Towards Runtime Detection of Novel Traffic Situations

Ishan Saxena[1,*], Dominik Grundt[1], Eike Möhlmann[1] and Bernd Westphal[1]

[1]*German Aerospace Center (DLR) e.V., Institute of Systems Engineering for Future Mobility, Oldenburg, Germany*

Abstract

Automated Vehicles developers need to define an Operational Design Domain (ODD) where such vehicles can operate safely. In order to extend the defined ODDs, the developers base their decision after detailed analysis of recorded data from multiple data collection drives. For the acquired data, it is important to know whether it is known traffic situation information (inside the automated vehicle's ODD) or novel information that can be used to expand the ODD. The large amount of data that is generated by a modern vehicle's sensors makes data storage and efficient analysis for expanding ODDs hardly feasible (most of the current approaches record all sensor data and then post-process the data using AI-based methods and finally perform manual checks in order to find the novel data). Hence, there is a need to classify traffic situations as novel at system runtime for an appropriately abstract notion of novelty so that the conceptually same traffic situation, e.g. on two similar days, is not considered novel only because of the different date.

We propose a new methodology for detection of novel traffic situations at system runtime. The methodology is based on a traffic catalogue that consists of abstract traffic situation descriptions, which are a formalized representation of sets of concrete traffic situations. Continuous, automatic checks for satisfaction of the current traffic situation against the traffic catalogue provides verdicts about the novelty of the current traffic situation. Using an example, we show how domain experts can utilize the detected novelties to create such a traffic catalogue such that the novelties are classified as known in the future. The proposed method doesn't require any pre-training of an AI-based classifier and is human understandable, explainable and traceable.

Keywords

Novel Traffic Situation Detection, Data Recording, Traffic Situations, Formal Specification, Runtime Monitoring

1. Introduction

In the automotive domain, ensuring the safe operation of automated vehicles is a topic of great interest. One of the ways in which Automated Driving Function (ADF) developers can ensure safe operation of automated vehicles involves allowing such vehicles to only drive while inside their Operational Design Domain (ODD). The ODD of an automated vehicle may be defined as the set of operating conditions, such as environmental factors, traffic conditions, etc., for which the vehicle has been designed and tested rigorously and has proven capable of safe operation [1, 2, 3, 4].

ADF developers continuously analyse the ODD of their automated vehicles to pursue their expansion in order to enable safe operation of such vehicles in additional operating conditions. The usual process for expanding the ODD involves manual identification of operating conditions by experts that are not yet covered by the ODD, carrying out training data collection drives aiming to encounter the desired operating conditions, filtering of the collected data, training the ADFs on the filtered data and finally testing whether the automated vehicle is now capable of safe operation for the new operating conditions [5]. This process of data collection, data filtering, ADF training and testing should be continued even after delivery to further improve the safe operation of automated vehicles.

This process has certain drawbacks. Data collection is currently carried out by recording all sensor data during the entire drive. A vehicle equipped with RADAR sensors, LiDAR sensors and cameras generates around 5 GB of data per second [6], which requires a large amount of on-device storage.

OVERLAY'25: 7th International Workshop on Artificial Intelligence and fOrmal VERification, Logic, Automata, and sYnthesis, October 26, 2025, Bologna, Italy

[*]Corresponding author.
ishan.saxena@dlr.de (I. Saxena); dominik.grundt@dlr.de (D. Grundt); eike.moehlmann@dlr.de (E. Möhlmann); bernd.westphal@dlr.de (B. Westphal)
0000-0003-0575-4402 (I. Saxena); 0000-0002-8233-7429 (D. Grundt); 0000-0003-3815-6353 (E. Möhlmann); 0000-0002-6824-0567 (B. Westphal)

Offline identification of novel operating conditions (i.e. conditions not yet present inside the ODD) from such a large dataset is a cost-, time- and resource-intensive process. This is because the data is available as unorganized and unlabelled collection of sensor data and recordings. This large amount of data must be post-processed, clustered, and then manually checked by experts to determine presence of novel operating conditions.

In this paper, we propose a methodology for automatic identification of novel operating conditions at system runtime. This allows targeted recording of novel sensor data during data collection. In detail, our approach proposes the use of runtime monitoring to classify the current traffic situation based on a traffic catalogue. The proposed traffic catalogue would consist of abstractly specified traffic situations representing operating conditions that experts have determined to be already present in the ODD. Concretely, we use the Spatial View (SV) formalism of the Traffic Sequence Charts (TSCs) [7] language. SVs allow a visual yet formal specification of abstract traffic situations, especially the spatial relations between various road users. This classification at system runtime would allow recording of only novel data, leading to a saving of storage space, time and compute resources. We have exemplarily applied the approach in a simulation-based environment for the first steps towards a traffic catalogue.

A number of approaches are present in literature dealing with identification of novelties in the automotive domain. For us, approaches that deal with novelties arising in a traffic situation due to interaction of traffic participants with each other are of interest. The approaches introduced in [8] and [9] encode their inputs using autoencoder and CLIP respectively. Finally, the encoded information is clustered based on different parameters to detect novelties. The authors in [10] propose a proprietary novelty metric called Unexpectedness Index that measures how unexpected the driving scenario is from perspective of the system under test. They use this novelty metric for the generation of unknown-unsafe scenarios in simulation as per SOTIF [11]. These methods require embedding of data into an AI-generated latent space for detection of novelties. Hence, they suffer from the known issues occurring due to the black-box behaviour of AIs. Additionally, they are only suitable for offline tasks and cannot directly be used to detect novelties at system runtime.

The research closest to this work is [6], where the authors propose a method for real-time identification and recording of novel dynamic behaviour using single-variate time-series signal classification. The authors propose construction of a Behaviour Forest based on continually arriving data points to discover dynamic behaviour. Once a novel dynamic behaviour is found which has not been encoded in the Behaviour Forest yet, the leaf nodes are extended and the corresponding time-series data is recorded. Our proposed approach allows for detection of novelties from multidimensional sensor data, and allows detection of novelties occurring due to interaction of multiple objects at system runtime.

2. Novel Traffic Situation Detection

In this section, we present our concept for novel traffic situation detection at system runtime. First, we provide definitions for the terminology used in the concept. On the basis of these definitions, we first define a novel traffic situation. Further, we introduce a method for detection of such novel traffic situations at vehicle runtime and recording of novel data. Finally, we present how a traffic catalogue can be created and extended by domain experts on the basis of the recorded novel data. The concept overview is presented in Figure 1.

A traffic situation consists of traffic environment and traffic participants. A *traffic environment* is defined as the context in which traffic participants operate, e.g. roads, lanes, etc. A *traffic participant* is an entity that interacts with traffic environment according to a set of behavioural and physical rules and contains attributes, e.g. vehicles, pedestrians, etc. We use an Object Model OM to model traffic participants and traffic environment. Let $OM = (\mathcal{C}, \mathcal{T}, Pred, Fun)$, where $\mathcal{C}$ is a set of classes, $\mathcal{T}$ is a set of basic types, $Pred$ is a set of typed predicate symbols and Fun is a set of typed function symbols. $attr(\mathcal{C})$ is a finite set of typed attributes for each object class. A *concrete traffic situation* is a spatial arrangement of traffic participants, at a particular point of time, within the traffic environment along with their state. Concrete traffic situations, over the Object Model OM, are defined as a function σ

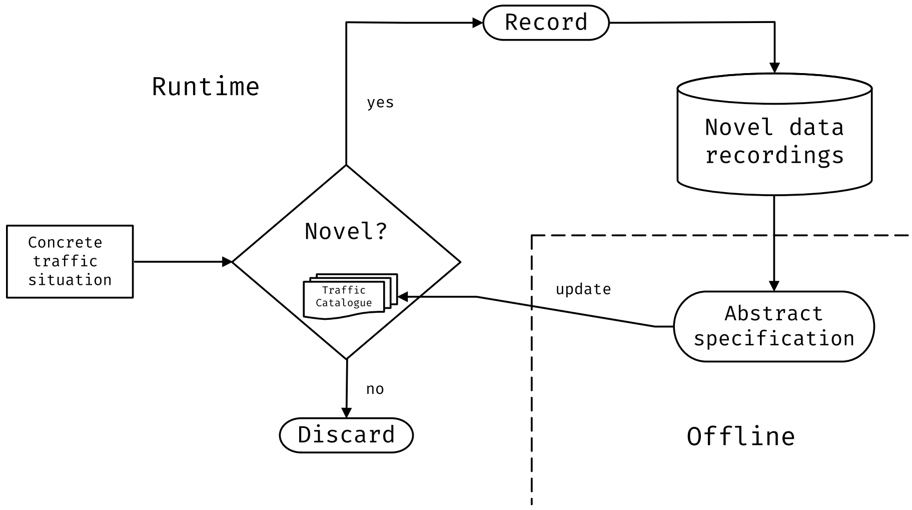


Figure 1: Proposed concept for detection of novel traffic situations

over a finite set of object identities ID (pertaining to the objects present in the situation) along with a type-consistent valuation of their attributes. $\sigma : \mathrm{ID} \nrightarrow (attr(\mathcal{C}) \nrightarrow \mathcal{D})$, where ID is the set of object identities provided to class instances, $\mathcal{D}$ is the domain set of attribute types. Hence, $\sigma(id)(a : t) \in \mathcal{D}(t)$.

Traffic situation specifications formally describe desired state and spatial properties of concrete traffic situations. They are defined as a set of well-typed predicate logic formulae over OM. Examples of traffic situation specification include Spatial Views [7], Abstract Scene Graphs [12], etc. Let ϱ represent the valuation of variables present in predicates derived from the attribute values of a specific concrete traffic situation. We define that this specific concrete traffic situation σ will satisfy a traffic situation specification S, i.e. $\sigma \vDash S \iff \varrho \vDash P(S)$, where $P(S)$ are predicates present in specification S. Let Σ represent a set of all concrete traffic situations that can exist, then the specific concrete traffic situations that are specified by the specification S can be represented as $\Sigma_S \subseteq \Sigma$. It is possible for a concrete traffic situation to satisfy more than one traffic situation specifications. Finally, a *traffic catalogue* is a finite set of traffic situation specifications. For the case of Novelty Detection, we assume that the concrete traffic situations represented by the specifications have already been seen by the domain experts and are no longer of interest for data recording. A traffic catalogue is represented as $TC = (S_1, S_2, \ldots S_m)$, where m is the number of traffic situation specifications present in the catalogue.

We define a *novelty* or a *novel traffic situation* as a concrete traffic situation which satisfies none of the traffic situation specifications present in a traffic catalogue. Formally, if $\forall S \in TC : \sigma_{\mathrm{N}} \nvDash S$, then σ_{N} is a novelty. We define *novel data* as the sensor values contained in the concrete traffic situation which satisfies none of the formalized situations present in a traffic catalogue. This is the data that should be recorded and is of interest to domain experts. For detection of novelties at system runtime, we require a method that can provide us continuous verdicts about whether a concrete traffic situation satisfies any of the traffic situation specifications present in a traffic catalogue. Such traffic situation specifications, occurring in the automotive domain, are usually quite complex. The complexity arises from the need to specify interactions between multiple traffic participants, the traffic environment and their attributes. In the field of Scenario-based Testing [13], Runtime Monitoring (RM) has already been used for continuous checking of complex system requirements for system runs in the automotive domain [14]. These complex system requirements are specified using abstract scenario specifications and concrete system runs are monitored for satisfaction against them. We also require a similar check for traffic situations, and since a scenario consists of finitely-many traffic situations, we conclude that we can use RM similarly for our purpose. RM for a concrete traffic situation is performed using runtime monitors which are software components that continuously take as input (processed) sensor signals and provide an immediate verdict regarding satisfaction with a traffic situation specification. We generalize the runtime monitoring definition for a complete traffic catalogue as follows:

$$\mathrm{Mon}(\sigma, TC) := \begin{cases} \top & \text{if } \exists S \in TC : \sigma \vDash S, \\ \bot & \text{otherwise} \end{cases} \tag{1}$$

When $\mathrm{Mon}(\sigma, TC) = \top$, this implies that the current concrete traffic situation is an already known situation and satisfies at least one of the traffic situation specifications present in the traffic catalogue. When a novelty is encountered i.e. $\mathrm{Mon}(\sigma, TC) = \bot$, then the data recording mechanism is triggered to record the corresponding sensor data. The recording of data should be continued till the next $\mathrm{Mon}(\sigma, TC) = \top$ verdict is received.

Once the novel data recordings are available to the domain experts, they are able to analyse and recognize the traffic situations occurring in the recordings. Formalization of a subset of the identified traffic situations into specifications allows them to discuss and decide, by including them in the traffic catalogue, for which set of traffic situations sufficient data is present and hence should be filtered out during future data recording.

3. Application Example: A Catalogue of 2-Lane Highway Situations

In this example, we assume that the domain experts want to start from scratch and intend to build their traffic catalogue along the way. Their Data Collection Vehicle (DCV) starts collecting data while driving on a two-lane highway. Since initially, there are no situation specifications present in the traffic catalogue ($TC = \emptyset$), hence all data is collected as novel data. After the first day of data collection, the domain experts check the collected data and identify certain traffic situations and decide to include them as traffic situation specifications in the traffic catalogue.

For creation of our example traffic catalogue, we use the Spatial View (SV) formalism from the Traffic Sequence Charts (TSCs) [7] specification language. SVs enable the specification of spatial relations between traffic participants and traffic environment, as well as the constraining of attributes, such as the distance between objects or the acceleration of an object. The semantics of a SV is a logical formula over the object model OM and respective attributes specified. The SVs shown in Figure 2 contain four object symbols that are assigned to objects in the object model OM(see Table 1). We assume the following assignment with (symbol, *unique object identity*, *Class*): (blue car, ego, *Car*), (green car, $other$, *Car*), (right lane, rL, *Lane*), and (left lane, lL, *Lane*). Furthermore, nowhere-boxes (light grey boxes with red borders and dashed crosses) are used, which prohibit the existence of the class specified therein. In the case of Figure 2b, ego is located on the right lane, and no other object instance of the class *Car* should be located behind or in front of ego, nor on the left lane. In this example, the nowhere-boxes present in the Spatial Views have been mapped to the $inRange$ attribute of ego. This attribute returns a list of objects which are present in the range of ego, which can be used to evaluate the logical sub-formula related to nowhere-boxes. Hence, the logical formula derived from the SV in Figure 2b is:

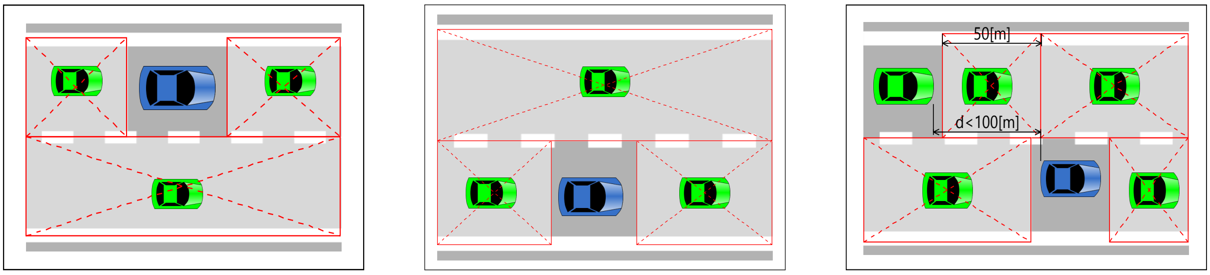
$$S_2 \quad = \quad ego.pos.y > lL.yR \ \wedge ego.pos.y < lL.yL \ \wedge ego.inRange = \emptyset_{Car}$$

where y is an abbreviation for the projection of the positional anchor to its second component.

The three traffic situation specifications identified by the domain experts and specified as Spatial Views are added to the traffic catalogue $TC = \{S_1, S_2, S_3\}$. The corresponding SVs are presented in

Table 1
Object Model OM used in the application example

Class	Attribute	Basic Type	Domain
Lane	yR	yposition	$\mathbb{R}$
	yL	yposition	$\mathbb{R}$
Car	pos	position	$\mathbb{R}^2$
	$inRange()$	function	$\{i \mid i \in \mathcal{C}\}$



(a) S_1: Ego vehicle on the left lane of a two-lane highway

(b) S_2: Ego vehicle on the right lane of a two-lane highway

(c) S_3: Ego vehicle with other vehicle on left lane at a distance of at least $50\,\mathrm{m}$ behind it

Figure 2: Example traffic catalogue consisting of three Spatial Views

Figure 2a, Figure 2b and Figure 2c. For more details about the Spatial View formalism, extraction of logical formula from them and construction of runtime monitors, we refer the readers to [7, 15]

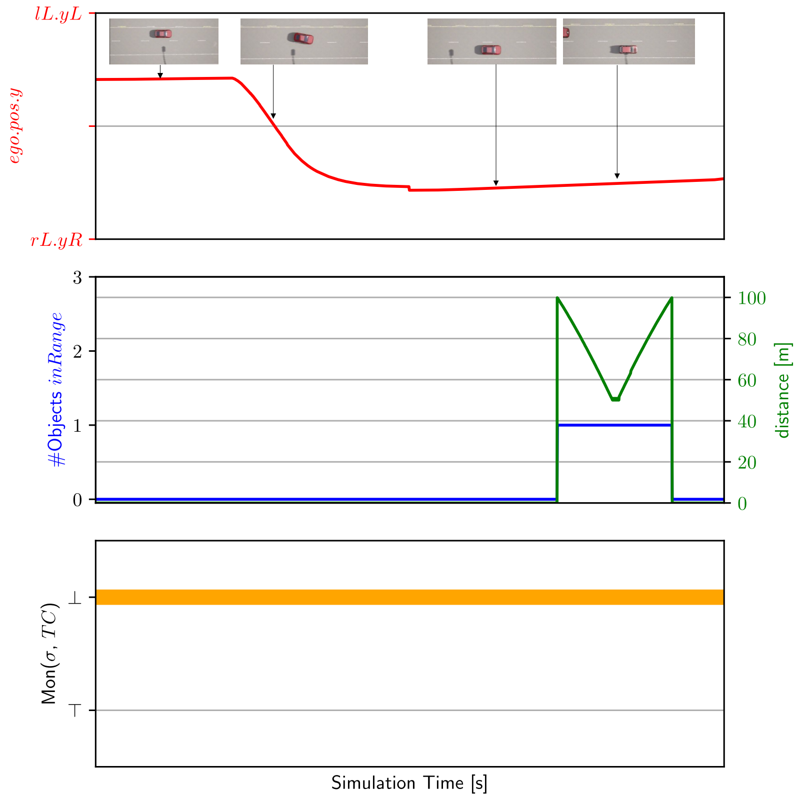
To validate our concept, we generated multiple simulation runs for the example situations described here, in the CARLA [16] simulator. We also implemented runtime monitors using the Spatial View based situation specifications present in the traffic catalogue TC (See Figure 2). While implementing the monitor, the sensor values provided by CARLA were mapped to the respective attribute values in each SV formula. The results from two simulation runs are shown in Figure 3. The first row shows the evolution of y-position of the DCV during the simulation run. In the second row, we present the number of objects of type *Car* detected by the DCV in its sensor's range and the distance from DCV to the detected object. In the final row, the runtime monitor verdicts for each simulation time-step are presented. Figure 3a shows a simulation run when the traffic catalogue is empty. Figure 3b shows a simulation run with the runtime monitors based on the extended TC containing the three Spatial Views shown in Figure 2.

The approach is effective because given a novel traffic situation, one can automatically confirm that the new traffic catalogue includes that situation by checking that at least one new Spatial View is satisfied by that situation. The potential for scalability lies in the fact that a Spatial View usually does not only describe one concrete situation but a whole set of them (generalization), yet comes at the price of experts' time for the formalization. Spatial Views have shown to specify a wide range of traffic situations in the automotive domain [17, 18, 15, 19] (also for maritime [20] and rail [21, 22]) and there is tool support for the creation of SVs and the generation of SV formulas [23]. Executing many runtime monitors with the simulation is not a limiting factor for scalability in practice.

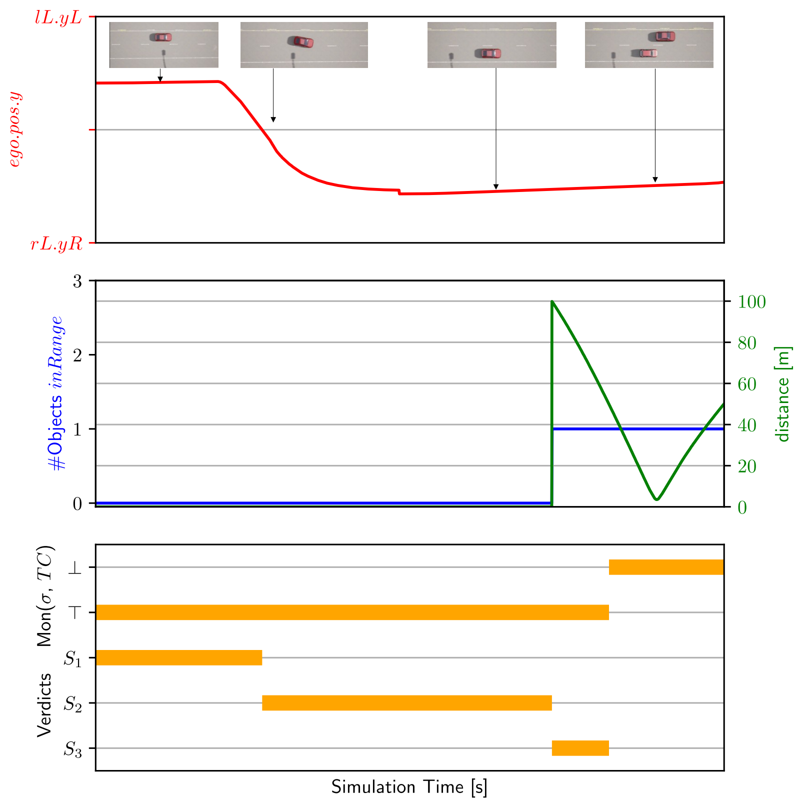
4. Conclusion and Future Work

We have presented a methodology to detect novel traffic situations based on a traffic catalogue that consists of formally specified known traffic situations. This method is useful for developers of Automated Driving Function (ADF) who want to ensure safe operation of their vehicles by only allowing them to operate within their Operational Design Domain (ODD). The method enables a structured construction of the ODD by recording sensor data only when the operating conditions of the vehicle do not correspond to those present in the traffic catalogue. The lack of AI-based components in our approach makes it explainable, traceable, and understandable to humans. We showed how to apply the concept for creating a traffic catalogue by domain experts on the example of 2-lane highway situations.

Future work is to test the approach rigorously on larger simulation and real world datasets (similar to [24], where runtime monitoring using TSCs is demonstrated for a research vessel), and to implement the monitors on automotive-grade hardware. Further efforts are needed to extend the methodology to detect novel traffic scenarios (i.e., sequences of situations), for tool-support for completeness argumentations for the catalogue, and for adapting the technique for use in other transportation domains such as maritime.



(a) DCV simulation run with empty traffic catalogue



(b) DCV simulation run with extended traffic catalogue

Figure 3: Simulation results from the prototypical implementation of our proposed concept

Acknowledgments

The research leading to these results is funded by the German Federal Ministry of Education and Research under grant agreement No 16MEE044 (EdgeAI-Trust) and by the Chips Joint Undertaking under grant agreement No. 101139892 (EdgeAI-Trust).

Declaration on Generative AI

During the preparation of this work, the author(s) used LanguageTool in order to: Grammar and spelling check. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles, Standard SAE J3016 202104, SAE International, 2021.

[2] Road Vehicles — Test scenarios for automated driving systems — Specification for operational design domain, Standard ISO 34503:2023, ISO (International Organization for Standardization), 2023.

[3] Operational Design Domain (ODD) taxonomy for an automated driving system (ADS) – Specification, Standard PAS 1883:2020, The British Standards Institution (BSI), 2020.

[4] A. Shakeri, Formalization of operational domain and operational design domain for automated vehicles, in: 2024 IEEE 24th International Conference on Software Quality, Reliability, and Security Companion (QRS-C), 2024, pp. 990–997. doi:10.1109/QRS-C63300.2024.00131.

[5] J. Reich, D. Hillen, J. Frey, N. Laxman, T. Ogata, D. Di Paola, S. Otsuka, N. Watanabe, Concept and metamodel to support cross-domain safety analysis for odd expansion of autonomous systems, in: J. Guiochet, S. Tonetta, F. Bitsch (Eds.), Computer Safety, Reliability, and Security, Springer Nature Switzerland, Cham, 2023, pp. 165–178. doi:10.1007/978-3-031-40923-3_13.

[6] P. Reis, P. Rigoll, E. Sax, Behavior forests: Real-time discovery of dynamic behavior for data selection, in: 2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC), 2024, pp. 1962–1967. doi:10.1109/ITSC58415.2024.10920178.

[7] W. Damm, S. Kemper, E. Möhlmann, T. Peikenkamp, A. Rakow, Traffic sequence charts - from visualization to semantics (2017). doi:10.13140/RG.2.2.15190.42563.

[8] J. Langner, J. Bach, L. Ries, S. Otten, M. Holzäpfel, E. Sax, Estimating the uniqueness of test scenarios derived from recorded real-world-driving-data using autoencoders, in: 2018 IEEE Intelligent Vehicles Symposium (IV), 2018, pp. 1860–1866. doi:10.1109/IVS.2018.8500464.

[9] R. Greer, M. Trivedi, Towards Explainable, Safe Autonomous Driving with Language Embeddings for Novelty Identification and Active Learning: Framework and Experimental Analysis with Real-World Data Sets, preprint, 2024. URL: http://arxiv.org/abs/2402.07320. doi:10.48550/arXiv.2402.07320, arXiv:2402.07320 [cs].

[10] N. Rajesh, E. Van Hassel, M. Alirezaei, Unknown-unsafe scenario generation for verification and validation of automated vehicles, in: 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC), 2023, pp. 1759–1764. doi:10.1109/ITSC57777.2023.10421804.

[11] ISO, 21448: 2022 road vehicles safety of the intended functionality, International Organization for Standardization in ISO 21448 (2022).

[12] I. Saxena, B. Westphal, M. Fränzle, Towards runtime monitoring of spatial properties of automated driving functions using abstract scene graphs, 2025. Accepted for publication in 12th Young Researchers Seminar (YRS2025), Bergisch Gladbach, Germany, forthcoming.

[13] C. Neurohr, L. Westhofen, T. Henning, T. de Graaff, E. Möhlmann, E. Böde, Fundamental considerations around scenario-based testing for automated driving, in: 2020 IEEE Intelligent Vehicles Symposium (IV), 2020, pp. 121–127. doi:10.1109/IV47402.2020.9304823.

[14] E. Zapridou, E. Bartocci, P. Katsaros, Runtime verification of autonomous driving systems in carla, in: J. Deshmukh, D. Ničković (Eds.), Runtime Verification, Springer International Publishing, Cham, 2020, pp. 172–183. doi:10.1007/978-3-030-60508-7_9.

[15] R. Stemmer, I. Saxena, L. Panneke, D. Grundt, A. Austel, E. Möhlmann, B. Westphal, Runtime monitoring of complex scenario-based requirements for autonomous driving functions, Science of Computer Programming 244 (2025) 103301. doi:10.1016/j.scico.2025.103301.

[16] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, V. Koltun, CARLA: An open urban driving simulator, in: Proceedings of the 1st Annual Conference on Robot Learning, 2017, pp. 1–16.

[17] D. Grundt, A. Köhne, I. Saxena, R. Stemmer, B. Westphal, E. Möhlmann, Towards runtime monitoring of complex system requirements for autonomous driving functions, in: M. Luckcuck, M. Farrell (Eds.), Proceedings Fourth International Workshop on Formal Methods for Autonomous Systems (FMAS) and Fourth International Workshop on Automated and verifiable Software sYstem DEvelopment (ASYDE), Berlin, Germany, 26th and 27th of September 2022, volume 371 of *Electronic Proceedings in Theoretical Computer Science*, Open Publishing Association, 2022, pp. 53–61. doi:10.4204/EPTCS.371.4.

[18] J. S. Becker, Safe linear encoding of vehicle dynamics for the instantiation of abstract scenarios, in: A. E. Haxthausen, W. Serwe (Eds.), 29th International Conference on Formal Methods for Industrial Critical Systems, FMICS 2024, volume 14952 of *Lecture Notes in Computer Science*, Springer, 2024, pp. 3–20. URL: https://elib.dlr.de/205077/. doi:10.1007/978-3-031-68150-9_1.

[19] D. Grundt, A. Rakow, P. Borchers, E. Möhlmann, What does ai need to know to drive: Testing relevance of knowledge, Science of Computer Programming 244 (2025) 103297. doi:https://doi.org/10.1016/j.scico.2025.103297.

[20] A. Austel, M. Steidel, B. Westphal, Formal specification of situations in scenario-based testing of maritime assistance systems, in: European Workshop on Maritime Systems Resilience and Security (MARESEC 2024), 2024. URL: https://elib.dlr.de/204947/.

[21] M. Wild, J. S. Becker, C. Schneiders, E. Möhlmann, A scenario-based simulation framework for testing of highly automated railway systems, in: VEHITS 2025, Proceedings of the 11th Internationa Conference on Vehicle Technology and Intelligent Transport Systems - VEHITS, SciTePress, 2025, pp. 88–99. URL: https://elib.dlr.de/212746/. doi:10.5220/0013286600003941.

[22] J. S. Becker, M. Wild, A. Rakow, E. Möhlmann, How far can we go? capturing traffic rules via traffic sequence charts, 2025. Accepted for publication in Law meets engineering for trustworthy Autonomous Vehicles (LawIng4AV) workshop proceedings, Baden-Baden, Germany, forthcoming.

[23] P. Borchers, T. Koopmann, L. Westhofen, J. S. Becker, L. Putze, D. Grundt, T. de Graaff, V. Kalwa, C. Neurohr, Tsc2carla: An abstract scenario-based verification toolchain for automated driving systems, Science of Computer Programming 242 (2025). URL: https://elib.dlr.de/211153/. doi:10.1016/j.scico.2024.103256.

[24] A. Austel, L. Panneke, J. Piotrowski, N. Wetzig, M. Steidel, B. Westphal, Using monitoring of maritime traffic scenarios in the validation of maritime systems, 2025. Accepted for publication in IEEE Symposium on Maritime Informatics and Robotics (MARIS'25) proceedings, Syros, Greece, forthcoming.