Using ontologies for dataset engineering in automotive AI applications

Martin Herrmann[1], Christian Witt[2], Laureen Lake[1], Stefani Guneshka[3],
Christian Heinzemann[1], Frank Bonarens[4], Patrick Feifel[4], Simon Funke[3]
[1]Robert Bosch GmbH, firstname.lastname@de.bosch.com
[2]Valeo Schalter und Sensoren GmbH, christian.witt@valeo.com
[3]Understand AI, stefani@understand.ai, simon.funke@understand.ai
[4]Stellantis, Opel Automobile GmbH, frank.bonarens@stellantis.com

***Abstract*—Basis of a robust safety strategy for an automated driving function based on neural networks is a detailed description of its input domain, i.e. a description of the environment, in which the function is used. This is required to describe its functional system boundaries and to perform a comprehensive safety analysis. Moreover, it allows to tailor datasets specifically designed for safety related validation tests. Ontologies fulfill the task to gather expert knowledge and model information to enable computer aided processing, while using a notion understandable for humans. In this contribution, we propose a methodology for domain analysis to build up an ontology for perception of autonomous vehicles including characteristic features that become important when dealing with neural networks. Additionally, the method is demonstrated by the creation of a synthetic test dataset for an Euro NCAP-like use case.**

***Index Terms*—ontology, dataset engineering, autonomous driving, artificial intelligence, neural network**

I. Introduction

Automated driving functions rely on perception functions that identify and classify all relevant elements in the vehicle's environment for making safe behavior decisions. Increasingly, such perception functions are realized using deep neural networks (DNNs). One of the greatest challenges for using these technologies in safety-critical applications like automated driving is creating a thorough and compelling safety argumentation [1]. The project KI Absicherung[1] (KI-A) is working on establishing such a stringent and provable safety argumentation for AI-based applications in highly automated vehicles with a specific focus on camera-based pedestrian detection in urban environments.

A basis for such a safety argumentation is a detailed description of the input domain of the perception function, i.e., a description of the environment in which the function is used [2]. Such a description of the input domain can be used to define intended operating conditions and functional system boundaries, and it allows to design and analyze datasets with specific content for safety related tests [3] that can be used as part of the safety argumentation. As such, the description of the input domain needs to be both understandable by human (safety) engineers and amenable to computer-aided processing, for example, for automated data generation and annotation.

Ontologies [4] as a knowledge representation format may satisfy the aforementioned requirements. They enable to model the concepts in a domain including properties of these concepts and semantic relationships between these concepts. As such, they are particularly suitable for describing the semantic contents of data including the semantic interplay of the contents (e.g., that a particular pedestrian holds a specific object thereby interacting with another pedestrian in a specific way). The power of an ontology lies in its foundation on formal logic that enables to formalize rules that can be used by reasoners for explicating knowledge that is implicitly contained in the ontology and tested for logical errors.

In the project KI Absicherung, we use ontologies for (i) creating machine-processible descriptions of (synthetic) images to be generated, (ii) providing semantic annotations of image contents, (iii) defining an operational design domain [5], and (iv) defining the taxonomy for the safety argumentation.

In this paper, we present a methodology for analyzing the input domain of a perception function and for deriving an ontology from the result of the analysis. The resulting ontology has a modular structure and formalizes most domain concepts such that the ontology is amenable to the use of automated reasoning. We illustrate our methodology and the use of the ontology based on the use case of generating synthetic image data for testing a perception function on an Euro NCAP[2]-like test scenario.

The remainder of the paper is organized as follows: Section II covers the motivation and main use cases of an ontology-based data engineering approach. Section III then elaborates on the developed strategy for creating an ontology for a DNN-based perception function. The results of image generation based on Euro NCAP-like scenarios using our ontology as well as an introduction of another two applications of our ontology are provided in Section IV. Section V captures an overview of related work on ontologies in the automotive area and Section VI concludes the paper by summarizing main contributions.

The research leading to these results is funded by the German Federal Ministry for Economic Affairs and Energy within the project "Methoden und Maßnahmen zur Absicherung von KI basierten Wahrnehmungsfunktionen für das automatisierte Fahren (KI Absicherung)". The authors would like to thank the consortium for the successful cooperation.

[1]https://www.ki-absicherung-projekt.de/en/

[2]European New Car Assessment Programme https://www.euroncap.com

II. Main Use Cases

We built the ontology for supporting two main use cases in KI Absicherung. The first use case is labeling of training and test data. The second use case, on which we will focus in the remainder of this paper, is the systematic generation of synthetic data sets. For this use case, we utilize the ontology for automatically deriving data specifications that, in turn, can be processed by a rendering pipeline for creating variations of a scenario. In particular, we use the ontology to identify and vary specific parameters that are applicable to such a scenario.

In the following, we show how to leverage ontology information to request images for a safety relevant Euro NCAP-like scenario. We define a rough base scenario where a pedestrian crosses a road in front of the ego vehicle from the near side between two parked cars. From earlier work in our project, we know that important context elements for pedestrian detection are, among others, occlusion, light & contrast and pedestrian pose. Furthermore it has been shown that DNN-based models are prone to *texture bias* and do not incorporate shape information like humans would expect [6]. Therefore, we strive for deriving such variations automatically from our ontology and to incorporate them into the image definitions for the scenario.

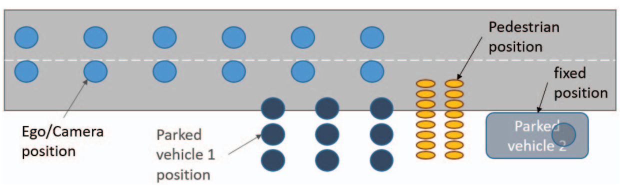

Fig. 1. Grid based positions to vary pedestrian occlusion in Euro NCAP-like scenario

Figure 1 shows a grid-based setup for this scenario, where we define different positions for the first parked car, the pedestrian and ego vehicle (camera). The different positions on the grid allow to change the occlusion ratio of the pedestrian. Apart from the geometric setup we also vary the global illumination, the vehicle types and colors. Hence, many systematic variations can be derived from our ontology. The goal was to develop a set of images with many different occlusion and illumination variations that can be further processed by a test strategy. Fig. 2 shows four generated images which have been produced based on our tooling by Mackevision[3]. A main advantage compared to expert-based scene creation is the automated creation based on formalized knowledge.

III. Methodology

In the following, we present a strategy for creating an ontology for a DNN function. Our strategy spans the entire process from analyzing the input domain until the derivation of the actual ontology. The strategy leading to our methodology has been developed in a structured approach with several steps.

[3]https://www.mackevision.com

Fig. 2. Pedestrian crosses a road between two parked vehicles. Synthetic Euro NCAP-like images produced by Mackevision with systematic variation of pedestrians, poses, positions, vehicle types, colors and illumination.

Review of Public Data Sources and Existing Standards

In the first step, existing standards targeting (part of) the domains under analysis are reviewed.

In our project use case, we aim at pedestrian detection in urban traffic situations based on camera and lidar sensors. Consequently, we review standards regarding the specification of traffic scenarios like OpenDrive [7] and OpenScenario [8] or ODD specification [9]. In addition, perception is intuitively concerned with the surface properties of the objects that need to be detected. Thus, a standard like glTF [10], but also checklists like CV-HAZOP [11] are additionally reviewed. Further considered documents are normative and law documents that restrict, e.g., the appearance and positioning of traffic infrastructure like traffic lights, traffic signs, etc. In our case, we extracted a first list of relevant domain elements that was used in the further steps.

As additional data source the GIDAS (German In-Depth Accident Study[4]) database has been analyzed. The accident classes used in GIDAS for accident classification may serve as a high-level grouping of base scenarios.

Workshop with Experts from Different Fields of Expertise

In a second step, parallel to reviewing public data sources, an initial brainstorming workshop with experts from different fields of expertise is performed to derive an initial set of relevant context dimensions. Hereby, context dimensions are elements or properties that can be varied to achieve a specific effect on a DNN function like weather conditions or degree of occlusion.

In our case, we used project workshops to let DNN, simulation, sensor, and safety experts write down context dimensions that they consider important. This analysis was complemented by an alignment with a corner case collection. The outcome was again evaluated in common expert workshop sessions.

[4]https://www.gidas.org

Structuring of Initial Results in Context Element Categories

The first two steps typically yield an already high number of context elements. Usually, working with and maintaining such a list in a flat manner is difficult. In a third step, we aimed at deriving a structuring of the obtained context elements into different categories. As a basis, we used the layer-based structuring of a context from the Pegasus project [12]. Since the Pegasus project had no focus on perception and, in particular, on DNN-based perception functions, we extended and refined the Pegasus proposal with two additional layers: materials and sensor characteristics, see Fig. 3. Please note that Pegasus layer 3 and 6 are not in scope for KI-A and thus not further considered.

1	Road level	roads, lanes, sidewalks, markings, parking spaces, ...	Pegasus Layers
2	Road-side Infrastructure	signs, traffic lights, poles, bus stops, buildings, ...	
3	Temporary Modification	roadwork signs, temporary markings, fallen trees, ...	
4	Dynamic Objects	vehicles, pedestrians, animals, objects on road (ball, trash), ...	
5	Environmental Conditions	illumination (brightness, angle, color), weather (sun, rain, snow, wind, fog), ...	
6	Digital Information	state of traffic lights, V2X messages, cell network coverage, ...	
7	Materials	surface characteristics, patterns, reflectance, brightness, ...	Visual appearance
8	Sensor Characteristics	resolution, distortion, noise, saturation, sensor position, ...	

Fig. 3. Ontology layers based on Pegasus model (blue) and extended by visual appearance layers (green). The gray layers are not in scope of KI-A.

Expert Interviews for each Context Element Category

The initial set of context elements and categories are then refined in expert interviews in a fourth step. The goals of this step are (i) to identify further important context elements, (ii) to derive a number of possible manifestations (also called variations) for each element, and (iii) to get a first expert opinion on importance of the different context element categories.

A variation of a context element is one possibility how this context element may occur in reality. The derivation of all (relevant) variations for each context element yields a way to describe the possible occurrences of the context element in the real world. To this end, we need both a description understandable by humans as well as a grounding into physical units that can be used in the data generation process or for data labeling. As an example, we might describe the daytime in human understandable terms as night, dawn, day, and dusk. This requires a grounding to physical variables with value ranges that define, for example, height of the sun, ambient light intensity, brightness of the sun, and so on.

For pre-structuring and focusing the data generation, a hypothesis and voting on important context element categories can be created. This is beneficial, because the dimensionality of the input domain is usually vast such that prioritization and explicit handling of the high complexity are required. In our understanding, important context elements are those where variations have caused malfunctioning of DNN functions in the past. While the prioritization is based on past experiences and expert guesses in this step, the information on important context elements needs to be complemented later on with experiments on actual data for the currently addressed DNN function.

Consolidation of context elements

To capture all relevant aspects of the input domain we used SCODE Analyzer [13] and Zwicky boxes [14]. Each context element category is represented as one Zwicky box. Each relevant context element is entered as a dimension in the form of a line. The different manifestations or value ranges of a dimension are captured in this line by exclusive alternatives, see also extract of pedestrian Zwicky box in Table I. In our practical work, this compact representation of high dimensional spaces turned out to be very helpful to collect, discuss and consolidate the input domain.

TABLE I
EXTRACT FROM PEDESTRIAN ZWICKY BOX. TO IMPROVE READABILITY ONLY SOME DIMENSIONS ARE SHOWN[5].

Dimension	Alternatives				
Age	child	teenager	adult	old person	
Gender	male	female	other		
Body shape	thin	normal	muscular	obese	
Body height	<80cm	80cm-120cm	120cm-160cm	160cm-200cm	>200cm
Pigmentation	high	medium	low		

The consolidation of context elements is performed iteratively and alternating with the expert interviews for the different context element categories. The consolidation has two major goals: identifying shared context elements and grouping coherent information while factoring out shared information. Performing such a consolidation is key for obtaining a usable and well-structured input domain model.

A prime example for a shared context element is color, since color is used at many different locations in the input domain model. Examples include hair color of a human, color of object surfaces, colors of clothes, color of the emitted light of a light source. For this reason, the description of color has been factored out into a separate Zwicky box.

In our case the resulting domain model after this step consists of 22 Zwicky boxes with around 250 dimensions and 1000 alternatives.

Along with the consolidation, the relationships between the different resulting Zwicky boxes have been documented and thereof, the main cluster relations are visualized in an overview figure (see Fig. 6). This figure proved to be useful for providing an overview of the created input domain model.

[5]Pedestrian dimensions not shown in Table I: Pose, Skin modification, Hair length, Hair color, Hair style, Beard size, Face shape, Special handicap, Body type, Body & Face position to sensor.

Grounding in Physical / Measurable Units

The input domain model shall be used for specifying data request for the production of synthetic data and to label existing data. This requires a clear understanding of the meaning of each context dimension (in literature also referred to as the semantics). Therefore, we strived at providing a grounding of each context dimension in a physical and/or measurable quantity with a defined unit. An example is given by the context dimension *rain* in the Zwicky box for weather conditions. For this context dimension, we chose the amount of water falling from the sky measured in the unit $\frac{mm}{h}$ for grounding the context dimension as also proposed in [9]. As stated above, we found it helpful to define a number of variations for each context dimension that capture in human understandable terms the possibilities how this context dimension may manifest in reality. Then, the different variations of the context dimension need to be associated to mutually exclusive values or intervals over the quantity. Wherever possible and useful, the grounding should be based on existing external definitions. In our example in Fig. 4, values for rain were obtained from the German Weather Service (Deutscher Wetterdienst, DWD). The grounding should

no	light	medium	strong	heavy
< 0.1	0.1 - 5	5 - 15	15 - 25	> 25

Fig. 4. Value ranges for dimension rain. All values in $\frac{mm}{h}$.

support objective measurement, either within a vehicle or by reference sensors for being able to associate concrete scenes with value ranges.

Refinement based on Data Analysis, Assurance Case, and Test

We applied an iterative process as shown on the left side in Fig. 5 to create the domain model using Zwicky boxes.

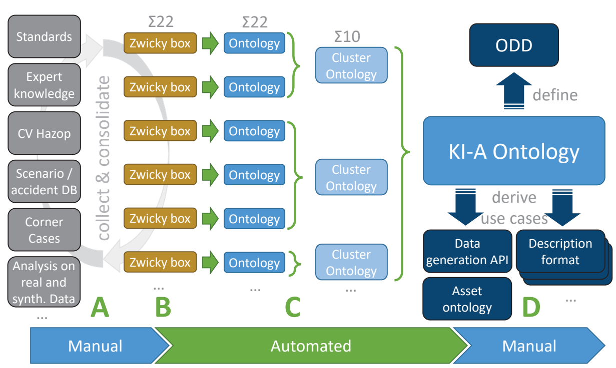

Fig. 5. Applied methodology to develop the KI-A ontology with manual and automated steps from A to D (description in text).

The creation of the domain model (A) provides a top-down structuring of the available knowledge on important context element categories and their variations. As described in the previous sections, this knowledge originates from expert knowledge and other available data sources. This knowledge needs to be challenged and substantiated by a bottom-up data analysis that either confirms or refutes the importance of context dimensions and that may also identify additional context dimensions that need to be taken into consideration. For example, the corner case search gives an indication on which additional context dimensions could be considered. Impossible combinations of variations can also be excluded from the model.

Derivation of an Ontology

So far, the domain knowledge in terms of context elements and variations has been maintained in Zwicky boxes (B). To transfer the knowledge from the existing collection of Zwicky boxes into an actual ontology, we developed a set of tools (C) that automatically generates an ontology[6] out of a Zwicky box and merges them into clusters. This is depicted by green arrows and brackets in Fig 5. A cluster contains one or more Zwicky boxes that are closely related topicwise, see Fig. 6 for more details. The clustering keeps the number of ontologies low, the scope light-weighted and allows for concurrent modifications. In addition, we generate a main ontology, called KI-A Ontology, that imports all cluster ontologies. In contrast to Zwicky boxes, the ontology additionally captures the relations between multiple sub ontologies (exemplary represented as labeled arrow connections in Fig. 6). Those relations are modeled within the main ontology, where an initial version of these relationships has been created automatically during the generation process.

Fine-Tuning the Ontology

After the generation step, the cluster ontologies as well as the main KI-A ontology contain all context dimensions and their variations along with their grounding into physical units. In the last step (D), additional superclasses and relationships between these superclasses need to be added manually to the ontology.

The resulting main ontology, the KI-A ontology, has around 890 classes, 2200 logical axioms[7] and 270 object properties.

We performed several reviews and re-structuring steps of the different cluster ontologies to facilitate readability and usability. More precisely, we added several new super-classes to introduce further hierarchy, we simplified complex classes by introducing new object properties and additional classes with subclasses like shown in Fig. 7. Additionally, common data properties and several super-object properties for structuring the object properties of the different cluster ontologies were relocated into a separate *utility ontology* that is imported into each cluster ontology.

Finally, we integrated ontology reasoners to infer the corresponding class from a value. If e.g. the description of a scene contains a rain precipitation value of $4.2\frac{mm}{h}$ the reasoner will infer this to the *rain light* class (see Fig. 4). This feature facilitates significantly the description of assets and scenes. For the future we are planning to implement more logic, using Semantic Web Rule Language (SWRL) rules, where reasoning will play a significant role.

[6]We used RDF (Resource Description Framework) format during the automated process steps.

[7]Rules in an ontology are called axioms, e.g. $Adult <=> age > 17$.

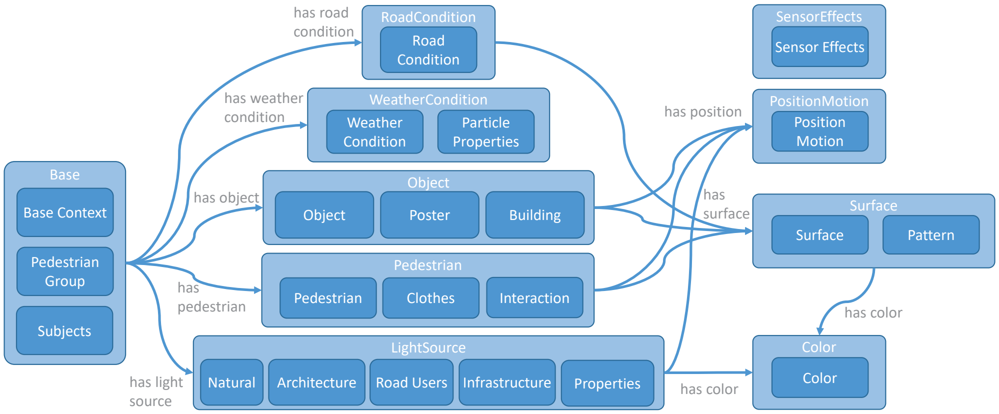

Fig. 6. KI-A Ontology: 22 ontologies (blue) bundled into 10 clusters (light blue). To improve readability only main cluster relations are shown.

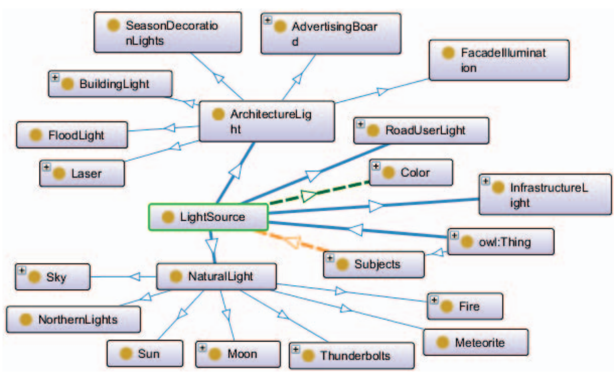

Fig. 7. KI-A Ontology: Some classes of the light source cluster ontology

IV. Results and Further Applications

In the following, we describe some results for the data generation of our Euro NCAP-like scenario and introduce two further applications of our ontology, namely defining an operational design domain (ODD) and labeling 3D assets.

Data Generation for Euro NCAP-like Scenario

To analyze complexity, we derived a Zwicky box by selecting 13 dimensions and up to 20 alternatives per dimension from our ontology, considering classes like PersonOcclusion, SunElevation, Sky, Pose, Color, RelativePosition, and so on. This Zwicky box spans a combinatorial space of 478 billion combinations or 27.6 billions when some color and illumination constraints are considered. As our budget of images was restricted to only 7500 frames we applied combinatorial testing [3] to reduce the number of combinations. We finally used 3-wise combinatorial testing which results in 6669 images and two times 2-wise with 408 images at different positions in the map to exhaust our budget as much as possible[8].

With this application we have shown that we are able to derive an interface from our ontology to automatically request images and build up test datasets that are part of the safety argumentation.

[8] $1 * 6669 + 2 * 408 = 7485$ images ($99,8\%$ of budget).

ODD definition

The Operational Design Domain (ODD [5]) for the overall system architecture within the consortia project has been defined based on KI-A ontology. Single ODD elements reference their corresponding ontology classes.

Asset description

Based on our KI-A ontology, an asset ontology was created that is used to define and store assets from our data generation pipeline as instances. Assets are hereby 3D models e.g. pedestrians, cars, buildings, but also street structures or different sky maps that are used for synthetic image generation.

We use an asset ontology to describe these assets, i.e., they are defined in our ontology by creating individuals and corresponding properties. As each asset has a universally unique identifier (UUID), the image description just needs to reference the UUID instead of labeling the asset in detail each time it appears on an image, as shown in Fig. 8.

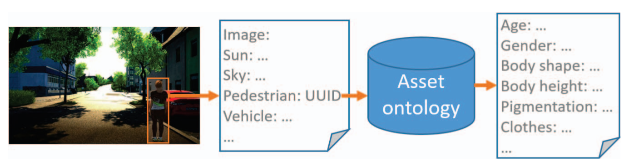

Fig. 8. Usage of asset ontology to get detailed information of a pedestrian.

Especially for pedestrians the description is quite detailed, taking into account potentially negative effects of pedestrian shape, texture and color on DNN applications in addition to the aforementioned dimensions like pedestrian position (distance and occlusion) and characteristics. Apart from the main aspects described in Table I, the description therefore includes the pedestrian interactions, clothes, the clothing surface and color. As the number of assets is limited and thus the same assets are visible in many images, the asset ontology approach prevents storing redundant information. The asset ontology is also used to analyze distributions of asset properties. The results can be used to develop new assets to complement the asset collection with missing combinations of properties.

V. Related Work

Ontologies as a means to structure knowledge were implemented for automotive use cases in various works before (Li et al. [15], Bagschik et al. [16], Neurohr et al. [17]). However, a widespread use is still to be reached, just as standardized ontologies for autonomous driving functions. Similar to the KI-A ontology, the openXOntology is developed at ASAM (Association for Standardization of Automation and Measuring Systems) with the aim of harmonizing description language across all ASAM standards. While KI-A focuses on safeguarding of autonomous vehicles which is reflected in the KI-A ontology, openXOntology has a broader scope which is also integrating different standards such as openODD and openScenario. Still, the overlap in content is very strong, especially due to the modular design and the use of reasoning. At the time of writing, openXOntology is nearing completion and a consolidation of the results of KI-A and openXOntology is crucial for future research.

The Pegasus project[9] defined a model for a systematic description of scenarios with six independent layers. We used this layer model and extended it for our use case. Based on the work in the Pegasus project Bagschik et al. [16] proposed a generation of traffic scenes in natural language. They focus on scenario creation for autonomous driving functions based on ontologies. Their ontology enables to generate road layouts and dynamic motorway scenarios. The objects and elements that represent the driving scene are arranged in a high level structure of different layers. Menzel et al. [18] extended this work to have a consistent terminology for the identified levels of abstraction which fulfil the requirements of the ISO26262 standard.

Schwalbe et al. [19] propose ontologies to cover the semantic input space in the context of DNN verification and validation, yet without focus on using them for data generation.

VI. Conclusion

This paper describes an approach to analyze and define the input domain of the perception task for an autonomous vehicle. In particular, we define a methodology to build up an ontology for the domain pedestrian perception in urban contexts. We show how to set up a semi-automatic process to transform domain elements from Zwicky boxes to ontologies, to clusters and finally to our main ontology. Our ontology is a controlled vocabulary of jargon in the KI Absicherung project and serves as a basis for different use cases. We demonstrate how our ontology can be used to build up an ODD, describe asset collections, define performance limiting factors, and to derive test images sets thereof. These test sets play a key role in the safety argumentation in DNN applications for autonomous vehicles.

[9]https://www.pegasusprojekt.de/en/home

Acknowledgment

We thank Frédérik Blank from Robert Bosch GmbH for the cross work package coordination on this topic and several contributions throughout this work. We also thank Thomas Schulik from ZF Friedrichshafen AG for the idea to generate images based on an Euro NCAP-like scenario.

References

[1] S. Burton, L. Gauerhof, and C. Heinzemann, "Making the case for safety of machine learning in highly automated driving," in *Computer Safety, Reliability, and Security : SAFECOMP 2017 Workshops, ASSURE, DECSoS, SASSUR, TELERISE, and TIPS*, ser. Lecture Notes in Computer Science. Cham: Springer, 2017, vol. 10489, pp. 5–16.

[2] L. Gauerhof, R. Hawkins, C. Picardi, C. Paterson, Y. Hagiwara, and I. Habli, "Assuring the safety of machine learning for pedestrian detection at crossings," in *Computer Safety, Reliability, and Security*. Cham: Springer International Publishing, 2020, pp. 197–212.

[3] C. Gladisch, C. Heinzemann, M. Herrmann, and M. Woehrle, "Leveraging combinatorial testing for safety-critical computer vision datasets," in *Workshop on Safe Artificial Intelligence for Automated Driving*, 2020.

[4] S. Staab and R. Studer, Eds., *Handbook on Ontologies*, 2nd ed. Springer Berlin Heidelberg, 2009.

[5] P. Koopman and F. Fratrik, "How many operational design domains, objects, and events?" in *Workshop on AI Safety @ AAAI 2019*, 2019.

[6] R. Geirhos, C. R. M. Temme, J. Rauber, H. H. Schütt, M. Bethge, and F. A. Wichmann, "Generalisation in humans and deep neural networks," *CoRR*, vol. abs/1808.08750, 2018. [Online]. Available: http://arxiv.org/abs/1808.08750

[7] ASAM - Association for Standardization of Automation and Measuring Systems, "ASAM - OpenDrive," 2019, https://www.asam.net/standards/detail/opendrive.

[8] ——, "ASAM - OpenScenario," 2021, https://www.asam.net/standards/detail/openscenario.

[9] BSI, "PAS 1883:2020 Operational design domain (ODD) taxonomy for an automated driving system (ADS). specification," 2020, https://standardsdevelopment.bsigroup.com/projects/2019-03092#/section.

[10] The Khronos Group Inc., "Graphics Language Transmission Format," 2017, https://github.com/KhronosGroup/glTF/blob/master/specification/2.0/README.md.

[11] O. Zendel, M. Murschitz, M. Humenberger, and W. Herzner, "CV-HAZOP: Introducing test data validation for computer vision," in *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE.

[12] M. Scholtes, L. Westhofen, L. R. Turner, K. Lotto, M. Schuldes, H. Weber, N. Wagener, C. Neurohr, M. H. Bollmann, F. Körtke, J. Hiller, M. Hoss, J. Bock, and L. Eckstein, "6-layer model for a structured description and categorization of urban traffic and environment," *IEEE Access*, 2021.

[13] ETAS GmbH, "SCODE-ANALYZER Software for describing and visualizing complex closed-loop control systems," 2019, https://www.etas.com/scode.

[14] M. Bitzer, M. Herrmann, and E. Mayer-John, "System CO-DEsign (SCODE): Methodology for the analysis of hybrid systems," *at - Automatisierungstechnik*, vol. 68, pp. 488–499, 06 2020. [Online]. Available: https://doi.org/10.1515/auto-2020-0037

[15] Y. Li, J. Tao, and F. Wotawa, "Ontology-based test generation for automated and autonomous driving functions," *Information and Software Technology*, vol. 117, Jan. 2020.

[16] G. Bagschik, T. Menzel, and M. Maurer, "Ontology based scene creation for the development of automated vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1813–1820.

[17] C. Neurohr, L. Westhofen, M. Butz, M. H. Bollmann, U. Eberle, and R. Galbas, "Criticality analysis for the verification and validation of automated vehicles," *IEEE Access*, vol. 9, pp. 18 016–18 041, 2021.

[18] T. Menzel, G. Bagschik, and M. Maurer, "Scenarios for development, test and validation of automated vehicles," *CoRR*, vol. abs/1801.08598, 2018. [Online]. Available: http://arxiv.org/abs/1801.08598

[19] G. Schwalbe, B. Knie, T. Sämann, T. Dobberphul, L. Gauerhof, S. Raafatnia, and V. Rocco, "Structuring the safety argumentation for deep neural network based perception in automotive applications," in *Computer Safety, Reliability, and Security. SAFECOMP 2020 Workshops*. Cham: Springer International Publishing, 2020, pp. 383–394.