the other hand, testing will be done within a real vehicle to qualitative assess the driving behaviour. Another validation criteria that will be used for validation is the Karush-Kuhn-Tucker condition well-known from optimization theory. The condition describes if the solution is optimal in the sense that the costs cannot be reduced further without violating the constraints. The KKT criteria used is the one of the optimization software used by the trajectory planner.

An important aspect for applicability is the computation time needed to finish one planning cycle. Run time measurements should result in times less than 500ms, since literature, experiments and experience have shown that this time is just acceptable to react to environment changes fast enough. For real-time-capable planner we strive to reach a planning time around 200ms.

4 Learning of intention from 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, which is developed in WP2. This Driver Intention Recognition model is carried out as a Dynamic Bayesian Network. It shall be learned offline from annotated driving data, as well as online from observations during the driving process. In this case online learning should be understood as the online recalibration of the parameters of the initially offline learned model.

At the beginning the initially offline learned model will rather be able to recognize the intentions of the average driver from all the driver data that was used to train the model offline. The online learning should then adapt the model parameters based on observation made while driving to recognize the intentions of the individual driver more robustly. Just like in the cycles before, in the third cycle of AutoMate the online learning is focused on modifying the parameters of an initially offline trained DIR model from WP2.

4.1 Verification

As mentioned in D3.4 Verification will guarantee that needed methods and interfaces are implemented and working correctly. For the third cycle it is intended to change the *lane change* intention in the Peter scenario related to the DIR to a two stage intention. Additionally, since the DIR shall be adapted to Martha and the Eva scenario some changes to the Online Learning module, especially in terms of event identification are required. The necessary modifications have to be checked for correct implementation and therefore verified. For the third cycle it is also planned to support complex versions of the DIR which contain Gaussian Mixture Model (GMM) distributions. Corresponding update methods for GMMs have to be implemented. This is related to requirement R_EN4_model2.3, stating that "The model must be able to modify/update the parameters of the driver model". The integration into further demonstrators during the third cycle might also require some slight interface changes to satisfy the verification requirement R_EN4_model2.8. Additionally, it will also be checked that the online learning module does not store any personal data in a not anonymized way to comply with requirement R_EN4_model2.7.

4.2 Validation

For the validation of the Learning of intention from the driver during the third cycle basically the same procedure as already describe in the previous deliverables D3.4 "Metrics and plan for V&V of the concepts and algorithms in the 2nd cycle" and D3.5 "Concepts and algorithms incl. V&V results in the 2nd cycle ". Thus, 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. Again the approach is to evaluate the output generated by a recalibrated model with test data sets D_{Test} and compared it with the output of the initial model for the same data set. While it is assumed that the initial model was trained on a data set D_{Init} the recalibration will be done by using a further data set

D_{New}. All data set consists of manually annotated multivariate time series which represent the ground truth. For each utilization of the models on the test set D_{Test} a binary confusion matrix as shown in Table 1 is created.

		Ground Truth	
		Positive	Negative
Predicted	Positive	TP	FP
	Negative	FN	TN

Table 1: Binary confusion matrix to visualize model output vs annotated ground truth

Based on this table we can calculate the following parameters:

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

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

- The *False Positive Rate* (FPR), representing the fraction of correctly recognized non-intentions over the total amount of non-intentions:

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

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

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

- The *accuracy*, representing the fraction of correctly classified samples among all samples:

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

- The *F_1-score*, which is the harmonic mean of precision and recall:

$$F_1\text{-}score = 2\frac{PRE \cdot REC}{PRE + REC}$$

The motioned parameters will be used to compare the classification performance of the initial and the recalibrated model for validation requirement R_EN4_model2.2. Since the recalibrated model shall be adapted to an individual driver it is crucial that D_{Test} and D_{New} origin from one and the same driver, while D_{Init} may contain data of multiple drivers.

The experiment for gathering the first validation data for the Peter scenario was already described in D2.3 "Metrics and Experiments for V&V of the driver, vehicle, and situation models in the 2nd cycle". Basically this data should be sufficient to validate the performance of the updating process to create a recalibrated model. But in order to validate if the update of the DIR works well in the new scenarios for the third cycle it is also considered to use new data sets. The experiments to gather the new data are described in D2.5 "Metrics and Experiments for V & V of the driver, vehicle and situation models in the 3rd cycle".

Since the aforementioned update algorithms for the GMMs are more complex than the ones for categorial and Gaussian distributions from cycle one and two a second part of validation for requirement R_EN4_model2.2 is planned.

The effectiveness of the model update for GMMs depends on the quality of sample generation. In order to evaluate the quality of the update algorithms without the influence of the sample generation the initial model is updated with D_{New} while not ignoring the expert annotations, thus updating the model by using the ground truth. A second model is learned offline by using D_{Init} and D_{New}. The updated model and the new offline learned model are evaluated on D_{Test} and compared in terms of the formerly mentioned parameters.

The runtime performance will also be considered for the third cycle the validation is related to requirement R_EN4_model2.9, stating "The update procedure must be sufficiently fast", The current goal is to keep required time for an update below 500ms.

5 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.

As a reminder, a visual example of a safety corridor is provided in Figure 1. A detailed description of the underlying definitions of safety corridors and algorithms for their computation has already been provided in the previous deliverables D3.3 "Concepts and algorithms incl. V&V results from 1st cycle" and D3.5 "Concepts and algorithms incl. V&V results from 2nd cycle" and any additional repetition will this be omitted.

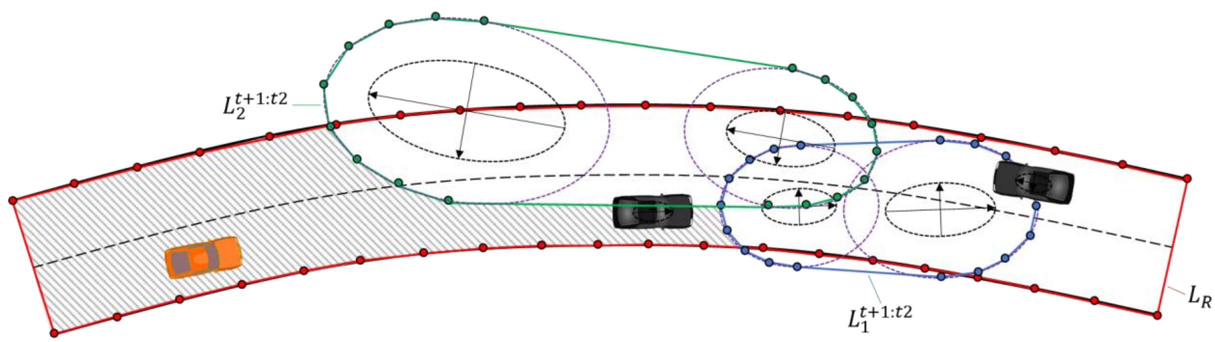

Figure 1: Exemplary visualization of a safety corridor for a temporal interval $[\boldsymbol{t+1}, \boldsymbol{t+2}]$, composed of a polyline $\boldsymbol{L_R^{t:t+n}}$ associated with the lane boundaries and two polylines $\boldsymbol{L_1^{t+1:t+2}}$ (blue) and $\boldsymbol{L_2^{t+1:t+2}}$ (green) associated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

Importantly, the algorithm for constructing the safety corridor can be divided into two independent parts, the construction of the safety corridor obtained from (the prediction of) other traffic participants and the construction of the safety corridor obtained from the lane boundaries. Within AutoMate, the realization of these two parts will be realized by different partners, with individual plans for verification and validation. In the following, Section 5.1 and will introduce the metrics and plan for verification and validation of online risk assessment in the third cycle in respect to other traffic participants (Section 5.1) and the lane boundaries (Section 5.2).

5.1 Safety corridor induced by other traffic participants

For the first and second cycle, we focussed on the development and improvement of algorithms for the computation of safety corridors, primarily in rural road scenarios (Peter). During the third cycle, we will adapt these algorithms to motorway scenarios (Martha) and will furthermore focus on the utilization of safety corridors for the assessment of trajectories provided by enablers for the planning and execution of safe manoeuvres. Finally, due to the intended integration of online risk assessment in the VED real vehicle demonstrator, we will try to optimize the computational efficiency of online risk assessment.

5.1.1 Verification

Verification intends to guarantee that all required methods and interfaces are implemented and working correctly. For the third cycle, it is intended to integrate the online risk assessment in the VED real vehicle demonstrator.

Accordingly, this will require some adaptations to the interfaces and potentially underlying details for computation. During the third cycle, the necessary modifications will be checked for correct implementations and therefore verified.

5.1.2 Validation

Concerning the validation of online risk assessment, it is most important that the safe area encoded by the safety corridor is indeed safe, i.e. that no obstacle is located within or penetrating the safe area, and that the construction of the safety corridor is provided fast enough to be usable in the context of the TeamMate vehicle. In the following, we will define a set of metrics to measure the performance of online risk assessment in respect to these goals. We note that any validation must be performed using a set of independent test data D_{Test}, representing ground truth time-series of traffic situations. Unfortunately, as the ground truth is not available in real world data sets available in the context of AutoMate, we must resort to data obtained from simulator environments.

In the deliverable D3.4 "Metrics and plan for V&V of the concepts and algorithms in the 2nd cycle", we proposed the concept of the "correct classification rate" as the ratio of "successes" and the sum of "successes" and "failures", where we defined a "success" to represent a safe area that was not penetrated by any object (for a specific prediction horizon) and a "failure" to represent a safe area that was penetrated by any object (for a more detailed description and utilization of this metric, we refer to D3.5 "Concepts and algorithms incl. V&V results from 2nd cycle"):

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

with $\#_s$ representing the sum of successes, $\#_f$ representing the sum of failures, i representing a temporal prediction horizon, and $\delta, 0 < \delta < 1$ representing the accepted probability of collision.

The proposed metric has the undesired drawback that the correct classification rate can be arbitrarily increased by (artificially) increasing the area in which another traffic participant is likely to be located for a specific prediction interval, therefore decreasing the size of the overall safe area. To account for this drawback during the third cycle, we will therefore include the overall area defined by the polylines for other traffic as an additional metric, with smaller volumes, which can be interpreted as more certain predictions, being preferred over bigger areas.

Test data for the validation of online risk assessment will be obtained from the experiments conducted for the improvement of the driver models for intention recognition in the third cycle (see D2.5 "Metrics and Experiments for V&V of the driver, vehicle, and situation models in the 3rd cycle"). Unfortunately, due to the test set arising from a simulator study in which the traffic flow is automatically controlled by the simulation software, the resulting behaviour of traffic participants in the vicinity of the TeamMate is highly predictable (c.f., D3.5 "Concepts and algorithms incl. V&V results from 2nd cycle"), and therefore not representing the variability of potential behaviour. To provide a more realistic assessment for humanly controlled traffic participants, we will therefore restrict the validation of online risk assessment to the humanly controlled vehicle.

Finally, we will extend the metrics for validation by performance metrics that assess the efficiency of the online risk assessment in terms of computational complexity. For this, we will rely on analytical and empirical approaches. For the analytical approach, we will analyse and report the relationship between the size of the input data and abstract execution time, by estimating the algorithmic complexity in the asymptotic sense, using the Big O notation (e.g., linear or exponential), when varying the number of objects, the number of points considered for construction of the convex hull, and the prediction horizon. For the empirical approach, we will use the validation data to measure and report the real running time of the online risk assessment on exemplary systems. If possible, we will repeat this analysis for the online risk assessment when integrated in the target demonstrators.

5.2 Safety corridor induced by road boundaries

This enabler is responsible for computing safety corridor between road boundaries by considering the Probability Of Collision (POC) and the current uncertainty of the TeamMate vehicle position. The safety boundary corridor extracted is further used to assess the risk of planned trajectories. Algorithms related to extraction of a safety corridor between road boundaries were presented in D3.3 during the first project cycle, and in D3.5 for the second project cycle.

In the first cycle and second cycle we focused on the development and improvement of the algorithms for safety corridor extraction between road boundaries for complex scenarios. These included junctions, intersections and urban roads. At the end of the second cycle, the algorithm was adapted for the target use case, i.e., the Martha scenarios on a motorway.

Furthermore, we extended the enabler with additional features for assessing the safety of planned trajectories within the extracted safety corridor between road boundaries. In the final cycle of the project, we will further optimize our component for reliability, as well as verify its correctness.

5.2.1 Verification

The online risk assessment will provide a unified interface and thus can be integrated into then TeamMate architecture. This requirement will be verified by checking the enabler involved, conducted by an expert.

The online risk assessment's functionality to assess the safety of a planned trajectory will be evaluated by calculating *Precision* and *Recall* as metrics.

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

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

where TP, FP and FN are true positive, false positive and false negative respectively. For these calculations, synthetic data will be used.

Regarding the storage of personal data, it can be shown that the road boundary based Safety Corridor Extractor does not receive any personal data, and hence cannot store any such data. In respect to performance of the algorithm, we will provide an estimation of the runtime, such as the memory footprint of the application.

5.2.2 Validation

To ensure that the safety corridor extraction between road boundaries does fulfill the required quality, in deliverable D3.5 we suggested to employ the *intersection over union* metric. This metric compares the extracted safety corridor with the ground truth. So far, we have used DLR data for the metric's computation. In the current 3rd cycle, we will use data from the VED for an evaluation that is closer to the target integration environment. Since this metric depends on the required *probability of collision* and the given *uncertainty regarding the TeamMate car's position*, it is currently under discussion what exact thresholds are deemed to be acceptable.

6 Conclusions

The deliverable D3.6 is an update of the previous deliverables D3.1 and D3.4 and describes the results of Task3.1 at the beginning of cycle 3. In particular, in this document, the verification and validation plans, as well as the associated metrics for the *trajectory planning*, for the *learning of intention from the driver*, and for the necessary *online risk assessment* have been illustrated. Considering D3.4, where the metrics and V&V approach of the second cycle was described for the three aforementioned modules, it can be observed that the main parts are still present for the third cycle. Nevertheless the introduction of new features or the adaptation to further scenarios requires for all modules, which are covered by this document, some code upgrades, new data sets, and also some refinements of the of the V&V procedures to gain an insight of the performance of the corresponding module. These refinements of the V&V procedures and metrics were described in this document while the results will be reported in D3.7 "Concepts and algorithms incl. V&V results from 3rd cycle".

References

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