Towards a Scenario-Based Assessment Method for Highly Automated Driving Functions

Korbinian Groh[1], Thomas Kuehbeck[1], Benjamin Fleischmann[1], Mark Schiementz[1] and Claude C. Chibelushi[2]

Abstract— **Current research into highly automated driving (HAD) functions aims to support drivers in various situations. These functions are devised to master different scenarios including traffic participants and environmental conditions. The challenge in this context is to guarantee a fault free operation within this indefinite number of scenarios and estimate the risk of a collision caused by driverless vehicles. Current risk assessment methods are not capable of assessing the performance of the HAD functions within the scenario space. This lack of valid assessment methods motivated the setting up of research partnerships such as the German PEGASUS project in order to measure the improvement of traffic safety. In this research we communicate our method for validating the scenario space using multiple test domains. The first part discusses the requirements towards a scenario description, transferring the test space into a scenario-depending representation which enables the comparison of scenarios across test domains. The second part introduces an evaluation system based on key performance indices for functional, legislative, and system dependent criteria regarding the HAD functions thus determining the performance per scenario. The paper concludes with the proposal of a novel approach towards how highly automated driving functions can benefit online from the evaluation process during drive time.**

I. INTRODUCTION

Rapid improvement in self-driving vehicle development is expected to lead to a ready-to-market technology during the next decade. Remote controlled functions [1] can be seen as the starting point of driver assistance systems which evolved towards the exectution of fixed determined motions. Todays understanding of self-driving vehicles includes the capability to interact with its environment making intelligent decisions [2]. Besides technological feasibility, the highest conceivable form of self-driving is characterized by the ability of navigating completely without human interaction also called autonomous or fully automated driving (FAD) [3]. An intermediate step between FAD and conventional advanced driver assistance systems (ADAS) is given by highly automated driving (HAD) in which the vehicles prediction horizon allows a take-over by the human driver in a reasonable time. Although some highly-developed ADAS seem already closely related to a HAD function, the underlying assessment for their release excludes the driver as last fallback layer. In the case of ADAS, the driver, the vehicle, and the environment are in the loop, continuously interacting with each other. In contrast, HAD systems allow inattentiveness of the driver concerning traffic awareness. This difference illustrates the difference between the today's assessment methods for ADAS compared to HAD systems.

The rest of the paper is structured as follows. It begins by giving an overview of the current assessment method for ADAS and the first approaches for HAD in Section II. In section III criteria for describing a scenario for a HAD function are derived and the paper describes how to evaluate its performance. Afterwards we present an approach for replacing test domains without consistently evaluating scenarios. A summary and outlook conclude the paper in section IV.

II. ASSESSMENT OF ADAS AND HAD

A. *Current Methods for the Assessment of ADAS*

State-of-the-art assessment of ADAS installs a test concept before the actual development process [4]. It consists of a collection of use cases, derived from the requirements that the system needs to fulfil. Each of the use cases is proven via field operational tests (FOT) or naturalistic driving studies (NDS). Both featuring real test domains, which aim to prove the system effectiveness in real traffic. Afterwards, the evaluation either confirms the expected behaviour or reveals functional errors. The results of the use case evaluation are fed back to specify or generate new use cases. The analysis of use case generation is in the first place accomplished by expert knowledge and expanded by experiences gained during the development process using virtual test domains. In contrast to FOT or NDS, virtual test domains are characterized by the deployment of models to simulate functions or components of the vehicle which are not yet available. The degree of virtualization varies depending on the test domain. At the beginning of the development process the software between driver, vehicle and environment is fully simulated by models, called software-in-the-loop (SIL). With increasing progress more realistic test domains can be established by replacing models partially with real components, leading to a decrease in the degree of virtualization. For example, hardware-in-the-loop (HIL) uses real electronic control units or vehicle-in-the-loop (VIL) inserts simulated sensor data into a real vehicle. To date nearly each use case for ADAS is confirmed by FOT or NDS [5]. For a single ADAS this process is economically acceptable as the required test miles are in the one- to two-digit million range [6]. Thus, there exists no need to determine the accuracy of a virtual test domains, their purpose is the generation of new use cases.

---

[1] K. Groh, T. Kuehbeck, B. Fleischmann and M. Schiementz are with BMW Group, Autonomous Driving, D-85748 Munich Garching, Germany, `{korbinian.groh, thomas.kuehbeck, benjamin.fb.fleischmann, mark.schiementz}@bmw.de`

[2] C. C. Chibelushi is with the School of Computing and Digital Technologies, Staffordshire University, Beaconside, Stafford ST18 0DG, U.K. `C.C.Chibelushi@staffs.ac.uk`

With increasing complexity and functional scope of ADAS the number of use cases grew significantly over the last few years and the focus moved towards limiting the number of use cases. First approaches tried to improve the information gain out of FOT and NDS [7]. Under the assumption that good performance in critical situations will be reflected also in uncritical ones, Eckstein et al. [8] extracted and gathered critical situations out of different FOT and NDS. These represent a promising starting point for use cases in virtual test domains. Based on an evaluation metric for scenarios, Tatar et al. [9] vary the input parameters of the applied test domain. The used algorithms perform searches for not yet tested scenarios or worst cases. A similar statistical approach is given by Schuldt et al [4]. The core idea is the reduction of use cases by excluding redundant combinations of parameters. A feasible number of use cases is not the only way to enable a better assessment of ADAS, also the continuous improvement of virtual test domains is an important factor. Besides SIL featuring a heavy parallelization and faster than real time simulations, VIL possess a lot of potential [10]. Especially the ability to simulate consciously critical situations without any risk of human health or material damage represents a unique advantage.

Another important issue besides the assessment of ADAS is their influence on traffic safety. It is determined by post-priori evaluations [11], meaning the comparision of the traffic safety before and after the launch of ADAS. Unfortunately, a post-priori evaluation for traffic safety relies on several critical aspects [12]. First of all, statistical analysis on accident databases cannot be used exclusivley as a large amount of samples and observation time is required [13]. Secondly, the samples have to provide detailed information concerning vehicle types, technical equipment, functional specifications, and driver interaction. In addition, an underlying evaluation metric has to be chosen for estimating the accident severity of each sample [14]. Beyond these problematics, the complete process needs an unacceptable long feedback loop for optimizing traffic safety [15]. Nevertheless, there is up to now no other promising approach to include traffic safety issues already during the development process.

B. *Validity of Future Assessment Methods for HAD*

The major difference between HAD and ADAS rely on an increased amount of traffic scenarios being mastered combined with different environments. In general a real traffic scenario can be described only up to a certain degree of accuracy using parameters. In reality it is simply not possible to observe and gather all details, while in a simulation the accessible parameters are defined by the framework [16]. Taking all parameters which are required for a sufficient representation into account leads to a high-dimensional test space. Its coverage is a sophistacted task, even for simulation. This test space cannot be divided uniformly into grids as parameters are correlated with each other, which causes varying relevant distances between grid points. A sufficient grid-distance for a certain parameter depends on its correlation with the remaining parameters. Especially for continuous parameters the distance may vary in arbitrary small steps. Also a take-over process has to be taken into account for some use cases [17], [18]. A complete list of all use cases to be considered is simply not realizable or results in a not economically affordable task to be performed in a FOT. In addition, the assessment has to be repeated for each relevant derivative of the vehicle [19] or future release of the HAD function.

Of course not every combination of parameters in the test space is of the same interest for the assessment. Situations appearing statistically less often in real traffic are also less likely to be performed during a FOT. Included in this category are particularly accidents or critical situations, whose coverage in the test space are indispensable for the traffic safety. Under the average occurrence of an accident Winner et al. [20] estimates that the required test volume to cover the test space is at least exceeding 10^8 testing kilometers. Regardless of knowing the exact numbers for the test volume, it is out of question that the required test miles exceed by far the capability of FOT and NDS. Moreover, there is no guarantee that all existing types of relevant use cases are discovered. An intelligent choice of the test volume has to be a central topic for the assessment of HAD. A few test miles including not yet considered use cases will give a more valuable feedback for the HAD function than thousands of already well understood ones. Another approach is based on safety by design. Mitsch et al. [21] derived a collision-free algorithm for autonomous robotic ground vehicles, only limited by the simplification of the environment. Thus, the assessment has not only to test the safety of a complex system but already influence the design of the system.

Taking the revealed issues into account and following the argumentation of Wachenfeld et al. [5] and Mazzega et al. [22], a valid assessment method for HAD has to satisfy several aspects. First, the test concept has to be representative in a sense that it can generate and display use cases including street and environment conditions. The degree of simplification in testing a use case must not depart appreciably from its real behavior. Second, the use cases have to be repeatable and therefore the parameters to describe a use case have to be observable. The same requirements are given for the criteria to evaluate a use case. Third, the test concept has to be economically acceptable by means of being feasible. These statements are also aligned with the goals of PEGASUS, a joint project promoted by the Federal Ministry for Economic Affairs and Energy [23]. The two central questions being addressed by PEGASUS are related to the minimum performance that the automated vehicles need to achieve and how this performance can be measured [24].

Although the requirements and the goals for the HAD assessment are clearly determined, the progress towards the realization remains ambiguous. A promising role in this process play the different test domains and their capability to replace FOT or NDS for selected cases. However, for a successful cross-validation of the scenario space by different test domains two decisive questions are still unanswered and we are going to focus our research effort on them in the rest of the paper.

- How can we describe the test space to be representative for all test domains?
- How can we derive a common basis to evaluate test domains?

III. SCENARIO-BASED ASSESSMENT FOR HAD

A. Describing a Scenario for a HAD Function

In the previous section we used terms like *situation*, *scene*, *scenario* and *use case* more or less equivalently. In the following we have to be more precise and distinguish these nomenclatures concerning time span and content. Ulbrich et al. [16] gave a reasonable definition for all four terms. A scene and a situation describe the traffic in form of a snapshot without a time scale. A *scene* contains all neutral information of the traffic or the environment and the self-representation of all participants. The *situation* differs from the scene by including only the information which is relevant for selecting a certain behavior. In contrast, a *scenario* consists of a sequence of scenes representing a temporal dependency. A *use case* wraps a scenario by adding requirements for testing.

Following these definitions, three scenes of a possible scenario in real traffic are pictured in Fig. 1. The scenario itself represents a cut-in (a) of another traffic object (TO), followed by a braking action of the TO in front of the EGO (b) and the decision to overtake the TO (c). If the decision to overtake the TO results in a correct system behavior or not is subject to a subsequent evaluation of the scenario. The scenario description itself includes no rating but simply spans a certain time of traffic involving the EGO vehicle driven by a HAD function. The important question for this section is how can the measured scenario be transferred into another test domain, be re-simulated or be applied to different versions of HAD functions. Thus, the scenario has to be broken down to a description language. We categorized the parameters defining the description language into three parts.

- The environment part contains weather conditions for example position of the sun, date, backlight, humidity or wind strength.
- In the static part we describe the parameters of the road like number of lanes, curve bending or line type. Additionally, non-moving objects, trees or crash barriers are listed.
- The dynamic part stores the features of all traffic participants like the EGO, TOs or pedestrians.

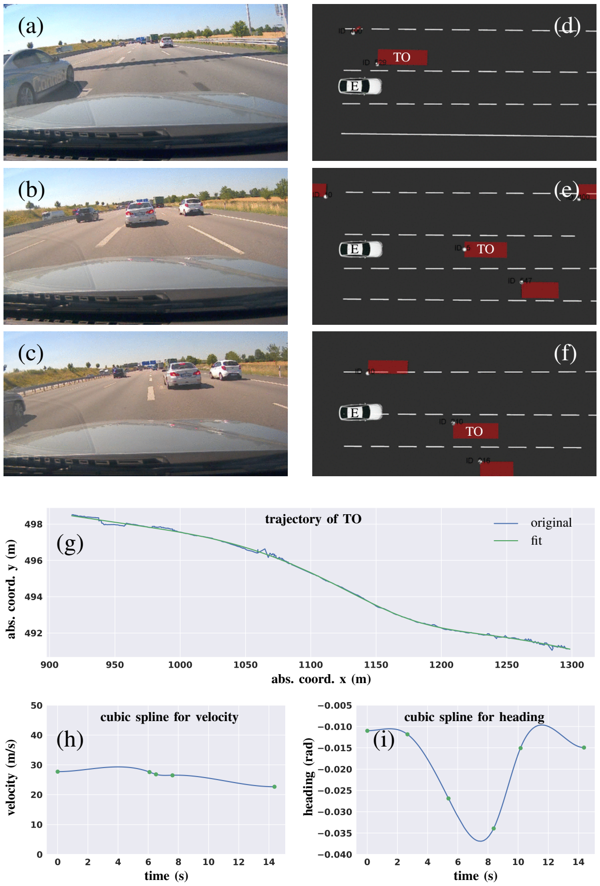

Fig. 1. The figures describe a scenario including a HAD function and its re-simulation in a SIL test domain. The sub-figures (a)-(c) show scenes referring to the scenario recorded in real traffic. It presents a cut-in of a TO visible on the left of the EGO in (a) followed by a braking of the TO in (b). As a result the HAD function decides to overtake the TO in (c). In the sub-figures (d)-(f) the corresponding re-simulated scenes are visualized by the ROS library rviz [25], only based on vehicle and not ground truth data. The trajectory of the TO for re-simulation is defined in the scenario description by a spline interpolation for velocity and heading illustrated in (g)-(i). In (h) and (i) the process for constructing the splines out of original measured data points is pictured. The cubic parameters for each spline section indicated by dots are optimized to yield after integration a trajectory close to the real one shown in (g). As a loss function the mean square error regarding each measured trajectory point was applied.

Concerning the EGO, the description contains only the necessary information to establish an unambiguous initialization at the beginning of the scenario. For all remaining TOs it contains a fixed defined behavior during the scenario. Triggers or events for changing the behavior of TOs are excluded as the resulting traffic of the scenario would be ambiguous for different HAD functions. Another decisive issue is that the scenario description contains no values in form of parameters given to a certain test domain or the HAD function itself. These parameters are of cause influencing the outcome of the scenario and are necessary for re-performing the scenario description in the applied test domain to generate the same outcome, but are not part of the description. The same holds for restrictions or desired goals given to a HAD function. They influence the behavior of a HAD function, thus influencing the outcome of a scenario, but again do not belong to the scenario description itself.

To complete the example from Fig. 1 the results of a SIL test domain for the corresponding scenes are shown in (d)-(f). The re-simulated trajectories of the TOs are obtained on a cubic spline interpolation for the velocity and heading optimized by the mean square error for each time step on the measured trajectory. An example for the optimized splines for the cut-in TO is shown in (g)-(i). The parameters for heading and velocity splines of each TO are afterwards listed in the scenario description relative to the underlying roadmap provided in the static part. This information is sufficient to derive the complete trajectory and higher order derivatives up to second order of the overtaking TO. Also the start position, start heading and start velocity of the EGO are notated in the description file. In contrast, relevant parameters in order to launch the HAD function are part of the simulation and have to be stored outside the description or as additional information in its header. In addition, it is the task of the SIL test domain to achieve the initial conditions for the first scene of the scenario for the ego as well as for all TOs. This includes for our SIL framework a short relaxation time before the start of a scenario in order to load all TO in the simulation and to give the HAD function enough time to recognize them and determine its driving strategy. The sum of all initial conditions for the scenario can be seen as a trigger condition for the test domain to start the scenario. Thus, the test domain is in charge to insure the correct initial conditions and again any parameter required for this relaxation phase can be saved for re-simulation issues in the description header but does not belong to the scenario description.

Historically, there are already approaches to provide similar scenario descriptions for simulation frameworks [26], but mainly the description mixes between the relevant parameters for the scenario and logical statements or simply includes values and goals of the HAD function to the description. An approach for a description which gets close to the requirements is OpenSCENARIO [27]. Up to date it is in implementation and not fully stable yet.

A critical point of a scenario description that is not addressed so far is the required degree of accuracy for representing reality. As mentioned before, reality is characterized by an arbitrary number of details, impossible to be converted into parameters without a loss of accuracy. This problem occurs in each virtual test domain for any component of a HAD function which is simulated. Referring to our example of using a spline interpolation to describe the trajectories of TOs measured in real traffic, the limit of accuracy is given by the mean square error concerning the trajectories in the optimization step. To verify whether this accuracy is sufficient to describe the scenario or not will be solved by design of the evaluation process in the next section. Besides this accuracy issue, we generally suggest to limit entries in the scenario description by the perceiving and interpreting feasibility of the vehicle sensors or the requirements of the driving strategy within a HAD function. If up to date technology is not able to provide certain aspects of a HAD function or a HAD function neglects those aspects in the driving strategy, it is not meaningful to list them in the scenario description. Summarizing this section, the construction of a scenario description according to the given requirements leads to a scenario-based representation of the test space valid for all test domains and a certain degree of accuracy.

B. Evaluating a Scenario

After establishing the description of a scenario and thus the presentation of a scenario-based test space, the question for an evaluation of the HAD function in a chosen scenario arises. At that point it is important to understand that not every test domain can assess the complete HAD function. Fig. 2 gives an overview of a simplified HAD architecture and the components of a HAD function which can be accessed by the most common test domains. The real test domains FOT and NDS of course cover the complete architecture - sensors, environment model, driving strategy and vehicle control. VIL cannot assess the sensors as it relies on already predefined object lists as well as simulated GPS coordinates. Therefore, parts of the environment model like sensor fusion or localization on the roadmap are also not possible to assess whereas the situation interpretation for example already is. In addition, all further components of the driving strategy and the vehicle control are testable without exception. Similar to VIL, SIL is able to access nothing else than the situation interpretation in the environment model and the complete driving strategy. In contrast, the vehicle controls are just partially testable by SIL as the correctness of sending signals can be verified, but not the resulting dynamics. HIL is very valuable as it can in principle test each component or compound of components but it is not feasible to chain too many, in reality.

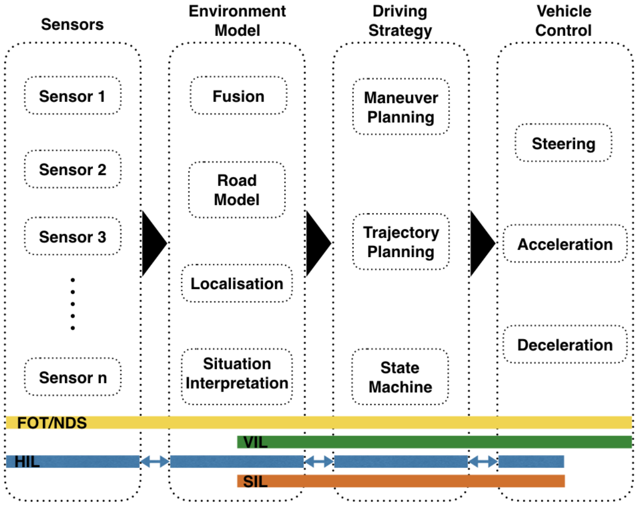

Fig. 2. Simplified architecture of a HAD function. The test domains SIL, HIL, VIL, FOT and NDS are assigned by bars to the fields of components they are able to assess [28].

Following Fig. 2 virtual test domains should be theoretically able to assess certain components of a HAD function. Unfortunately, to close the loop for the assessment process the remaining components which are not accessible in the virtual test domain were replaced by models or approximations. Of cause, there is no guarantee that production ready hardware and software combined with simulated components yield the same results in a single scenario as the complete real loop in FOT or NDS. To make such a comparison we have already established a common basis via the description language but we need also to define scenario-based quantities. In the following we will call this set of quantities calculated from the outcome of a scenario objectifications. In general we distinguish three types of objectification.

- Functional objectifications cover values or technical aspects of the operating HAD function, like sensors availability or GPS signal intensity.
- Quantitative objectifications reflect the behavior of the HAD function, like time to collision (TTC) or specific distances to other objects. Also feelings like convenience are included if an accurate objectification on the given data exists.
- Legal objectifications based on legal requirements, like speed limit restrictions or recognition of traffic signs.

An objectification itself includes no rating but simply contains a value for a quantity. The evaluation of a single scenario itself is established by criteria. A single criterion represents a requirement for the performing HAD function in the scenario and takes one or more objectifications to return a key performance index (KPI). The set of all KPIs in a scenario defines a measure for the risk or in other words the performance of the HAD function in the scenario. Moreover, KPIs make scenarios recorded in different test domains comparable to each other as in sum they express all requirements on a HAD function and thus characterize the scenario sufficiently. In sum, a scenario recorded in FOT or NDS and the same scenario re-simulated by the appropriate scenario description in another test domain will be compared not based on objectifications but on the KPIs of the criteria. The design of a KPI-based comparison also solves the remaining approximation problematic in the scenario description of the last section.

To illustrate this important fact, assuming without loss of generality the scenario and its description from Fig. 1, some objectifications like the TTC values at each time step t and a single criterion $\mathcal{C}$ being the requirement: *The TTC value in respect to TO should not undercut a certain threshold k for any t during the scenario*. In formulas the objectifications in respect to t are given in equation (1) with d for the distance and Δv for the velocity delta between EGO and TO.

$$TTC(t) := \frac{d(t)}{\Delta v(t)} \tag{1}$$

Given the formulation of $\mathcal{C}$, its possible KPIs are binary - $true$ if all TTC values are greater than the threshold or otherwise $false$. A suitable representation of this statement is shown in equation (2), as a sum over a step function Θ.

$$\begin{aligned} \mathcal{C} &:= \begin{cases} true, & \sum_t \Theta(t) = 0 \\ false, & \sum_t \Theta(t) \geq 1 \end{cases} \\ \Theta(t) &:= \Theta(k - TTC(t)) = \begin{cases} 0, & TTC(t) \geq k \\ 1, & TTC(t) < k \end{cases} \end{aligned} \tag{2}$$

The decision whether a certain virtual test domain can replace FOT in the example scenario is made by comparing the KPIs of the criteria for the real scenario and the simulated one. In this small example only the KPI of $\mathcal{C}$ is required. If it is identical the approximation for the scenario description as well as the applied models in the virtual test domain are sufficient to access this single scenario based on the criterion $\mathcal{C}$. It is important to notice that only equality of the KPIs is required, the actual values of the KPIs like $true$ or $false$ reflect the performance of the HAD function. The limitations of a virtual test domain concerning the components of a HAD function in Fig. 2 are still valid. Furthermore, if the KPIs differ, either the models in the virtual test domain or the description language or even both are not precise enough to assess the chosen scenario and a cross-validation is not possible.

C. Cross-validation of Test Domains

We settled in the last two sections our two research questions from the introduction, by defining a scenario description language and combining it with an evaluation which enables the comparison of test domains for a single scenario. Up to this point no advantage regarding the minimization of the scenario space or computation time efficiency is generated, because to yield this replacement statement we had to perform and evaluate the scenario in both test domains. However, the information that a certain scenario can be cross-validated leads to the assumption that this statement is also true for similar scenarios in the test space. The idea for the resulting progress is pictured in Fig. 3. A data lake contains the scenario data measured in FOT or NDS. This data is cut in time series representing scenarios and stored in a database. Afterwards the KPIs for each scenario are derived and linked to the

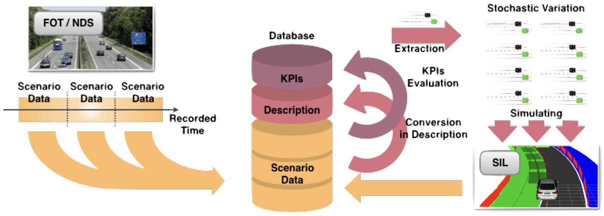

Fig. 3. Architecture for storing scenarios, scenario descriptions and corresponding KPIs. Scenario descriptions can be extracted and re-simulated in another test domain like SIL, representing the possibility of cross-validating. Also the simulation of varied scenario descriptions is possible.

corresponding scenario description. Similarly, each scenario description is extracted, linked to the original FOT data and re-simulated in another virtual test domain, like SIL. The simulated scenarios run through the same process in the database besides that the scenario description generation can be skipped as it will be identical to one used before. If a scenario in FOT and a virtual test domain can be cross-validated because of sharing the same KPI set, slightly varied scenarios based on the parameters in the scenario description are also simulated. The goal is to find parameters which will scale continuously in terms of the resulting KPIs. For these parameters a simple check with a real test domain can ensure that the direction of variation still yields similar KPIs and thus legitimizes the cross-validation for all scenarios with parameter values in between the original and the varied one. In contrast, if the check fails we cannot trust any of the scenarios including parameters in the variation range. The origin of this behavior lies as already mentioned in the correlation to other parameters. We do not know the exact impact of small variations of parameters on the KPIs. Promising candidates are mainly parameters representing the dynamics of TOs as these are bound to physical limitations and their predictions are already part of the driving strategy. Parameters in the environment or static part of the scenario description tend to show no smooth behavior on the KPIs. One example for this is the variation of a junction angle in arbitrary small steps. Assuming at some point the junction is not visible anymore. This fact will cause a decisive change on the driving strategy of the HAD function and most likely in complete different KPIs as in the last variation step.

IV. SUMMARY AND OUTLOOK

We demonstrated that a scenario description in combination with KPIs enables the cross-validation of a scenario by several test domains. Taking the nature of the description format into account, let us assume that dynamic parameters are sufficiently continuous. Combining these two facts lead to a scenario-based test space for dynamic parameters in which FOT or NDS can be locally exchanged by virtual test domains.

Our recent research focuses on the partial online deployment of the assessment method for improving traffic safety. Theoretically, the assessment method is limited by the computational time for the numbers of variations on the scenario description file, the subsequent simulation for each generated scenario description and its evaluation. Obviously, increasing the number of variations will also increase the quality of prediction. Unfortunately, the last two parts are particularly time and hardware intensive, including multiple simulation runs and evaluations on a huge amount of data. For that reason we follow the deep learning based approach of training neural networks on predicting the performance assessment of a scenario. Scenario description and KPIs act as training sample. In contrast to virtual test domains, neural networks process scenario descriptions without any simulation or the generation of data. Based on their computational efficiency, trained networks will not only significantly contribute to the on-board assessment. Also pre-screening for further exploration of the scenario space and detection of hot spots are possible areas of application, enabling a precise deployment of more time critical test domains.

REFERENCES

[1] "'Phantom Auto' will tour city," The Milwaukee Sentinel, Dec. 1926.

[2] "Google's self-driving car project," https://waymo.com, accessed: 2017-03-30.

[3] SAE International, "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," Sept. 2016.

[4] F. Schuldt, F. Saust, B. Lichte, M. Maurer, and S. Scholz, "Effiziente systematische Testgenerierung für Fahrerassistenzsysteme in virtuellen Umgebungen," *AAET2013-Automatisierungssysteme, Assistenzsysteme und eingebettete Systeme für Transportmittel, Braunschweig*, 2013.

[5] W. Wachenfeld and H. Winner, "Die Freigabe des autonomen Fahrens," in *Autonomes Fahren*. Springer, 2015, pp. 439–464.

[6] M. Fach, F. Baumann, J. Breuer, and A. May, "Bewertung der Beherrschbarkeit von Aktiven Sicherheits-und Fahrerassistenzsystemen an den Funktionsgrenzen," *VDI-Berichte*, no. 2104, 2010.

[7] H. Lietz, T. Petzoldt, M. Henning, J. Haupt, G. Wanielik, J. Krems, H. Mosebach, J. Schomerus, M. Baumann, and U. Noyer, "Methodische und technische Aspekte einer Naturalistic Driving Study," *FAT-Schriftenreihe*, no. 229, 2011.

[8] L. Eckstein and A. Zlocki, "Safety potential of ADAS–Combined methods for an effective evaluation," in *23rd International Technical Conference on the Enhanced Safety of Vehicles (ESV) Seoul, South Korea*, 2013.

[9] M. Tatar and J. Mauss, "Systematic test and validation of complex embedded systems," *ERTS-2014, Toulouse*, pp. 05–07, 2014.

[10] R. Pfeffer and T. Leichsenring, *Continuous Development of Highly Automated Driving Functions with Vehicle-in-the-Loop Using the Example of Euro NCAP Scenarios*. Cham: Springer International Publishing, 2016, pp. 33–42. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-32345-9_4

[11] K. Kompass, C. Gruber, C. Domsch, *et al.*, "Der Beitrag von Fahrerassistenzsystemen zur Aktiven und Passiven Sicherheit–die Integrale Sicherheit als Antwort auf die wachsenden Anforderungen an die Fahrzeugsicherheit." *4. Tagung Sicherheit durch Fahrerassistenz*, 2010.

[12] T. Helmer, M. Neubauer, S. Rauscher, C. Gruber, K. Kompass, and R. Kates, "Requirements and Methods to Ensure a Representative Analysis of Active Safety Systems," in *11th International Symposium and Exhibition on Sophisticated Car Occupant Safety Systems*, 2012, pp. 6–1.

[13] F. Fahrenkrog, A. Zlocki, and L. Eckstein, "Bewertung aktiver Sicherheit. Vom Test zur Wirksamkeitsanalyse," *Automobiltechnische Zeitschrift*, vol. 116, no. 1, 2014.

[14] T. M. Gasser, C. Arzt, M. Ayoubi, A. Bartels, L. Bürkle, J. Eier, F. Flemisch, D. Häcker, T. Hesse, W. Huber, C. Lotz, M. Maurer, S. Ruth-Schumacher, J. Schwarz, and W. Vogt, "Rechtsfolgen zunehmender Fahrzeugautomatisierung," *Berichte der Bundesanstalt für Straßenwesen. Unterreihe Fahrzeugtechnik*, no. 83, 2012.

[15] T. Helmer, K. Kompaß, L. Wang, T. Kühbeck, and R. Kates, *Safety Performance Assessment of Assisted and Automated Driving in Traffic: Simulation as Knowledge Synthesis*. Cham: Springer International Publishing, 2017, pp. 473–494. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-31895-0_20

[16] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt, and M. Maurer, "Defining and substantiating the terms scene, situation, and scenario for automated driving," in *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*. IEEE, 2015, pp. 982–988.

[17] C. Gold, D. Dambck, L. Lorenz, and K. Bengler, "Take over! How long does it take to get the driver back into the loop?" *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 57, no. 1, pp. 1938–1942, 2013. [Online]. Available: http://dx.doi.org/10.1177/1541931213571433

[18] J. Radlmayr, C. Gold, L. Lorenz, M. Farid, and K. Bengler, “How Traffic Situations and Non-Driving Related Tasks Affect the Take-Over Quality in Highly Automated Driving,” *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 58, no. 1, pp. 2063–2067, 2014. [Online]. Available: http://dx.doi.org/10.1177/1541931214581434

[19] F. Burgdorf, *Eine kunden-und lebenszyklusorientierte Produktfamilienabsicherung für die Automobilindustrie*. KIT Scientific Publishing, 2010.

[20] H. Winner and W. Wachenfeld, “Absicherung automatischen Fahrens, 6,” *FAS-Tagung München, Munich*, 2013.

[21] S. Mitsch, K. Ghorbal, and A. Platzer, “On provably safe obstacle avoidance for autonomous robotic ground vehicles,” *Proceedings of Robotics: Science and Systems*, 2013.

[22] J. Mazzega, F. Köster, K. Lemmer, and T. Form, “Absicherung hochautomatisierter fahrfunktionen,” *ATZ-Automobiltechnische Zeitschrift*, vol. 118, no. 10, pp. 48–53, 2016.

[23] F. M. for Economic Affairs and E. (BMWi), “Pegasus research project,” http://www.pegasus-projekt.info/en/, 2016, accessed: 2017-03-30.

[24] F. Köster, T. Form, K. Lemmer, and J. Plättner, “Wie gut müssen - automatisierte Fahrzeuge fahren - PEGASUS,” in *AAET 2016 - Automatisierungssysteme, Assistenssysteme und eingebettete Systeme für Transportmittel*, H. I. automotive nord, Ed., Februar 2016, pp. 292–300. [Online]. Available: http://elib.dlr.de/103235/

[25] “Robot operating system.” [Online]. Available: http://www.ros.org

[26] J. Kearney, P. Willemsen, S. Donikian, and F. Devillers, “Scenario languages for driving simulation,” *DSC99*, 1999.

[27] VIRES, “OpenSCENARIO,” http://www.openscenario.org, accessed: 2017-03-30.

[28] M. Aeberhard, T. Kühbeck, B. Seidl, M. Friedl, J. Thomas, and O. Scheickl, “Automated driving with ros at bmw,” *ROSCon 2015 Hamburg, Germany*, 2015.