Deliverable D08

Advances on the Criticality Analysis for Automated Driving Systems

Version 1.0

Editor Dr. Christian Neurohr

Project coordinator Robert Bosch GmbH und BMW AG

Due date 31.07.2023

Creation date 26.07.2023

Publication 31.12.2023

Approval INT, LI, PU

INT: Release within the project, partners, scientific subcontractors and project sponsors (concerns the application of concepts)

LI: Release VDA-LI, other projects of the project families and members of the VDA lead initiative (concerns conceptual results)

PU: Release to public

Gefördert durch:

Bundesministerium für Wirtschaft und Klimaschutz

aufgrund eines Beschlusses des Deutschen Bundestages

Dokument Information

Authors

Christian Neurohr – DLR e.V., Institute of Systems Engineering for Future Mobility, Oldenburg
Lukas Westhofen – DLR e.V., Institute of Systems Engineering for Future Mobility, Oldenburg
Martin Butz – Robert Bosch GmbH, Renningen
Martin Bollmann – ZF AG, Friedrichshafen
Lina Putze – DLR e.V., Institute of Systems Engineering for Future Mobility, Oldenburg
Tjark Koopmann – DLR e.V., Institute of Systems Engineering for Future Mobility, Oldenburg
Roman Gansch – Robert Bosch GmbH, Renningen
Michael Knoop – Robert Bosch GmbH, Abstatt
Armin Rasch – ZF AG, Friedrichshafen
Bogdan Cojocaru – ZF AG, Friedrichshafen
Johannes Daube – ZF AG, Friedrichshafen

Reviewer

TP2

Contact

Dr. Christian Neurohr
German Aerospace Center (DLR) e.V.
Institute of Systems Engineering for Future Mobility
Escherweg 2
26121 Oldenburg, Germany

Phone: +49 441 770507-217
Email: christian.neurohr@dlr.de

Advances on the Criticality Analysis for Automated Driving Systems⋆

Christian Neurohr[1], Lukas Westhofen[1], Martin Butz[2], Martin Bollmann[4], Lina Putze[1], Tjark Koopmann[1], Roman Gansch[2], Michael Knoop[3], Armin Rasch[4], Bogdan Cojocaru[4], Johannes Daube[4]

[1] German Aerospace Center (DLR) e.V., Institute of Systems Engineering for Future Mobility, Escherweg 2, 26129 Oldenburg, Germany
{firstname.lastname}@dlr.de
[2] Robert Bosch GmbH, Renningen, Germany
[3] Robert Bosch GmbH, Abstatt, Germany
{firstname.lastname}@de.bosch.com
[4] ZF AG, Friedrichshafen, Germany
{firstname.lastname}@zf.com

Abstract. In this work, we summarize the progress regarding the methodical *criticality analysis*, an open context analysis with the goal to structure the operational domain of an automated driving systems with respect to the emergence of criticality by eliciting a manageable set of artifacts. The criticality analysis is developed in the VVM project and it has been featured in numerous publications. In order to provide an overview of the developments of recent years, we provide summaries of the works that have already been published, but also write up ideas and present results that were not available before. In particular, we analyze which research questions could be solved to an satisfactory extent, where key problems remain unsolved and how the artifacts produced by a criticality analysis can be leveraged for subsequent activities regarding the safeguarding of automated driving systems.

Keywords: Criticality Analysis, Automated Driving Systems, Open Context, Verification and Validation, Criticality Metrics, Ontologies, Causality

⋆ The research leading to these results is funded by the German Federal Ministry for Economic Affairs and Climate Action within the project 'Verification & Validation Methods for Automated Vehicles in Urban Environments'. The authors would like to thank the consortium for the successful cooperation.

Table of Contents

1 Introduction

In their concept paper, Neurohr et al. introduce a method called *criticality analysis* that, given a class of *automated driving systems* (ADSs) and an *operational domain* (OD), produces a finite and manageable set of artifacts that explain the emergence of criticality in traffic, in general, and for automated driving, in particular [1]. The proposed approach combines expert-based and data-driven methods to identify relevant phenomena and explain the underlying causalities. Leveraging on abstraction, a criticality analysis converges towards a manageable set of artifacts based on two nearby assumptions on the nature of traffic. As the OD is analyzed for a class of systems, rather than for a concrete realization, it is relevant for any ADS-equipped vehicle operating within that domain. Therefore, its results can subsequently be used to derive safety principles and mitigation mechanisms for ADSs and to set up a coherent safety argument for the homologation process.

In this work, we review recent advances regarding the criticality analysis for ADSs. Based on years of experience, the authors provide an update of its procedure and evaluate, for each branch and each process step, the works that realize, extend, and refine the original concept of the criticality analysis. Therefore, in this document, we provide

i) a detailed update of the criticality analysis' procedure and its process steps,
ii) summaries of publications that instantiated these process steps for all three branches of the criticality analysis,
iii) hints to research questions that remain unsolved, and
iv) a discussion how the criticality analysis' results can be used downstream in a development process.

The manuscript is organized as follows: In Section 2 we recall the framework and basic definitions from the criticality analysis, followed by updates regarding its procedure on a methodical level. Thereafter, we collect, in detail, the advances on the criticality analysis and its conduction split up according to its three branches, i.e. the method branch in Section 3, the information branch in Section 4, and the scenario branch in Section 5 respectively. In Section 6 we discuss how the downstream usage of artifacts resulting from a criticality analysis in an ADSs' development process. Finally, we elaborate on how these advances on the criticality analysis were perceived by the scientific community in Section 7, before concluding with Section 8.

2 General Considerations

Before we dive into the progress regarding the content and conduction of the process steps of the criticality analysis, in this section, we provide a brief overview of high-level changes to its procedure.

Firstly, note that the definition of the term *criticality* regarding traffic situations remained, although applied frequently, unchanged over the past years, indicating that the definition has some merit to it. Let us recall its definition [1, Definition 1], here combined with the elaborating remark [1, Remark 1]:

Definition 1 (Criticality). *Criticality (of a traffic situation) is the combined risk of the involved actors when the traffic situation is continued.*

(i) In order to determine criticality, probabilities and types of harm, dynamical and behavioral models and actions restrictions of the involved actors are taken into account.
(ii) The time-horizon of criticality of a situation is bound by the fulfillment of the intentions of the involved actors.
(iii) Criticality is inversely correlated with the amount of (sequences of) actions to avoid harm that are available to the involved actors.

The extension of criticality from situations to scenarios is done via aggregation over time, using appropriate functions.

The main goal of the criticality analysis in the context of ADSs was represented concisely by [1, Figure 1]:

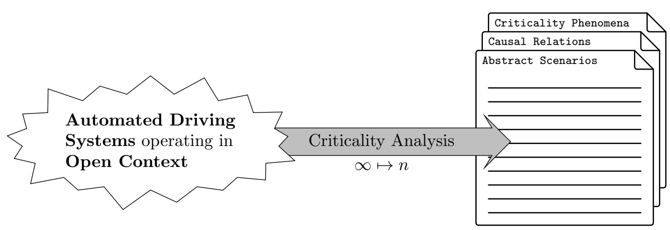


Fig. 1: The goal of the criticality analysis is to map the infinitely-dimensional open context to a finite set of artifacts by analyzing the underlying structures.

This was extended by the *basic concept* [1, Figure 3], which is left out in this work as it is subsumed in the detailed flowchart, cf. [1, Figure 4], which has been refined over the course of the VVM project, resulting in Figure 2 and explained in the following subsections.

Let us recall, that the criticality analysis, as a procedure, operates under two *fundamental assumptions*, cf. [1, Section 4]:

(A1) The number of relevant criticality phenomena is limited and manageable.
(A2) The relevant criticality phenomena leave traces in a growing information basis.

The following section will provide evidences, albeit non-conclusive, that these assumption might hold true. At least for automated passenger cars at SAE Level 4/5 within the OD *urban areas in Germany*, relying on a sensor setup comprised of camera, radar, and lidar sensors. In particular, Section 3.1.2 provides evidence

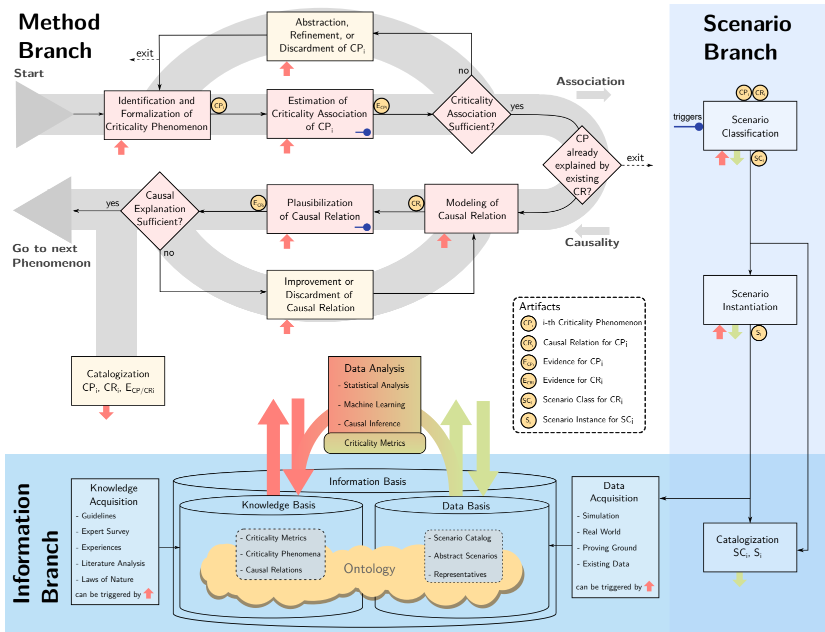


Fig. 2: Overview of the revised procedure of the criticality analysis.

for (A1) and (A2), Section 3.1.4, provide evidence for (A2), and Section 4.2 for (A1), Section 4.2.2 for (A1) – to name some contributions.

The three branches that structure the procedure of the criticality analysis, i.e. method branch, information branch, and scenario branch, were already used to structure the concept paper, cf. [1, Section 5]. Hence, we reuse this structure to present the *advances on criticality analysis* for ADSs along these branches in Section 3, Section 4, and Section 5, respectively.

2.1 Updates to the Method Branch

First, let us mention that the method branch is the guiding branch of the criticality analysis, constantly interacting with the information branch and triggering the scenario branch when needed. As such, most of the effort for developing the criticality analysis in VVM was invested in the method branch. Consequently, this led to a variety of changes to its process steps, although the key concept of identifying criticality phenomena and analyzing their causal relations remains unchanged. The most significant change in structure is the division of the method branch into an *associative analysis* part and a *causal analysis* part. Implicitly, this distinction was already present in the concept paper, cf. [1, Figure 4]. The distinction was made explicit when we put the causal analysis of CP on a formal mathematical basis using Pearl's causal theory, cf. [2, Figure 1]. Besides providing a solid structure to the method branch, distinguishing associational from causal process steps provides a classification of the respective artifacts.

2.1.1 Associative Analysis Originally, the associative analysis part consisted of only one process step, namely 'Identification of Criticality Phenomenon'. This step was initially conceptualized in [1, Section V.A.1] and exemplarily carried out for two data sources: i) using an urban drone data set [3] and ii) analyzing the GIDAS accident database [4]. Lessons learned from writing these papers led to the following adaptions: the process step is now called 'Identification and Formalization of Criticality Phenomenon', as formalization is required before CP can be algorithmically identified – in any source of data. Moreover, the former is followed by a newly introduced process step 'Estimation of Criticality Association', which essentially implements, the *criticality property* of [1, Definition 3]. More specifically, in order to estimate the associational relevance of CP we require the link to (validated) criticality metrics that make the CP's effect visible [5]. Together with a suitable threshold value we can decide whether the criticality association is sufficient to continue with the causal analysis of the CP at hand.

Further but rather cosmetic changes to the associative part include adding a scenario branch 'trigger' to 'Estimation of Criticality Association' to clarify the connection between the branches here, and adding 'discardment of CP' as an option when there is no sufficient criticality association, together with a potential 'exit' option after CP-discardment. Also note that the double iteration loop of [1, Figure 4] has been detangled into one iteration loop for each, associative part and causal part, in the updated procedure, cf. Figure 2.

2.1.2 Causal Analysis Similar to the associative analysis, the causal analysis part of the method branch received major updates. After the criticality association of a CP is deemed 'sufficient' and before starting with causal analysis, we included a check whether any existing causal relation already explains the CP under consideration, e.g. due to being an abstraction or concretization of another CP that already underwent causal analysis. If there already exists such a causal relation we can either skip the causal analysis entirely ('exit') or, at least, reuse the knowledge that is already available for that related CP.

The following initial process step of the causal analysis has been renamed from 'Proposal of Hypothesis over Causal Relation' to 'Modeling of Causal Relation'. As explained later on in Section 3.2, we refined the definition of a *causal relation* based on the framework of causal theory, cf. [2]. While the causal relation still encodes the causal assumptions of what influences the CP, how the CP increases criticality, and in what context, the expression 'hypothesis over causal relation' is not needed anymore. Hence, the corresponding hypothesis artifact H_i has been removed as well.

The next step, originally called 'Plausibilization', was renamed to 'Plausibilization of Causal Relation' and, due to the need for evaluation of causal effects within a given context, this step also received a 'trigger' for the scenario branch. The following decision whether the evidence for the causal relation is good enough, previously labeled 'E_i sufficient for H_i?', now evaluates whether the causal explanation is sufficient based on appropriate quantities for assessing the modeling quality [2, Section 3.3]. Notably, compared to [1, Figure 4], the causal analysis now has its own plausibilization loop called 'Improvement or Discardment of Causal Relation' where at this stage in the method branch discardment does not mean that the CP at hand can be discarded altogether, but rather that the modeling of its causal relation requires broad modifications.

2.2 Updates to the Information Branch

Although much has been achieved in the realization of the information branch, as described in Section 4, there were only minor changes regarding the procedure of Figure 2, compared to the original concept [1, Figure 4]: First, we slightly reworked the examples illustrating the process step 'Data Analysis' by replacing 'other' with 'causal inference', as the causal analysis part of the criticality analysis was consolidated [2]. Regarding 'Knowledge Acquisition' we removed 'other' and added 'Laws of Nature' and changed 'expert analysis results' to 'literature analysis'. As to avoid repetition, in the 'information basis', we removed the corresponding bullet points completely. Concerning the list of contained artifacts in the knowledge basis, we removed 'Criticality Hypothesis' and 'Relational Evidences' in order to keep the flowchart comprehensible and not overloaded.

2.3 Updates to the Scenario Branch

Regarding the scenario branch, no significant changes have taken place. However, there were minor changes to the flowchart of Figure 2: we added 'triggers' to clarify when the method branch actually involves the scenario branch in a criticality analysis. This link was missing in earlier versions.

3 Advances within the Method Branch

Being the guiding branch of the criticality analysis, most of the effort was put into the method branch. Major advances were achieved regarding a) the identification of criticality phenomena in various sources (which also necessitates their formalization), cf. Section 3.1, and b) the modeling and analysis of their respective causal relations, cf. Section 3.2. Advances on the elicitation and evaluation of safety principles, cf. Section 3.3, were only theoretical and could not be conducted due to effort constraints.

3.1 Identification and Formalization of Criticality Phenomena

First, let us remark that the definition of a *criticality phenomenon*, similar to the definition of *criticality*, has not changed since its introduction, cf. [1, Definition 2, Remark 2]:

Definition 2 (Criticality Phenomenon). *A criticality phenomenon CP is a concrete influencing factor in a scenario (or a combination thereof) which is associated with increased criticality.*

Remark 1 (Criticality Phenomena). Criticality phenomena therefore represent classes of danger.

The process step 'Identification of Criticality Phenomenon' was initially described as a 4-step procedure by Neurohr et al. [1, Section V.A.1] where it has been instantiated for the CP OCCLUSION and several of its concretizations such as OCCLUDED PEDESTRIAN or OCCLUDED VEHICLE. In a more recent publication, this 4-step procedure has been slightly revised, cf. [4, Section 2.1]. In the course of the VVM project, much effort and time was invested to carry out this process step more broadly. In particular, the criticality analysis was conducted for a quite generic class of ADSs at SAE Level 4/5 in urban environments as operational domain [1]. As suggested by Damm and Galbas [6], the process step 'Identification and Formalization of Criticality Phenomenon' has been divided into two parts, namely

(i) CP that are relevant to human traffic, e.g. not specifically dependent on perception of the environment through a sensor setup; and
(ii) CP that become relevant for ADSs that rely on sensor technology for perception.'

As to give structure to this distinction, the concepts of 'perfect perception' and 'real perception' were introduced.

3.1.1 Criticality Phenomena and Perception

Root causes for criticality in highly automated driving use cases can often be identified as insufficient performance of the perception subsystem. This applies to humans as well as a machine performing a driving task.

The perception subsystem has the task of interpreting a complex scene from sensor information and build a world model as representation of the real world. For highly automated driving the most frequently used sensor technologies for this task are camera, Radar and Lidar. For interpretation of the sensor information complex algorithms or artificial intelligence is required.

In order to systematically analyze the criticality due to perception, we introduce an abstraction concept ranging from perfect perception, perception with perfect technology to perception with real technology. These three abstraction layers are defined as follows:

Definition 3. *The **perfect perception** is an error-free observation of all for a given task relevant objects of the world and their properties.*

Definition 4. *The **perception with perfect technology** is an error-free observation of objects and their properties within defined constraints of an idealized technology.*

Definition 5. *The* ***perception with real technology*** *is an error-prone observation of objects and their properties with an implementation of a chosen technology.*

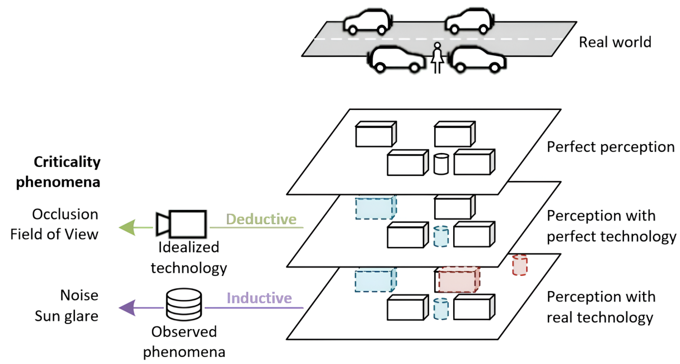


Fig. 3: Abstraction approach for perception: *perfect perception* as error-free all-seeing perspective, *perception with perfect technology* as error-free observation with idealized technology, and *perception with real technology* as error-prone observation based on a concrete implementation.

The perfect perception is based on the philosophical idea of Laplacian determinism [7]. The french mathematician Laplace describes a hypothetical entity (referred to as *Laplace demon*) that knows the complete state of the world and by applying a set of physical rules is able to accurately predict the future. With perfect perception we refer to the ability to perceive the complete state of the world analogous to the Laplace demon. However, we restrict ourselves to the states relevant for a given driving task. The intention of the definition of perfect perception is to provide a reference for ground truth.

By the choice of a perception technology, i.e. sensor and perception algorithm, we constrict ourselves to only a partial observation of the world. For example, with a camera it is only possible to capture incident light in the visible regime on a point of observation within a given field of view. It is not possible to observe the complete state of the world, but rather only a part of it. We define the perception with perfect technology as this part of the observable world based on well-defined constraints of the technology (e.g. wavelength, field of view, etc.). But these observations are free from errors, i.e. no deviation from the defined constraints. The constraints are not goverened by physical limitations (e.g. quantum mechanics, refraction limit, etc.) but rather serve as an idealized version of a technology. By defining these constraints we are able to deductively analyze the criticality emerging in a scenario under the assumption of perception with perfect technology. This enables the investigation of scenarios early in the development lifecycle without the need of an actual implementation.

Lastly the perception with real technology incorporates the phenomena due to a concrete implementation of a technology. It includes the errors arising due to faults in the implementation which will be encountered in the real world. These can be classical E/E faults (e.g. pixel errors or noise) but also more complex faults fromt the SOTIF domain (e.g. missclassification of objects due to incomplete dataset or sun glare leading to overexposure). The phenomena due to perception with real technology can only be characterized for a specific implementation and require an inductive approach by generalizing observed phenomena. For analysis the causal relations method as discussed in Section 3.2 can be applied.

3.1.1.1 Perfect/Real Sensors Most frequently used sensors for automated driving tasks are camera, radar, and lidar. For example, the perfect camera can be defined as capturing light in a well-defined wavelength range with ideal cut-off. This leads to an acausal system response, but as mentioned before the definition of perfect technology is free from physical constraints. Further, the field of view has a well-defined opening angle and range. Again with an ideal cut-off and free of fringe effects. The resolution is infinite which eliminates effects caused by the discretization into pixels.

3.1.1.2 Perfect/Real Perception Algorithm The concept not only applies on the sensor part of perception, but also the interpretation by an algorithm or artificial intelligence. Providing an abstraction for a perfect perception algorithm is more intricate than for sensors as it deals with more abstract concepts rather then quantities grounded in a physical world. Further, the task of perception algorithms can differ significantly (e.g. pedestrian detection with bounding boxes vs. semantic segmentation) for various tasks. Therefore, we propose a generalized definition: A perfect perception algorithm interprets sensor information free of errors into a pre-defined ontology when sufficient sensor information (i.e. exceding a perception threshold) is provided.

3.1.2 Identification of Criticality Phenomena within the GIDAS Accident Database Babisch et al. provide a blueprint for the identification of CP in accident databases and instantiate this blueprint for the extensive GIDAS database [4]. Following the identification of 166 candidate CP for automated driving, judged to be relevant for human traffic, i.e. corresponding to part (i) of this process step [4, Sect. 3.1], they analyze the GIDAS database regarding the presence of these CP in a subset of accident cases. The subset of accidents was chosen to reflect the operational domain (OD) of the VVM project, i.e. *urban scenarios with passenger car involvement* [4, Sect. 3.2], amounting to $n = 15\,417$ accident cases within the OD. For comparison, in total, there are 38 571 documented and reconstructed accidents in GIDAS, cf. [4, Figure 2].

As to search the accident cases in GIDAS, the identified candidate CP were translated to the database scheme. At this point, no formal representation of the CP was available. Therefore, Babisch et al. skipped the formalization step and

Table 1: An excerpt of the 15 417×116 case-phenomenon relation matrix together with an anonymized GIDAS case number, weighting factor, extrapolation factor, and accident severity for each case. A value of **1** means that the corresponding criticality phenomenon was present in the respective GIDAS accident case, while a value of 0 indicates its absence.

GIDAS case number	Weighting factor	Extrapolation factor	Severity	Criticality Phenomenon ID								
				#17	#19	#20	#21	#22	#23	#24	#25	#26
47415	1.089	13.433	1	0	0	0	0	0	0	0	0	0
24580	0.362	4.472	2	**1**	0	0	0	0	0	0	0	0
32474	0.743	9.171	1	**1**	0	**1**	0	0	0	0	0	0
45433	0.468	5.777	2	**1**	**1**	0	0	0	0	0	0	0
40614	1.089	13.433	1	0	0	0	0	0	0	0	**1**	**1**
93567	0.743	9.171	1	**1**	0	**1**	0	0	0	0	0	**1**

directly translated the CP to GIDAS-compatible SQL queries, albeit clearly stating the need for such a formalization step in general [4, Section 3.3]. The process for the formalization of CP has been addressed explicitly, cf. Section 3.1.3.

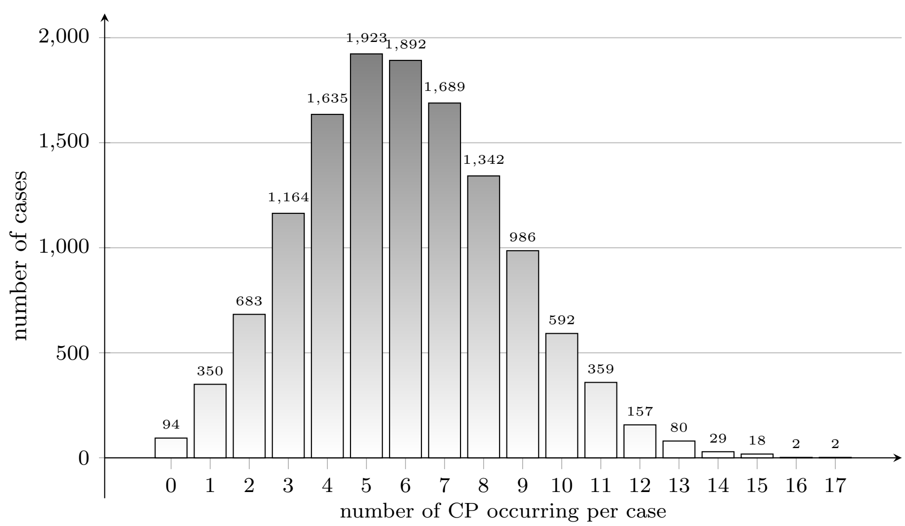


Fig. 4: Plot of the distribution of accident cases in the operational domain along the number of criticality phenomena occurring per accident case.

In total, 116 of the initial 166 CP could be identified within the GIDAS database, leading to a 15417 × 116 binary matrix, named *case-phenomenon relation matrix*, cf. Table 1 for an example or the supplementary material for the complete matrix [8]. The distribution of the number of CP per accident case is depicted by Figure 4. This statistic has been adjusted to exclude common abstraction-concretization relations in order to avoid double-counting of CP.

Table 2: Frequentist quantities for three example CP. Absolute and relative frequencies describe their occurrence in the OD, while the projection provides an estimate for the total German traffic accident statistics in 2019.

ID	**Criticality Phenomenon**	**Absolute Frequency**	**Relative Frequency**	**Projected Frequency**
#17	Intersecting Planned Trajectories of TPs	7 156	55.1%	88 305
#31	Non-Ego-TP violating Right of Way	2 644	20.3%	32 628
#131	Occlusion	2 978	22.9%	36 746

From the case-phenomenon relation matrix absolute and relative frequencies, as well as the projections to the national German accident statistic are easily calculated using the formulas of [4, Equation (4)]. Table 2 shows the resulting values for three example CP.

3.1.2.1 Bayesian Approach for the Assessment of Risk associated with Criticality Phenomena The main contribution of Babisch et al. is the estimation of CP-associated human risk based on the following formula, cf. [4, Equation (1)]:

$$\begin{aligned} &\mathit{Risk}_{\mathit{Human}}(\mathrm{CP}, \mathit{Accident}, \mathit{Severity} \mid \mathrm{OD}) \\ &= \underbrace{P(\mathrm{CP} \mid \mathrm{OD})}_{\text{Exposure}} \underbrace{P(\mathit{Accident} \mid \mathrm{CP}, \mathrm{OD})}_{\text{Controllability}} \underbrace{P(\mathit{Severity} \mid \mathit{Accident}, \mathrm{CP}, \mathrm{OD})}_{\text{Severity}} \\ &= P(\mathrm{CP} \mid \mathit{Accident}, \mathrm{OD}) P(\mathit{Accident} \mid \mathrm{OD}) P(\mathit{Severity} \mid \mathit{Accident}, \mathrm{CP}, \mathrm{OD}), \end{aligned} \tag{1}$$

where OD corresponds to *urban areas in Germany* and *accident* refers to *accidents with passenger car involvement and damage to persons*. Hence, the unit of resulting risk value is accidents (of a given severity) per 10^9 kilometers driven.

For the exact details of how the quantities of equation (1) are estimated, we refer to [4, Section 4.3]. A generalization to conjunctions of CP is also provided, cf. [4, Section 4.4]. Table 3 shows the results of the (human) risk estimation for three example CP and four conjunctions thereof. Such values can be used as reference values for the right hand side of the inequality $\mathit{Risk}_{ADS} < \mathit{Risk}_{Human}$ for a positive risk balance, cf. [9,10], within the abstract scenario classes defined by (conjunctions of) CP. A risk model around this inequality is provided by Salem et al. [11], while Putze et al. sketch first ideas to evaluate its left hand side [12].

Taking into account accidents of arbitrary severity leads to an interesting ranking of the CP, cf. Figure 5, which essentially corresponds to their occurrences

Table 3: The estimated probability of severity and the associated human risk for three criticality phenomena and three combinations thereof. A severity level of $s = 2$ refers to accidents with at least *serious injury* and $s = 3$ refers to accidents with *fatal injury*.

Criticality Phenomenon	**Severity** with $s=2$	with $s=3$	$\mathbf{Risk_{Human}}$ with $s=2$	with $s=3$
INTERSECTING PLANNED TRAJECTORIES OF TPS (#17)	15.43%	0.30%	70.3	1.4
NON-EGO-TP VIOLATING RIGHT OF WAY (#31)	17.94%	0.52%	30.2	0.9
OCCLUSION (#131)	17.20%	0.21%	32.6	0.4
$\mathrm{CP}_{17} \wedge \mathrm{CP}_{31}$	18.74%	0.51%	28.2	0.8
$\mathrm{CP}_{31} \wedge \mathrm{CP}_{131}$	20.89%	0.48%	15.4	0.4
$\mathrm{CP}_{17} \wedge \mathrm{CP}_{131}$	18.17%	0.23%	27.6	0.3
$\mathrm{CP}_{17} \wedge \mathrm{CP}_{31} \wedge \mathrm{CP}_{131}$	21.42%	0.44%	14.7	0.3

in the OD. Such a ranking can e.g. be used within the criticality analysis for prioritizing CP for causal analysis. Moreover, these values instantiate the process step 'Estimation of Criticality Association', cf. Figure 2.

The final analysis performed by Babisch et al. considers pairwise associations between the CP, calculated using the Φ-coefficient. Figure 6 shows the $\binom{3}{2} = 3$ resulting values for three CP. For the complete list of $\binom{116}{2} = 6670$ Φ-coefficients we again refer to the supplementary material [8].

3.1.3 Process for the Formalization of Criticality Phenomena Based on the previously described issue of requiring a consistent, unified semantics of CP, Wellßow delineates a process starting from the identification up to the formalization of such safety-critical influencing factors [13].

Here, the main idea is to use expert knowledge and data-driven approaches to recognize CP in the first place. This includes guided brainstorming approaches, analysis of accident data bases, examination of video recordings of critical scenarios and driving school catalogs. Afterwards, CP have to be described using natural language. Specifically, using a controlled language (e.g. by having a fixed vocabulary) can be helpful. Wellßow proposes a formal grammar based approach to give structure to such a description. This semi-formal structure forms the basis for the subsequent formalization. For this, ontologies in the form of Description Logics (DLs) form the basis. However, DLs are not capable of representing temporal facts well. Therefore, various temporal logics can be used on top of DLs, e.g. Allen's calculus. Moreover, spatial calculi (such as RCC8) can support describing and inferring spatial facts within a given scene. This formalization can then be used for various analyses, for example, to derive whether one CP is an

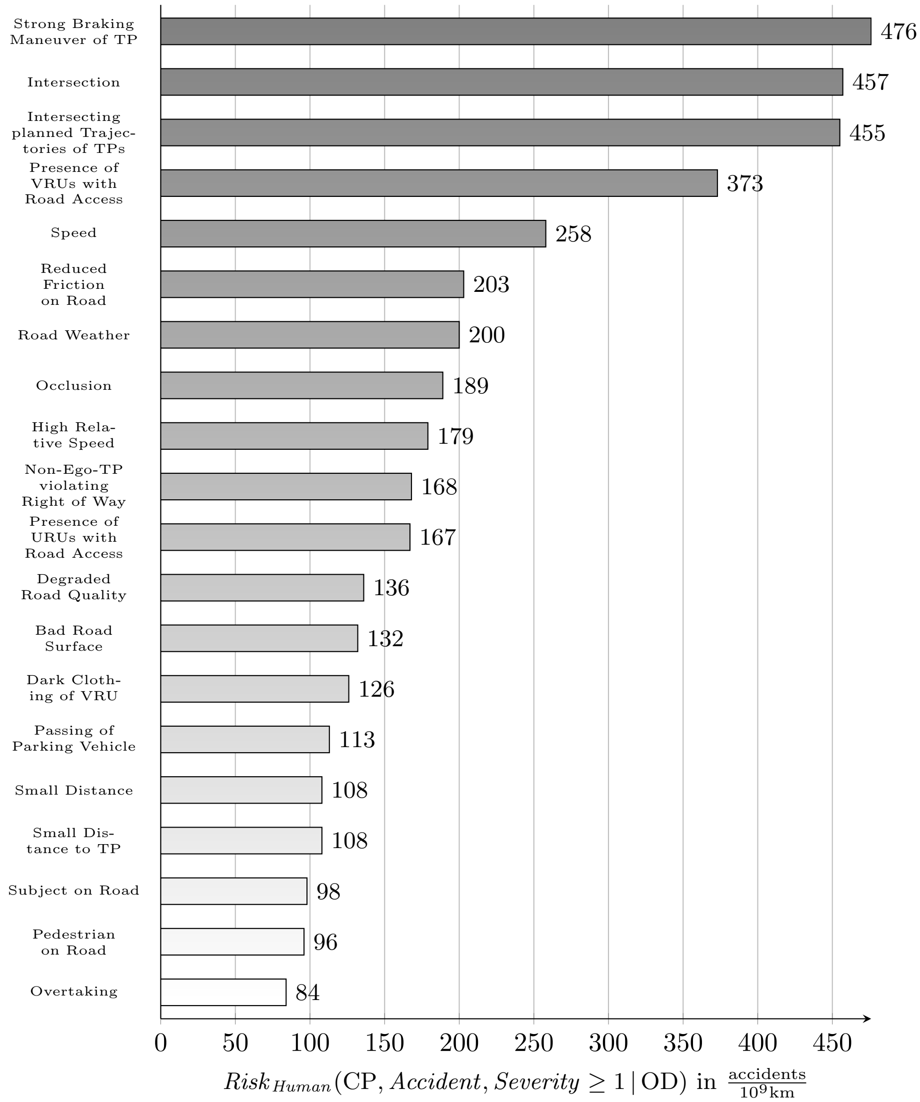


Fig. 5: Top twenty criticality phenomena ordered according to the quantity $Risk_{Human}(\mathrm{CP}, Accident, Severity \geq 1 \mid \mathrm{OD})$, estimated using [4, Eq. 11], with unit *accidents with passenger car involvement and damage to persons per billion kilometers.*

abstraction (or a concretization) of another. This is especially helpful when the catalog of CP grows and their interrelations become non-obvious.

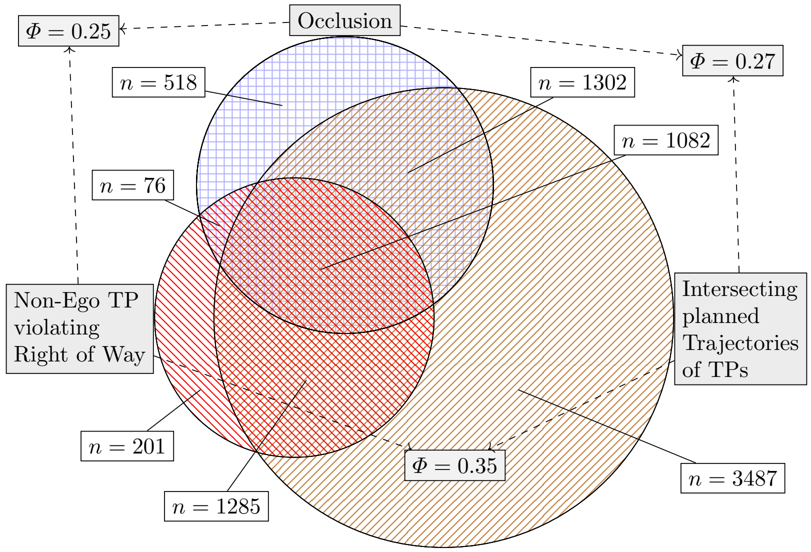


Fig. 6: Proportional Venn diagram showing the absolute frequencies and the Φ-coefficient in the operational domain for the pairwise combinations of three criticality phenomena.

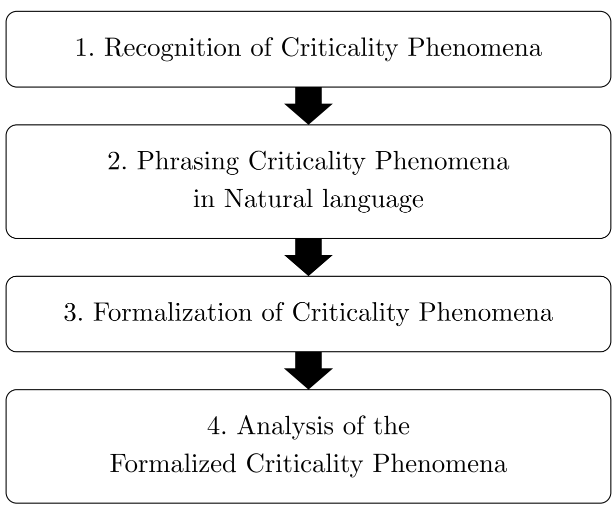


Fig. 7: Process for the formalization of criticality phenomena according to Wellßow [13].

One artifact that is created in this process is the *database of criticality phenomena*, which is an artifact of the knowledge base and is thus described in

Section 4.2.1. The next section will show, in detail, how CP formalized in DL can be used to analyze data regarding their presence in that data.

3.1.4 Ontology-Based Analysis of Criticality Phenomena Westhofen et al. instantiate this process by detailing how CP can be uncovered in data using a formalization on an ontological basis [3]. For this, a suitable ontology is assumed to be given, which is then refined iteratively. The resulting ontology that originated in the context of the criticality analysis in the VVM project, called *Automotive Urban Traffic Ontology* (A.U.T.O.), is described in Section 4.2.2.

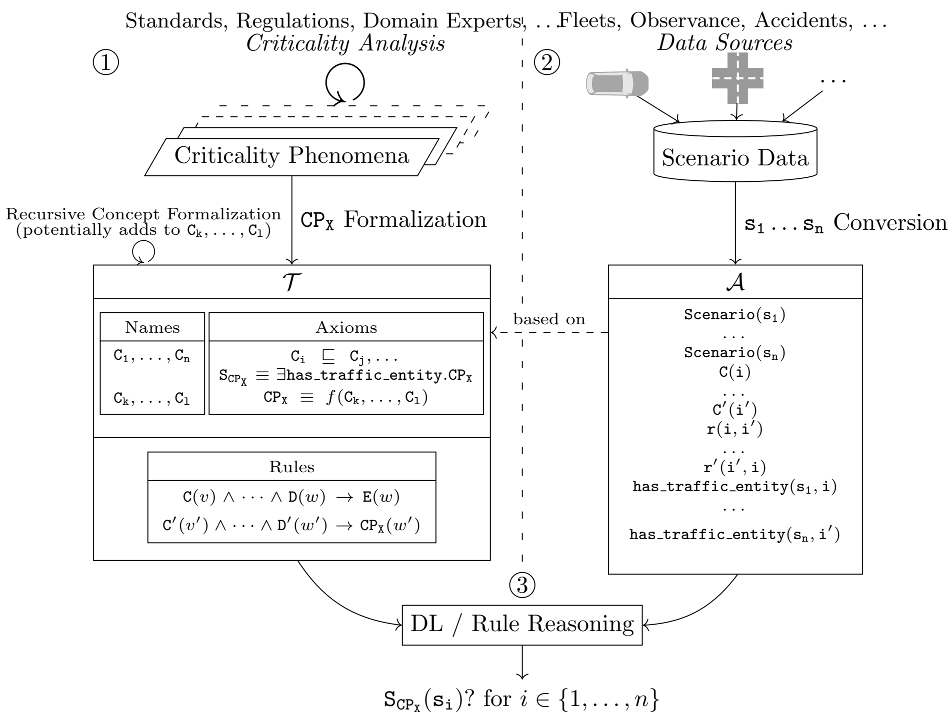


Fig. 8: A process for ontology-based analysis of criticality phenomena according to Westhofen et al. [3].

Here, the idea is to assume a natural language description of a set of CP, i.e. step 2 of Figure 7 has been performed. Then, the underlying ontology is used to formalize (aspects) of these CP, while iteratively refining this ontology with missing concepts and axioms. For example, the CP ILLEGITIMATE USE OF PEDESTRIAN CROSSING BY BICYCLIST relies on 'bicyclist' and 'pedestrian crossing' as ontological concepts. Moreover, rules such as $\texttt{Rain}(?r) \wedge \texttt{has\_intensity}(?r, ?i) \wedge ?i > 50 \rightarrow \texttt{Heavy\_Rain}(?r)$ can enrich the formalization.

For the final analysis, scenario data are converted into the ABox of the ontology. A DL reasoner can then infer new facts based on the modeled background knowledge, e.g. whether a vulnerable road user is present.

Westhofen et al. show how this process can be instantiated using the Pellet reasoner and the inD data set [14], exemplarily uncovering a fixed set of CP on the scenarios modeled in the OMEGA-format [15]. Based on the presence of CP, various analyses can be performed such as the examination of temporal dependencies, which can serve as an initial hint towards causal relations.

In general, such a unified model of CP can also be used in all downstream method steps, including the creation of causal relations. In particular, ontologies can be used to specify the *context* of such causal relations, cf. Definition 6. Moreover, when performing data analyses, e.g. assessing effect sizes of causal relations, the previously described process can be used to ensure a consistent semantics across different data sources.

3.2 Modeling and Analysis of Causal Relations

The concept paper of the criticality analysis merely sketched a process for proposing hypotheses about causal relations and their subsequent plausilibization, cf. [1]. Naturally, the need arose, to put the causal analysis of CP on a framework that is formally grounded. This need motivated the work of Koopmann et al. [2]. They outline four types of causal queries that need to be investigated within a criticality analysis, cf. [2, Section 2.1]:

Q1 The explanation of a CP by a set of predecessors in a causal relation
Q2 The causal effect of a CP on criticality as measured by a suitable metric
Q3 The explanation of measured criticality by a CP
Q4 The causal effect of safety principles on reducing measured criticality

In order to perform a systematic investigation of these causal queries Koopmann et al. build on the framework of causal theory introduced by Pearl, cf. [16]. This framework enables an investigation of causalities based on observational data by combining graphical modeling of causal assumptions with Bayesian stochastics. The main idea is that on a certain level of detail causalities define deterministic functional relationships. While the concrete functions are typically unknown, the causal relationships can be represented by directed acyclic graphs (DAGs), called causal structures, consisting of a set of exogenous and endogenous variables $V \cup U$ and a set of edges E between the variables representing the potential relations. These causal structures can then be instantiated with data. Pearl's do-calculus defines a mathematical language to express causalities in form of hypothetical interventions based on the causal structures. It provides requirements as well as concrete formulas to estimate post-interventional distributions based on the pre-interventional data.

As to apply causal theory within a criticality analysis Koopmann et al. formalize the CP under investigation in a way that it can be expressed as a value of a binary random variable X with $Im(X) = \{cp, \neg cp\}$. Further, a suitable criticality metric φ needs to be chosen, which is able to uncover the CP's influence on criticality. Based on this, the emergence of the CP and its influence on the criticality metric can be modeled in a causal structure.

However, such a causal structure can be arbitrarily large and complex, especially if it has to cover the whole scenario class in which the CP is present. Therefore, Koopmann et al. propose to derive different sub classes that are manageable by defining additional constraints and assumptions under which the causal structure can be applied – the *context of a causal structure*, cf. [2, Definition 1]:

Definition 6. *The **context of a causal structure** is a set of statements about the existence of and constraints on the individuals contained in a suitable ontology.*

A simple example of a context structured along the 6-Layer-Model, cf. Figure 16, is given by Table 4 for the CP REDUCED COEFFICIENT OF FRICTION. Restricting a CP's causal structures to a certain context leads to the following formalized definition of causal relations, cf. [2, Definition 2]:

Definition 7. *A **causal relation** for a criticality phenomenon CP is a tuple $(\mathfrak{S}, \mathfrak{C})$ with respect to a suitable traffic domain ontology $\mathcal{O}$ where $\mathfrak{S} = (V \cup U, E)$ is a causal structure and $\mathfrak{C}$ is a context such that there exists*

(i) a binary random variable X with $\mathrm{Im}(X) = \{cp, \neg cp\}$ corresponding to a node in V and
(ii) a criticality metric $\varphi : \mathcal{S} \to [0, \infty)$ corresponding to a sink (i.e., no outcoming edges) in V,
(iii) where the variables in U are the exogenous variables corresponding to error terms,
(iv) the variables in V are defined on properties of the individuals in $\mathcal{O}$, and
(v) the context $\mathfrak{C}$ is defined as in Definition 6 with respect to $\mathcal{O}$.

Definition 7 can be seen as an update to the original definition, cf. [1, Definition 7], providing a better formal basis for causal relations as important artifacts of a criticality analysis. The causal structure of Figure 9 together with the context of Table 4 give a detailed example of a causal relation for the CP REDUCED COEFFICIENT OF FRICTION, modeling the emergence of the CP, as well as its influence on a generic criticality metric $\mathrm{agg}(\mathrm{BTN}_{\mathrm{DT}}, \mathrm{STN}_{\mathrm{DT}})$ - an aggregate of the driving task induces Brake resp. Steer Thread Number ($\mathrm{BTN}_{\mathrm{DT}}$ resp. $\mathrm{STN}_{\mathrm{DT}}$), cf. [2, Section 4].

Causal relations serve as a basis for the subsequent analysis of the causal queries (**Q1**-**Q4**). In a first step, the causal relation needs to be plausibilized which includes an investigation whether the emergence of the CP (**Q1**) and the measured criticality (**Q3**) are explained sufficiently by the causal relation. Koopmann et al. introduce various causality indicator functions that can serve to assess the modeling quality, cf. [2, Definitions 5-7]. However, an in-depth analysis of modeling quality measures is considered future research. If a causal relation is deemed to be stable, i.e. iterated and plausbilized, it can be used to assess the influence of the CP on criticality (**Q2**), for example by estimating the average and relative causal effects.

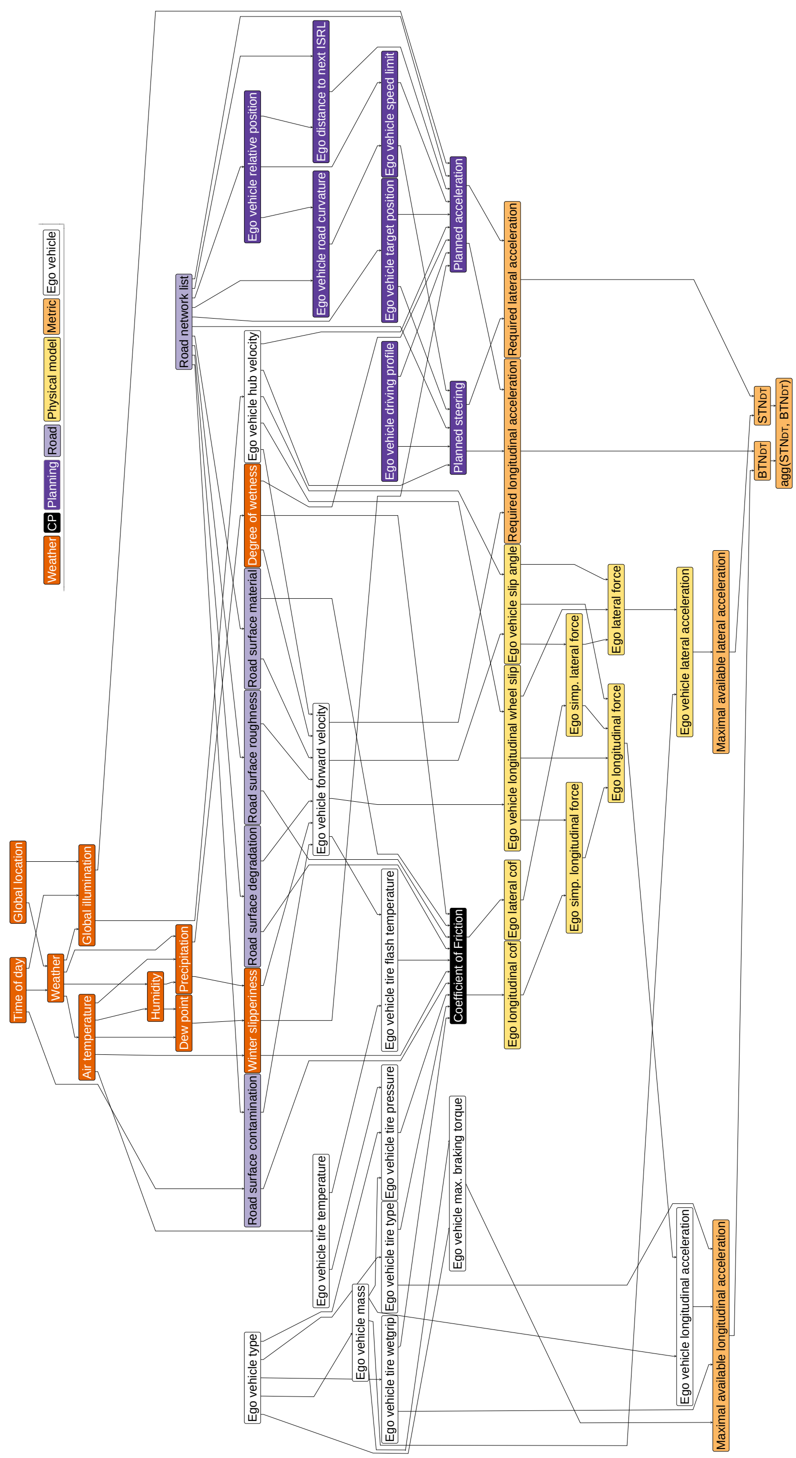


Fig. 9: Causal structure for the criticality phenomenon REDUCED COEFFICIENT OF FRICTION, cf. [2, Figure 6].

Table 4: Example context for the causal relation of the criticality phenomenon Reduced Coefficient of Friction, cf. [2, Table 1].

Layer	**Property**
(L1) Road Network and Traffic Guidance Objects	A road network shall exist and shall consist of either a curved road or a junction.
(L2) Roadside Structures	Roadside structures may exist and are not further constrained.
(L3) Temporary Modifications of (L1) and (L2)	There shall be no temporary modifications to layer 1 and 2.
(L4) Dynamic Objects	An ego vehicle shall exist. No other vehicle relevant to the ego vehicles actions and behavior shall exist.
(L5) Environmental Conditions	Environmental conditions shall exist and remain unconstrained.
(L6) Digital Information	Digital information might exist, but shall remain unconstrained.

3.3 Derivation and Evaluation of Safety Principles

The method branch of the criticality analysis, as portrayed by Figure 2, focuses on safety-critical influencing factors associated with *increased* criticality and discovery of their underlying causal relations. However, after a CPs causal relation has been iteratively plausibilized and causal effects have been calculated, one might be interested in teaching ADSs so-called *safety principles* that either reduce the probability of encountering the CP, or avoid or, at least, mitigate the causal effect of the CP on criticality. These can also be used as a blueprint for safety goals in a system-dependent hazard analysis and risk assessment.

Koopmann et al. extended the criticality analysis to cover such safety principles, albeit only on a methodical level, cf. Figure 10 (taken from [2, Figure 1]).

The key idea (for future work) is to use the plausibilized causal relation for a CP to derive and evaluate safety principles that reduce/mitigate the criticality induced by a CP. Here, various methods are conceivable, for example, modeling them as (stochastic) interventions or slightly extending the causal structure by modeling them as exogenous variables. Based on such a formal model, the effectiveness of a safety principle can then be *formally proven*, assuming plausibility of the causal relation. An engineer can then select a suitable set of safety principles, which provides enough effectiveness for ensuring safety, but also results in a well-performing behavior in traffic.

Besides the open issue of modeling causal relations formally, such relations also help us to identify possible safety principles through their structure. For this, we image safety principles to be divided into two main categories.

The first are those that reduce the probability of the CP by an intervention on its predecessors in the causal relation. Prominent examples of this category are ODD exclusions or taking alternative routes to avoid certain situations such as construction sites.

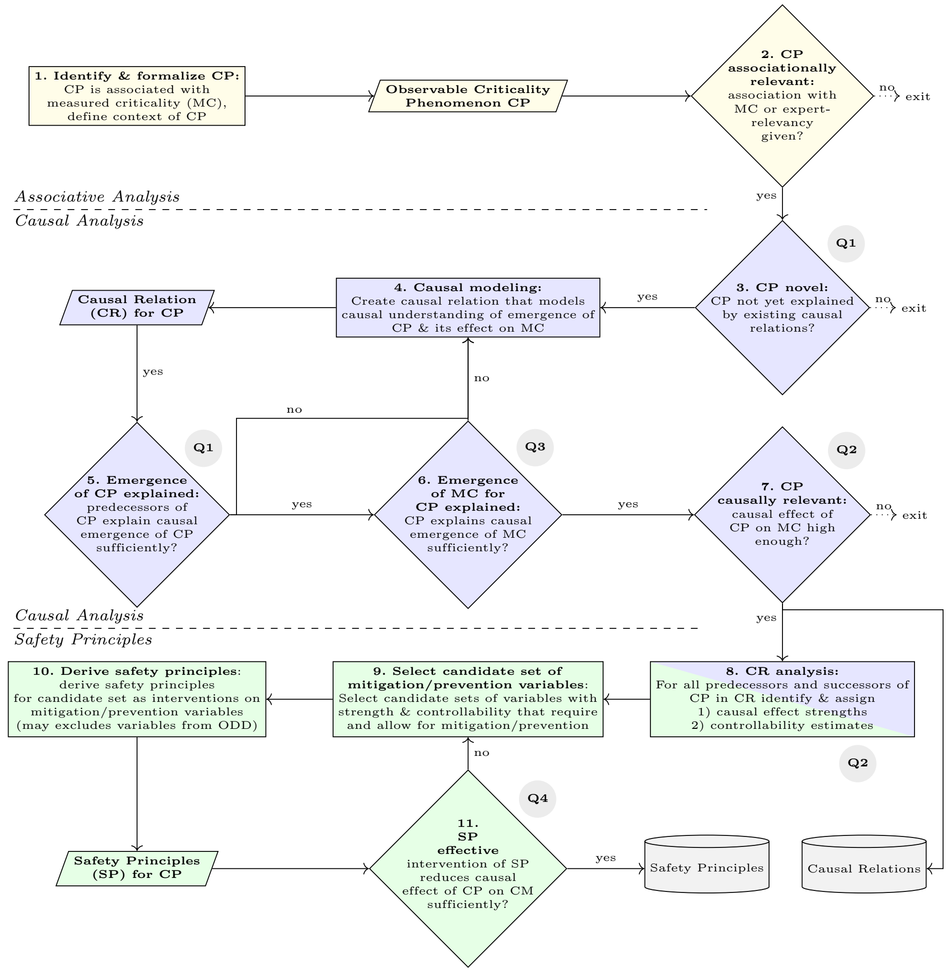


Fig. 10: The methodical part of the criticality analysis for unveiling causalities behind criticality phenomena, extended by the derivation and evaluation of safety principles. Relevant steps are annotated with their respective causal queries **Q1** to **Q4**, cf. Section 3.2.

The second category includes those safety principles that reduce the effect on criticality once the CP is encountered by intervening on one of the successors of CP or the predecessors of criticality. The first case – intervention on the successors of CP –, are preemptive measures based on anticipation. This includes behavioral changes, e.g. adapting the planned path if a child is playing close to the driving lane. In the second case – intervention on the predecessors of criticality –, the safety principle is rather independent of the CP and can be applied universally. This includes safety principles like 'reduce speed' or 'keep sufficient distance to front'.

A general theme of both classes is that there are always certain vertices in the causal relation *inducing* the criticality (often, in combination with other nodes) is the given context. These vertices can be addressed differently according to their underlying mechanism. This includes the following means:

1. Behavioral adaptations (e.g. driving slowly), or
2. Communication of planned behavior or possibly unknown situations to other traffic participants (e.g. turn signals, V2X communication, ...), or
3. Improving technical capabilities of the system (e.g. increasing performance of sensors, adding computational resources for longer prediction horizons in the planner, ...), or
4. Structural changes in the traffic system (e.g. adaptation of traffic rules, adding new infrastructure to intersections, ...), or
5. exluding certain factors from the ODD.

Such a classification can aid system designers and engineers in deriving a variety of safety principles.

4 Advances within the Information Branch

Advances in the information branch have focused on two major and one minor area. First, criticality metrics were examined. Second, a method and prototype for the formalization and recognition of CP was developed. Both can be used on data: criticality metrics can uncover scenarios with high criticality, in which again the presence of CP – factors potentially causing increases in measured criticality – can be recognized. One possibility is the use of real-world data, e.g. collected by sensor-equipped measurement vehicles. In this regard, the results of a criticality analysis can be used to guide real-world data collection drives towards relevant scenarios, for example to increase the chances of encountering certain CP.

4.1 Criticality Metrics for Automated Driving

Criticality metrics are generally used to measure criticality (cf. Section 3) on a predetermined scale of measurement $\mathcal{O}$, e.g. $\mathcal{O} \subseteq \mathbb{R} \cup \{-\infty, +\infty\}$. They can measure a given scene $S \in \mathcal{S}$ or scenario $Sc : \mathbb{R} \to \mathcal{S}$, i.e. they can be viewed as a function $\kappa : \mathcal{S} \to \mathcal{O}$ respectively $\kappa : \{\mathbb{R} \to \mathcal{S}\} \to \mathcal{O}$.

4.1.1 State of the Art Review on Criticality Metrics A criticality analysis' success rests upon valid criticality metrics, i.e. showing high sensitivity and high specificity. If criticality is measured invalidly, the drawn conclusions (e.g. which safety principles to implement on a behavioral or technical level) may not be sound. Westhofen et al. thus investigate the state of the art of criticality metrics extensively [5]. Here, the authors give a comprehensive list of criticality metrics and describe them using a unified mathematical notation as to enable comparability. Moreover, relevant properties for an exemplary list of applications (such as critical scenario identification) are extracted and examined for each identified criticality metric. Finally, Westhofen et al. describe how this analysis can be customized if presented with a specific application and its unique requirements on measuring criticality. The results are depicted at the corresponding website `https://purl.org/criticality-metrics`.

As an example, consider the well-known Time-To-Collision (TTC) metric, cf. [5, Section 5.2], defined as

$$TTC(A_1, A_2, t) = \min\left(\{\tilde{t} \geq 0 \mid d(p_1(t+\tilde{t}), p_2(t+\tilde{t})) = 0\} \cup \{\infty\}\right).$$

Whether the TTC correctly identifies all critical situations (i.e. its sensitivity) is highly dependent on whether the models p_1 and p_2 can predict a future collision at all (i.e. $d(p_1(t+\tilde{t}), p_2(t+\tilde{t})) = 0$). For example, a point-based kinematic prediction model has a greatly reduced chance of identifying such future collisions. Even if actors are represented using two or three dimensions, near-miss scenarios in the prediction model are not considered by the TTC. Due to this reason, criticality metrics such as the Worst-Time-To-Collision that effectively predict a set of future evolutions were developed, thereby increasing the sensitivity. However, such a set-based approach may suffer from a reduced specificity if some combination of predicted trajectories are unrealistic given the current circumstances. As can be seen from this example, devising a valid criticality metric can be a challenge. Moreover, finding valid target values to classify scenarios into critical and uncritical adds even more complexity: typical values for the TTC range between one and three seconds, i.e. an interval where the highest threshold is three times larger than the lowest one.

Due to this, their validation and calibration, as well as the development of improved metrics are imperative as a valid basis for the conduction of a criticality analysis.

4.1.2 Validation and Calibration of Criticality Metrics As motivated before, an important property of criticality metrics is their validity. In our context, we define validity as 'the closeness of the metrics' measurement to representing the actual accident probability and severity' [5]. In practice, this property is often broken down into specificity and sensitivity. These work by a comparison of the metric's classification results against a labeled data set and, thus, require ground truth data. We define *sensitivity* as the True Positive Rate (TPR), i.e. $TPR = \frac{TP}{TP+FN}$, where TP is the number of true positive and FN the number of false negatives. Similarly, *specificity* is defined using the True Negative Rate (TNR), i.e. $TNR = \frac{TN}{TN+FP}$, where TN is the number of true negatives and FP the number of false positives.

A promising way of analyzing sensitivity and specificity is to obtain the ground truth by using human assessments of scenarios. For example, one expert can label a given set of concrete scenarios as critical and uncritical [17]. If we then select a suitable target value (e.g. $-4.5m/s^2$) for the criticality metric (e.g. $a_{req,long}$), the expert-based classification can be compared against the one provided by the target value. This idea can also be extended by not relying on one expert for gathering ground truth data but conducting a large-scale study. In such a study, a larger sample of people can be asked to subjectively assess how critical a given scenario felt to them. In this way, ground truth data become more representative w.r.t. the overall population of traffic participants and the validity of the validity assessment itself increases.

Moreover, such expert or study data can not only be used to assess validity, but also to calibrate target values. We are specifically interested in choosing target values that optimize sensitivity and specificity. A graphical way of doing so is using Receiver-Operating-Characteristic (ROC) curves.

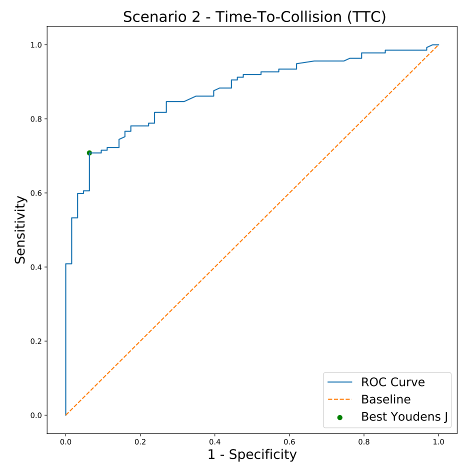


Fig. 11: Exemplary Receiver-Operating-Characteristic curve for the Time-To-Collision criticality metric.

Figure 11 shows an exemplary ROC curve for the TTC in a set of samples from a logical scenario [17, Section 4.2.2], where the target value was varied, leading from complete under-approximation of the critical scenarios (i.e., a sensitivity of 0 and a specificity of 1) to a complete over-approximation (i.e., a sensitivity of 1 and a specificity of 0). We can now identify the best target value by maximizing Youden's J, which is defined as $J(\tau) = TPR(\tau) + TNR(\tau) - 1$. For the example of Figure 11, the maximum Youden's J is located at a specificity of around 0.94 and a sensitivity of 0.71, leading to a target value of $\tau = 0.24$ [17, Appendix C].

In this way, criticality metrics can be calibrated in an empirical, data-driven manner which can be referenced in a rigorous safety argumentation.

4.1.3 Threat Metric Evaluation in Complex Urban Scenarios

One key item within the criticality analysis is the evaluation and interpretation of criticality metrics for traffic situations. Often these metrics are based on behavior prediction models of other traffic participants. Usually these models are computed with respect to lane information, which is often not defined explicitly. Schneider et al. presented a modeling and conformance testing approach for criticality metrics focused on urban scenarios, cf. [18]. Furthermore, they studied the influence of lane association together with the prediction models. One of the key results is that metrics constructed in such a way are generally prone to false positives as well as false negatives if maps with static lanes are used.

Hence, for the efficient and beneficial construction of criticality metrics in complex lane-based traffic spaces, a good notion of the lanes the other traffic participants are using is necessary.

4.1.4 Development of Novel Criticality Metrics Within the VVM subproject 'criticality analysis', various new criticality metrics were developed, adding to the plethora of already existing metrics. These were developed such that they are distinctly applicable to the characteristics of the considered scenarios (e.g. urban intersections) and are introduced in the following.

4.1.4.1 MerLin Here, we provide a brief overview of the *MerLin* criticality metric developed in VVM project. For a detailed description, we refer to [19]. MerLin defines the potential of each road user in accordance with a three-dimensional Gaussian curve, whose mean value lies in the center of gravity of the modeled road user and whose density is defined in both x and y directions. While the Gaussian distribution has significant densities inside the limits of the respective road user, the potential in the cross-section of the object is normalized to one on the boundary of the vehicles and limited to one inside the boundary.

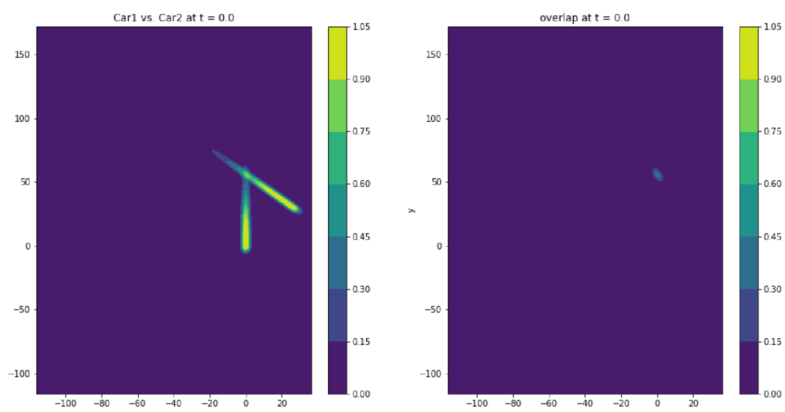


Fig. 12: Potentials used for *MerLin* in a intersecting trajectory scenario (left) and their convolution (right).

In the further course of the calculation this ensures that the criticality can never assume values above one. While previous approaches used the potentials and their overlap to determine a virtual force to solve the respective scenario, this approach uses the intersection volume of two or more potentials to obtain the criticality as a product. As to take the inertia of the movement into account, the potential in the direction of the movement is also exploited. With this approach, the positions of the respective road users and their velocity vectors at each point in time suffice as inputs for the criticality calculation according to MerLin, cf. Figure 12.

4.1.4.2 PrET and its Variants The Predictive Encroachment Time (PrET) is a predictive variant of the Post Encroachment Time (PET). The original purpose of the PrET was to evaluate scenario traces a posteriori in a criticality analysis. Naïvely, for this purpose, one could assume that the PET would suffice. However, the PET fails to detect close encounter scenarios in which one participant takes a rather long time to increase its speed after the close encounter. Due to this, the Neurohr et al. proposed the PrET, which predicts the PET using a dynamic motion model [1]. This enables to uncover critical situations in which a close encounter would happen if no one adapts its behavior (e.g. in case of a constant velocity model).

$$PrET(A_1, A_2, t) = \min(\{|\tilde{t}_1 - \tilde{t}_2| \mid p_1(t + \tilde{t}_1) = p_2(t + \tilde{t}_2),\ \tilde{t}_1, \tilde{t}_2 \geq 0\} \cup \{\infty\})$$

Here, $p_i(t)$ is a prediction model and computes the point that actor i reaches at time t. This works well for at most a few seconds prior to reaching the predicted conflict area. If the participants are both similarly far away (say, 20 seconds) from the conflict area, the PrET would be equal to zero. However, it is highly likely that for such long distances, actors adapt their behavior accordingly. Thus, the PrET is only valid temporally close to the conflict area. To mitigate this problem, Neurohr et al. introduced the Scaled PrET (SPrET) which basically penalizes long temporal distances of the actors [1]:

$$SPrET(A_1, A_2, t) = \min(\{|\tilde{t}_1^2 - \tilde{t}_2^2| \mid p_1(t + \tilde{t}_1) = p_2(t + \tilde{t}_2),\ \tilde{t}_1, \tilde{t}_2 \geq 0\} \cup \{\infty\})$$

Fehnker further developed the SPrET into the Duration-dependent PrET (DPreT) [17], which is defined as

$$DPrET(t) = \begin{cases} \max(1\text{s}, |\tilde{t}_1 - \tilde{t}_2|) \cdot \max(1\text{s}, \min(\tilde{t}_1, \tilde{t}_2)) & \text{if } \exists \tilde{t}_1, \tilde{t}_2 > 0 : p_a(t + \tilde{t}_1) = \\ & p_b(t + \tilde{t}_2) \wedge\ (|\tilde{t}_1 - \tilde{t}_2| \geq 1 \\ & \vee \min(\tilde{t}_1, \tilde{t}_2) \geq 1), \\ \max(|\tilde{t}_1 - \tilde{t}_2|, \min(\tilde{t}_1, \tilde{t}_2)) \cdot 1\text{s} & \text{if } \exists \tilde{t}_1, \tilde{t}_2 > 0 : p_a(t + \tilde{t}_1) = \\ & p_b(t + \tilde{t}_2) \wedge\ |\tilde{t}_1 - \tilde{t}_2| < 1 \\ & \wedge \min(\tilde{t}_1, \tilde{t}_2) < 1 \\ \infty & \text{else.} \end{cases} \tag{2}$$

Compared to the SPrET, the DPrET uses an analogous computation for cases where the PrET is ≥ 1. However, the DPrET uses $s = \min(\tilde{t}_1, \tilde{t}_2)$ instead of $\tilde{t}_1 + \tilde{t}_2$ as a scaling factor. Moreover, it catches the special case where both factors (PrET and s) fall below one. Then, the largest value of PrET and s is chosen.

Fehnker also used the DPrET as a component in the Conflict Index (CI), leading to a criticality metric named *Predictive Conflict Index* (PCI) [17].

4.1.4.3 Evasion Threat Metrics Starting point for the Evasion Threat Metrics (ETM) is the need for a criticality metric that relates the predicted outcome of a defined scenario to the criticality of the situation. Using the current state of the ego vehicle and another road user, families of trajectories are predicted and checked for collisions. Depending on the predictions, the following two metrics are combined:

- A first metric to compute if an accident is unavoidable, e.g. the minimum distance d_{min} between the ego vehicle and the other road user, predicted from the current state of motion, and,
- a second metric, computing the injury probability, in case the accident cannot be avoided.

As injury probability, the probability for the other road user being at least slightly injured is chosen below. The link to the injury probability is of special interest, since it enables the direct evaluation of accident severity.

For both cases, the trajectories of the ego vehicle and the other road user are independently predicted over a defined time horizon. The other road user is predicted by linear continuation of its velocity, i.e. using a constant velocity model. For the ego vehicle, a kinematic single-track model with zero side-slip angle is applied for reasons of simplicity. The ego prediction trajectories comprise pure braking, pure steering and combinations thereof. Braking and steering are limited by realistic actuator dynamics, comprising a dead-time and a linear ramp up to the stationary value, and by the friction coefficient.

With the predicted motion of ego vehicle and other road user, a collision check is subsequently performed using a bounding box approach. If no collision is found, the minimum distance d_{min} between the bounding boxes of ego and other road user is taken as criticality metric. In case of a collision, the predicted motion is evaluated yielding the collision velocity. The injury probability is calculated from the collision velocity and other collision variables, applying an injury risk function, which is taken in the below example for car vs. pedestrians.

Figure 13 shows an example using an automatic emergency braking situation according to Euro NCAP. On the left side a static diagram for one point in time with iso-lines for the minimum distance d_{min} is shown. The diagram is spanned by the target values of longitudinal and lateral acceleration. Kamm's circle depicted in red, limiting the accelerations w.r.t. friction, in this case lies safely above the maximum lateral acceleration, which can be achieved w.r.t. steer angle dynamics. Below that red line, the iso-lines for the minimum distance are found. The light yellow iso-line stands for a comfortable minimum distance of 2 m; whereas the dark blue iso-line represents $d_{min} = 0$, i.e. the limit to collisions. Collisions occur in the white, nose-shaped area bordered by the dark blue line. On the left side of the white nose, the vehicle stops in front of the pedestrian.

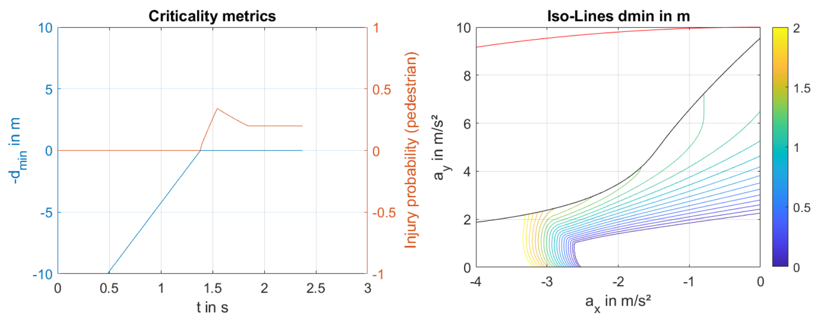


Fig. 13: Example for a static diagram (left) and a dynamic diagram (right) for the Evasion Threat Metrics.

Above the white nose, the vehicle passes the pedestrian laterally. On the right hand side of Figure 13 the evaluation of the ETM for this example is depicted. The blue line shows the negative value of d_{min}, continued in the orange line indicating the injury probablity for the involved pedestrian.

The ETM is used for filtering large data-sets using distance, velocity, and acceleration information to identify critical situations and/or sub-critical events.

4.2 Management of Criticality Phenomena in the Information Branch

Recall from Section 3.1 that the method branch of the criticality analysis hinges upon the central artifact of a *criticality phenomenon* (CP), cf. Definition 2. This yields two questions:

(1) how to practically manage a set of CP on an informal basis in early steps of a criticality analysis, and
(2) on which basis to formalize them once this becomes necessary in later phases.

We first show how a *Neo4j* database can be used for (1), and then proceed to describe the ontology used for their formalization in (2).

4.2.1 A Database of Criticality Phenomena The execution of the process step 'Identification and Formalization of Criticality Phenmenon' of the criticality analysis produces a vast amount of information that needs to be organized. This is where the method branch meets the information branch.

During part (i) of the process step, the CP have been kept in a simple spreadsheet. This spreadsheet, which was also used for the GIDAS analysis, described in Section 3.1.2, contained 116 CP and is available online [8].

Due to the limitations of such a spreadsheet, for conducting part (ii) within the VVM project, a graph database was created to save and organize the CP, their classification, properties and relations.

Based on the ontological model for the entity 'criticality phenomenon', cf. [1, Figure 13], a database scheme for the open source graph database *Neo4j* was created. Figure 14 shows a colorful overview over the CP in the database and the relations among them. An export of the CP database as OWL-file is publicly available[5].

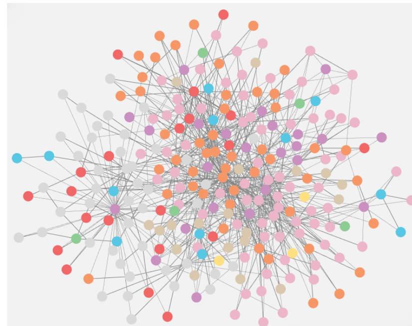


Fig. 14: Criticality phenomena and their relations in the Neo4j graph database.

4.2.1.1 Contents of the Database There are, represented as nodes, 299 CP in this database on varying levels of abstraction (effective 31.03.2022). Every node is assigned a combination of the following Node Labels

- Phenomenon (299)
- Ontology (an ontological classification of the criticality phenomenon based on 6-Layer-model [20] + Extensions in A.U.T.O. [3])
- Tags:
  - Behavior (75)
  - Complexity (11)
  - Dynamics (111)
  - Environment (32)
  - Human (35)
  - Perception (108)
  - Space (65)
  - Time (13)
  - Unpredictability (89)
  - Communication (28)
- and the attributes regarding the impact on perception technologies, i.e. implementation of concept 'perfect vs. real technology', cf. Section 3.1.1:
  - RealHuman (31)
  - PerfectHuman (16)
  - RealCamera (35)
  - PerfectCamera (32)
  - RealLidar (22)
  - PerfectLidar (28)
  - RealRadar (22)
  - PerfectRadar (21)
  - RealPerceptionAlgorithm (23)

4.2.1.2 Relations of Criticality Phenomena (in the Database) Besides saving CP in a structured manner and their classification according to various features, there exist many conceivable relations among the CP. The following relations are implemented in the database (effective 31.03.2022):

[5] https://github.com/lu-w/auto/blob/main/criticality_phenomena.owl

- abstraction_of (308),
- concretization_of (308),
- synergizes_with (283),
- could_lead_to (17)
- ontological classification (287)
- has_partial_evidence (36)

Note that the relation 'could_lead_to' was not in the original ontological model, but was introduced to serve as a first hint towards modeling of causal relations.

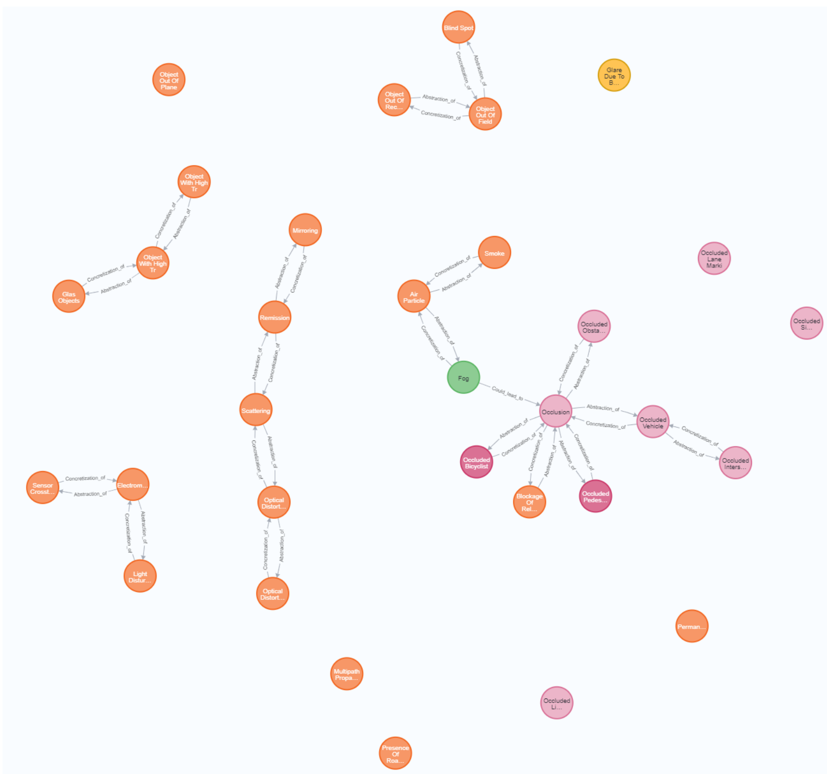


Fig. 15: A database query for criticality phenomena with the attribute 'Perfect-Camera' yields 32 nodes.

4.2.1.3 Query Examples The query language of the Neo4j CP database is *Cypher*. Entering such queries the user can filter them according to their attributes and relations. For example, the following query

MATCH (n:PerfectCamera) RETURN n

will return all CP in the database which are labeled as relevant for a camera sensor under the assumption of a perfect technology. As can be seen from the Figure 15, this query returns 32 CP.

4.2.2 Ontologies as a Knowledge Base for Criticality Phenomena

When considering factors within complex contexts, ontologies present a helpful (and sometimes necessary) tool to achieve a consistent semantic basis of concepts between a wide range of stakeholders. In a broader sense, an ontology represents a conceptualization of some domain, such as urban traffic. In a narrower sense, it is often associated with a *formalization* of such a conceptualization.

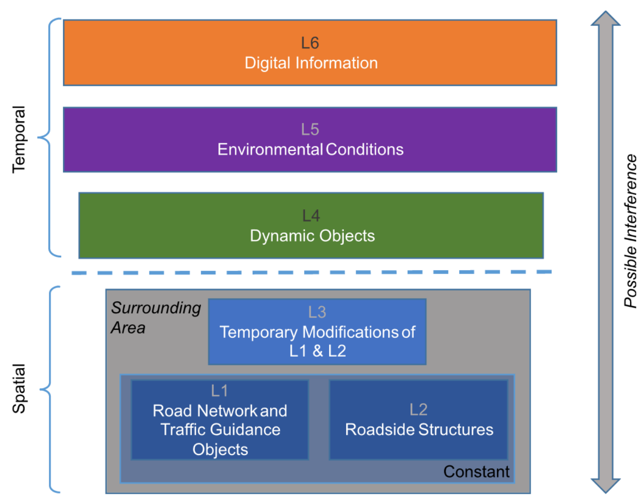


Fig. 16: The 6-Layer Model by Scholtes et al.

For the criticality analysis, the broader and informal conceptualization was examined by Scholtes et al. [20]. The well-known 6-Layer Model (6LM), a deliberately informal structure to categorize entities of the traffic domain, was harmonized between project partners and adapted for an urban context. This model is abstractly presented in Figure 16.

Subsequently, a formal ontology was defined based on this informal categorization [3]. The Automotive Urban Traffic Ontology (A.U.T.O.) is an implementation of the 6LM and available online[6]. It provides terms of the traffic domain, such as `Parking_Vehicle`, assigns them into at least one of the six layers, and provides a semantics in Description Logics, such as that each `Parking_Vehicle` is a `Vehicle` with zero speed. An excerpt of A.U.T.O. is illustrated in Figure 17.

Details on the architecture and its implementation can be found in the corresponding publication [3, Section V].

[6] `https://github.com/lu-w/auto/`

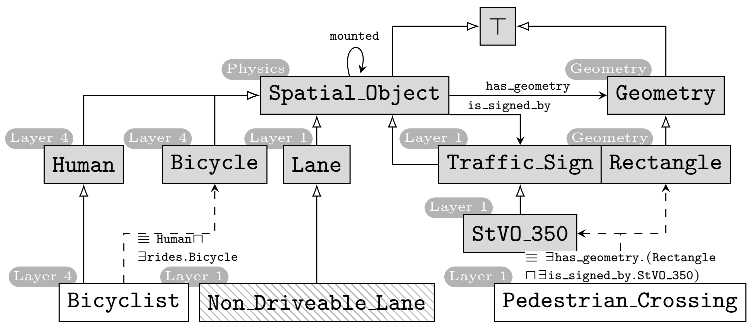


Fig. 17: An excerpt of A.U.T.O.

5 Advances within the Scenario Branch

The process steps 'Scenario Classification' and 'Scenario Instantiation', cf. Figure 2, are heavily intertwined and depend on the level of abstraction. In particular, the process step 'Scenario Classification' implies the specification of a certain scenario class. Scenario specification languages can be formal & informal, graphical, ontology-based, text-based or a combination of these.

In the concept paper, we introduced the *abstract scenario* as *formalized, declarative description of a traffic scenario focusing on complex relations* and closely tied to an ontology, cf. [1, Definition 13]. For the other scenario abstraction levels, we followed the terminology introduced by Menzel et al. i.e., *functional, logical* and *concrete* [21]. In relation to that terminology, the abstract scenario is situated between functional and logical regarding the level of abstraction, cf. [1, Figure 14¸].

5.1 A Simulation Toolchain for Logical Scenario Exploration

In the scope of the VVM project, three so-called *functional use cases* (FUCs) have been introduced [22]. As the scope of this work is to match criticality metrics and CP, in this subsection, we will focus on the 'FUC2-3', cf. Figure 18.

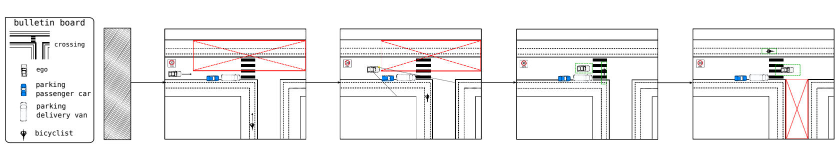


Fig. 18: Functional Use Case 2-3 visualized as Traffic Sequence Chart (TSC).

5.1.1 Logical Parameter Description The derivation of logical scenarios from functional scenarios (*logification*) is a major challenge in the test specification. As part of this work, some basic assumptions were therefore made.

- Unaffected movement of road users: Road users who challenge the ego vehicle are modeled without the possibility of perception and reaction.
- All road users follow the kinematic equations of motion and, as the road users in this scenario are just moving in a longitudinal direction, each road user is modeled using at most three starting parameters (a_0, v_0, s_0).

Additional boundary conditions are added specifically for FUC2-3:

- All road users are initiated with zero acceleration $(a_0 = 0)$. Since the crossing pedestrian also has neither perception nor reactive capabilities, he moves along a prescribed path at a constant speed v_0.
- Collision: we require that both road users, if the ego would follow its path with the given starting parameters (v_0, s_0) without responding to external challenges, are at a common point on the map at a time in the scenario.
- The following independent logical parameters result in their limits for the movement of the two road users from the given boundary conditions:

$$v_{0,ego} \in [2, 30]\text{km/h}\,,$$
$$v_{0,ped} \in [1, 10]\text{km/h},$$
$$\Delta s \in [-1.01, 1.01]\text{m},$$

  where Δs is the lateral position where the pedestrian and the ego intersect.

In addition, the start position of the ego can be derived from the collision condition:

$$s_{0,ego} = v_{0,ego} \cdot \frac{s_{0,ped} + \Delta s}{v_{0,ped}}\,.$$

5.1.2 Criticality Phenomena under Analysis From the CP database, cf. Section 4.2.1, OCCLUSION, REDUCED COEFFICIENT OF FRICTION, and, LIMITED RANGE OF SIGHT were chosen for investigation.

5.1.2.1 Occlusion In FUC2-3, an occlusion is realized by two vehicles parked at the roadside. As to model the degree of occlusion, the distance of the vehicle closest to the crosswalk is denoted by DistStat O. DistStat $O = 0$m is the minimal distance and corresponds to the maximal degree of occlusion. Contrarily, if DistStat $O = 11$m, the parking vehicles are far away from the crosswalk, so that their concealment does not overlap with the concealment of the house's corner, cf. Figure 19. This means that in this case, a scenario without concealment by dynamic objects can be assumed.

5.1.2.2 Coefficient of Friction The coefficient of friction μ is a dimensionless number and is varied within the limits $\mu \in [0, 1]$. $\mu = 1$ represents a road surface with ideal adhesion conditions (dry road at medium temperatures). While $\mu = 0$ does not occur in reality, $\mu = 0.1$ describes the friction coefficient of an icy road.

5.1.2.3 Range of Sight The visibility parameter vis_fog describing the range of sight for camera sensors that assume the sensor and perception tasks in our simulation environment. In the logical scenario, this visibility is in the range of $\text{vis\_fog} \in [40, 170]\text{m}$.

5.1.3 Simulation in CarMaker For the analysis of criticality metrics and phenomena, the simulation environment *CarMaker* is used. Figure 19 shows three different views of the FUC2-3 visualization in CarMaker. The driver's view is shown on the left. This perspective best reflects the images a front camera sensor would provide. Next, the bird-eye view in the middle provides an impartial overview of the scenario. Here, the pedestrian and his intention of crossing the road can already be seen on the sidewalk between the house and the road. The perspective on the right shows the 3rd-person-view known from computer games, looking over the vehicle's 'shoulder'.

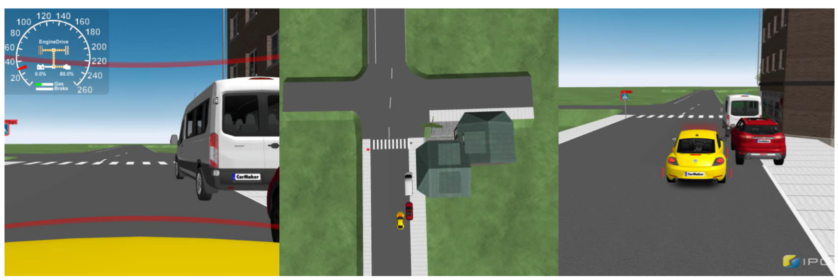


Fig. 19: Three-view visualization from the CarMaker simulation environment for a concrete instance of FUC2-3.

In this simulation setup, the pedestrian is modeled without his own perception and decision-making skills, i.e. he strictly follows the assigned paths.

5.1.3.1 Map Modeling It begins with the selection of a suitable junction and the corresponding map material. The maps are provided in the OpenDRIVE (xodr) [23] format and are read into CarMaker via an xodr-converter and converted into the CarMaker format (rd5).

In the CarMaker Scenario Editor, cf. Figure 20, crosswalks, buildings, plants, parked vehicles, etc. are then placed and, if necessary, parameterized. In our case, the starting position of the ego vehicle and the positions of the parked, occluding vehicles are parametrized.

5.1.3.2 Vehicle Modeling A standard vehicle from the CarMaker database is used as vehicle model to which the necessary sensors and actuators are added. One goal was to implement the criticality phenomenon of occlusion with the help of a suitable sensor, so that the influence on object recognition is guaranteed and measurable.

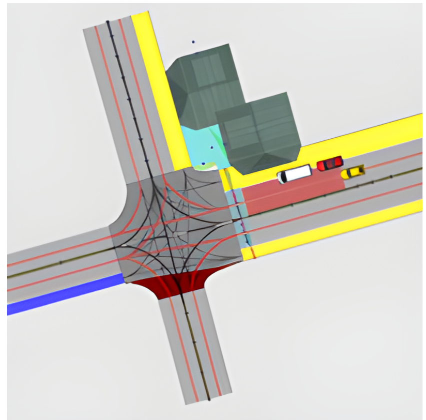


Fig. 20: Snapshot of the CarMaker scenario editor.

5.1.3.3 Sensors and Perception A high-fidelity camera sensor was chosen to take occluding elements into account when recognizing objects. The camera generates an object list of objects that exist in the simulation. The objects' bounding boxes are used to take the occlusion into account, cf. Figure 21.

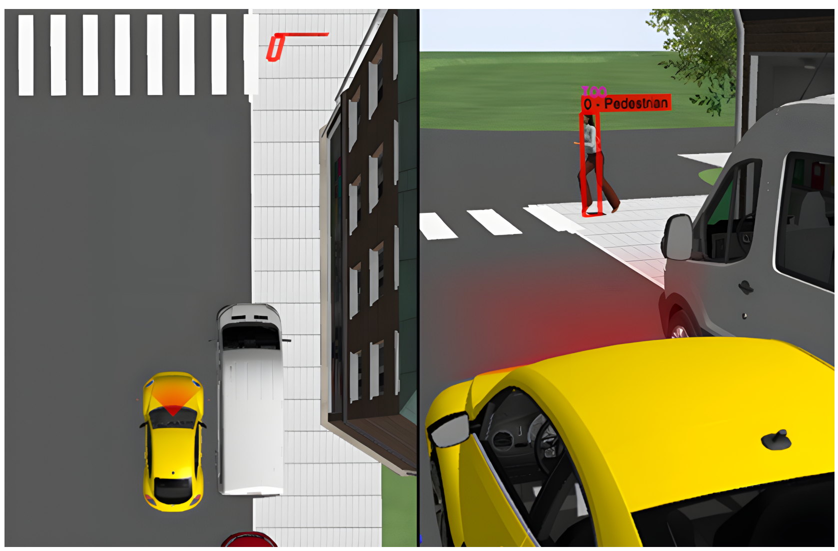


Fig. 21: Visualization of a pedestrian's bounding box in CarMaker.

The bounding boxes contain a defined number of visible pixels. During the simulation, the number of actually visible pixels is continuously determined, considering occluding bounding boxes of the traffic objects.

5.1.3.4 Driving Automation, Driving Intelligence, and Planner As driving automation, the *IPG Driver* is used as a controller for following a prescribed path and speed on a map. The IPG Driver offers the possibility of adding control actions to vehicle simulation, e.g. a braking function. The functionality to recognize traffic objects automatically was deactivated.

5.1.3.5 Actuator, Control, and Actuators A simple brake model calculates required brake torques based on a defined deceleration value to make a full stop when the perception detects the object in the field of view of the sensor, regardless of where the vehicle is on the route. Then a flag is set, and the brake torque (based on target deceleration) is applied to the four wheels. This model was built with MATLAB Simulink[7], compiled to a Functional Mock-up Unit, and implemented in CarMaker.

5.1.3.6 Sampling OptiSlang[8] is used to control the entire simulation process on the one hand and to plan and carry out the design of experiment (DoE) and the sensitivity analysis on the other hand. The parameter selection for a CarMaker test run is controlled in the IPG-Automotive node, while the CriSys node contains the stored Python program CriSys with the criticality calculation. The selected parameters and additional arithmetic operations are carried out in the higher-level process control of OptiSlang. Furthermore, parameter limits, distributions and the form of distribution are specified, cf Figure 22.

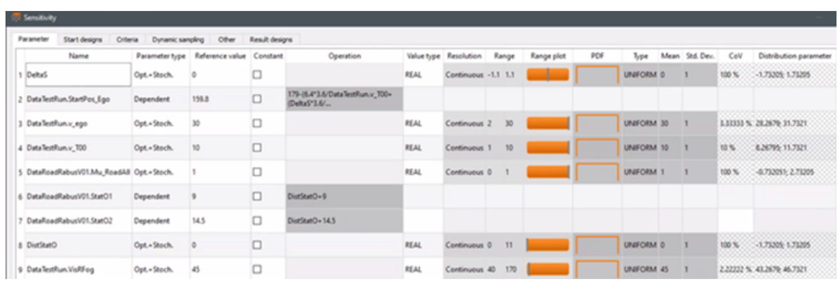


Fig. 22: Parameter section of OptiSlang sensitivity wizard.

It is assumed that for deterministic sampling methods, where the samples are distributed very regularly in space, the number of necessary samples increases exponentially with the number of parameter dimensions. Thus, stochastic sampling strategies were chosen to reduce the number of samples. Another relevant point in a sensitivity study is the correct evaluation of the correlations of the input parameters, so that Advanced Latin Hypercube Sampling (ALHS) is used in the OptiSlang process described above. After determining the number of samples, the process chain is started, and the corresponding criticality is calculated

[7] de.mathworks.com/products/simulink
[8] ansys.com/products/connect/ansys-optislang

for each variation. In this case, parallelization by OptiSlang is possible according to the availability of licenses. The process ends with the output of the OptiSlang post-processing file and the storage of the desired files to be backed up.

5.1.3.7 Simulation For each entry in the DoE, a specific scenario is stored in the format of a CarMaker test run and transferred to it as input for a simulation run. As a result, the description of the specific trajectories of all relevant road users in tabular form of a comma-separated text file is obtained from CarMaker. Here, each line represents a time step at equidistant intervals of 100 milliseconds. The columns show the values for the x and y positions of each road user as well as further, in CarMaker definable signals. For the calculation of the presented criticality dimensions, the speeds and accelerations of all participants are therefore also output here. Based on the trajectories, derived velocities and accelerations and selected signals, some measures may need, and CarMaker is able to provide, we get a file with time rows for each of these parameters as output of the simulation.

5.1.3.8 Evaluation of Criticality Metrics The simulation output-file was analyzed using the Criticality Identification System (CriSys), developed by ZF [24]. As shown in Figure 23, CriSys reads scenarios from various data formats, e.g. the CarMaker output-file format, whether it comes from a simulated scenario, a real-world scenario or whether it was stored in a intermediate database. Scenario data can be processed by CriSys either by evaluating already implemented criticality metrics or by linking external metrics. While MerLin, COP and TTC are implemented directly in CriSys, the Evasion Threat Metrics, cf. Paragraph 4.1.4.3, was implemented in MatLab and linked to CriSys.

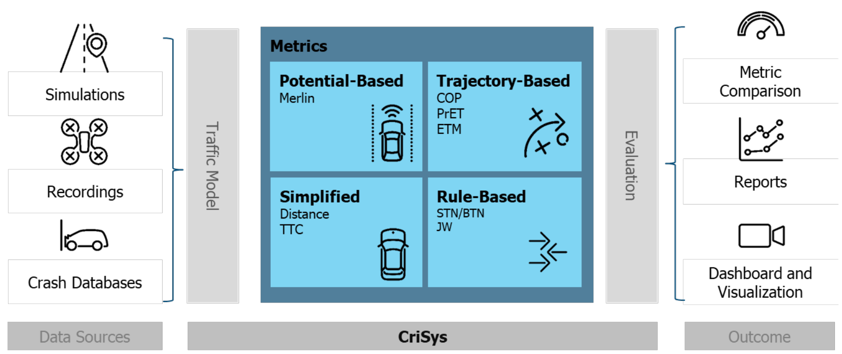


Fig. 23: The architecture of the Criticality Identification System (CriSys) [24].

All selected and calculated criticality metrics were added as an additional time row per measurement to the input file, so that for each time step the criticality is available besides the trajectory data and additional signals.

5.1.3.9 Visualization of Measured Criticality For criticality visualization, we use a web application based on Streamlit[9] and the Python libraries pyplot and altair. Concrete scenarios are visualized by coloured dots in a three-dimensional parameter space, cf. Figures 24 and 25. As criticality is time-dependent, it is aggregated to a scalar by taking the maximum w.r.t. time. Due to the free selection of the available axes and the possibility of filtering each column of the input variables into the DoE as well as the output variables from the simulation, the web application offers the possibility to clarify problems and insights through individual views.

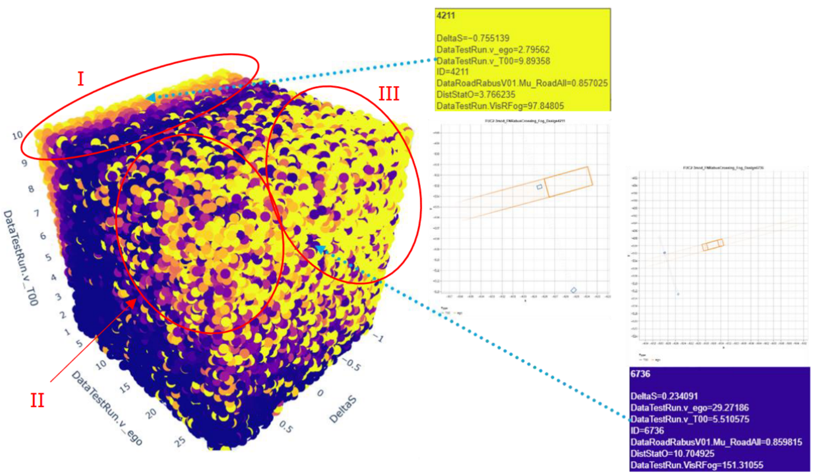


Fig. 24: Visualization of a campaign of 100 000 concrete scenarios using MerLin on the color-axis, and with two detailed scenario views.

5.1.3.10 Results In the following the three base parameters, discussed in Section 5.1.1, are shown on the x-, y- and z-axis and the criticality metric on the color-axis. First, consider the results for the MerLin criticality metric depicted by Figure 24:

- The higher the ego velocity, the more criticality increases (due to a increased braking distance).
- The higher the pedestrian's velocity, the more criticality increases (due to the shorter time for the ego's reaction).
- The closer the collision point is located to the nearside of the ego, the more criticality increases.

In addition to these obvious characteristics, two other things can be observed:

[9] https://streamlit.io

- There is a high criticality at very low ego velocities, combined with high pedestrian velocities (region I).
- In the high criticality area (region III - high velocities, more critical to the near side) it looks like there is some 'noise' in the criticality metric, as not all the scenarios in that region are highly critical.

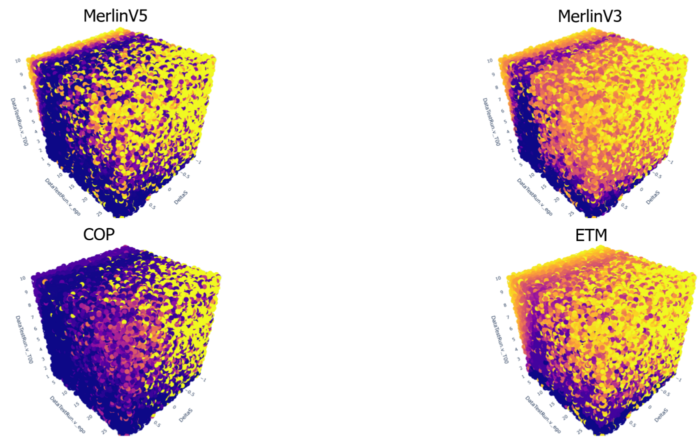


Fig. 25: Parameter spaces colored according to measured criticality for four different criticality metrics.

Region I is shown in trajectory plot 4211 of Figure 24. The ego stopped right in front of the zebra crossing. The reason why the criticality is close to 1, even if there is no crash, is discussed in detail in Section 5.1.4 below. The second issue is that criticality depends on three additional parameters, namely occlusion, friction and sight due to fog, which are hidden in this view, cf. Section 5.1.5 below.

5.1.4 Discussion of Criticality Assessment The campaign of 100 .000 concrete scenarios was processed for each measure. Choosing those different measures for the colour axis of this cube, we get four different views to the scenarios, shown in Figure 13.

In this overview, the criticality is evaluated very different for the various measures. Where it is obvious that the region of high criticality in parameter space is on the near side for high ego velocities and high pedestrian velocities for all measures, it differs in all other regions. The measures on the right side (ETM and MerLin/ DTC) are partly much different. There is the region, mentioned already above, for low ego velocities, that is shown as critical for MerLin and even with more criticality (with more volume) for ETM, this region is not conspicuous for COP. Another region, that can be observed, is the region for a far side hit of the pedestrian. Where ETM shows nearly no decrease, if the pedestrian interaction happens on the nearside to the far side, the decrease of criticality to the far side, is the most for the measure COP. In [23], methods to analyse those differences in criticalities were introduced. By analysing the differences of two or more cubes, the reason for the difference is due to any variation of the simulation toolchain (see section 2.4). Following this, the comparison can be classified into two groups. In the first group, the input data / scenario definition for concrete scenarios is the same for each point in parameter space. Looking at the simulation toolchain, this would be the case, if the driving intelligence / virtual driver is different, the measure, used in criticality assessment or the simulation tool itself. On the other hand, there are parameter spaces for comparison, where the input data is not complete identic to each other. This could be for example by variation of environment conditions, like it was done with the phenomenon parameters, introduced in section 2.3.

5.1.5 Discussion of Criticality Phenomena As introduced in Section 5.1.2, in addition to the three base parameters consisting of the velocities of the two traffic participants and the collision point, there are three additional CP-parameters varied in the sampling scheme that were not visualized above. The effect of those CP-parameters can be visualized using filter criteria for the CP-parameters in combination with the three-dimensional view of the base parameters. Table 5 gives an overview of the filter criteria for the eight CP-combinations under analysis. As the DoE was performed for 100 .000 concrete instantiations and the CP-parameters were varied at continuous scale, it is not possible, to choose just those scenarios, where a CP is completely neglected. Instead, we introduce categories where the dedicated CP has the lowest or the highest influence, respectively. As to define these categories, we choose the lowest and the highest 20% of each CP's parameter range. For the criticality of FUC2-3 without the influence of any CP, the upper 20% of each CP are filtered (see first row in Table 5). In the next three rows, each of the three CP-parameters is reduced to the lowest 20% by keeping the others at the upper 20%. From row five on, each two of the CP-parameters were filtered to the lowest 20% before, in the last row, all three CP were combined.

Figure 26 shows the criticality density per listed range properties, introduced in Table 5, for criticality metric MerLin. While the criticality evaluation without CP shows no highly critical and less moderately critical scenarios, the highly critical scenarios already increase with just a single CP present. Furthermore, a different characteristic can be observed for the different CP: for low visibility the highly critical scenarios are scattered in the parameter space whereas, if an occluding vehicle is standing between 0 and 2.20m in front of the pedestrian crossing they accumulate in region III. For the CP of reduced friction, the criticality evaluation is similar to occlusion. However, the slope of the criticality seems to be much steeper, as there are much less scenarios with criticality around 0.5 (colored orange) than for occlusion.

Table 5: Filter criteria for three dedicated criticality phenomena.

Criticality Phenomenona	μ_{Road} Low-Value	μ_{Road} High-Value	DistStat O All-Low-Value [m]	DistStat O High-Value [m]	VisRFog O Low-Value [m]	VisRFog O High-Value [m]
Without CP	0.8	1.0	8.8	11	144	170
Reduced Visibility	0.8	1.0	8.8	11	40	66
Occlusion	0.8	1.0	0	2.2	144	170
Reduced Friction	0.0	0.2	8.8	11	144	170
Occlusion and Reduced Visibility	0.8	1.0	0	2.2	40	66
Reduced Friction and Visibility	0.0	0.2	8.8	11	40	66
Reduced Friction and Occlusion	0.0	0.2	0.0	2.2	144	170
All three CP combined	0.0	0.2	0.0	2.2	40	66

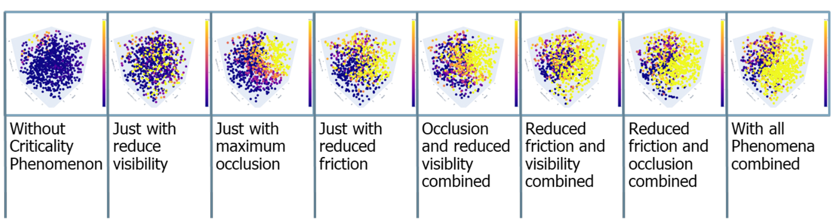


Fig. 26: Overview of criticality evaluation per listed criticality phenomenon combination.

For those filter property sets, modeling the interaction of two CP, the criticality evaluation seems to be super positioned to those with each of those phenomena. The criticality is maximal for all three CP combined. In order to get a better understanding of the differences between these observations, we assess the overall criticality by aggregation to a scalar value. For this, we calculate the mean value and the total sum for each criticality evaluation. Both aggregates are shown in Figure 27.

These diagrams support the hypothesis drawn from the visualizations: it shows an increasing criticality starting without CP, over those with just one, followed by those with two and ending with the criticality evaluation with all three CP combined.

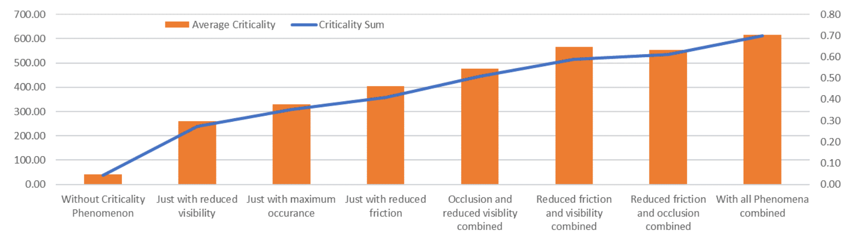


Fig. 27: Line and bar diagrams for criticality aggregation as a sum and average, respectively.

5.2 From Abstract Scenario Specification to Concrete Scenario Simulation

Within the context of automated driving, the need for different abstraction levels for scenario specification was identified by Menzel et al. [21] and initially realized in the hierarchy

$$\text{Functional Scenario} \succeq \text{Logical Scenario} \succeq \text{Concrete Scenario}\,.$$

This hierarchy was extended to include the *abstract scenarios* by Neurohr et al., cf. [1, Definition 13,Figure 14], situated between the functional and the logical scenario regarding its abstraction level. However, abstract scenarios include logical scenarios as a simple special case where all the constraints are explicitly given as probability distributions on intervals of real numbers. Hence, we could also subsume the hierarchy as

$$\text{Functional Scenario} \succeq \text{Abstract Scenario} \succeq \text{Logical Scenario}$$

with the abstract scenario filling the continuum between functional and logical scenario.

There are many approaches to specifying abstract scenario, cf. [25–27] or [28, Section 2.3]. One way of abstract scenario specification are so-called *Traffic Sequence Charts* (TSCs) which is a graphical specification language with well-defined semantics based on formal logic [29]. Within the VVM project, the TSC formalism has been used for the specification of the functional use cases [22]. Two examples are provided by Figure 28 and Figure 29.

The underlying tooling *TSC-Editor* enables the engineer to graphically specify a functional scenario and obtain an abstract scenario with formal semantics is still under development [30]. Specifically, the TSC specification is translated into a SMT-formula which is then solved by a state-of-the-art constraint solver, e.g. Z3. If a solution is found, a trajectory of the world model is returned that suffices the constraint system defined by the corresponding abstract scenario. This solution essentially corresponds to a concrete scenario.

This approach has been integrated with a toolchain called *TSC2OpenX* which translates the resulting world model trajectory into the OpenDRIVE and Open-

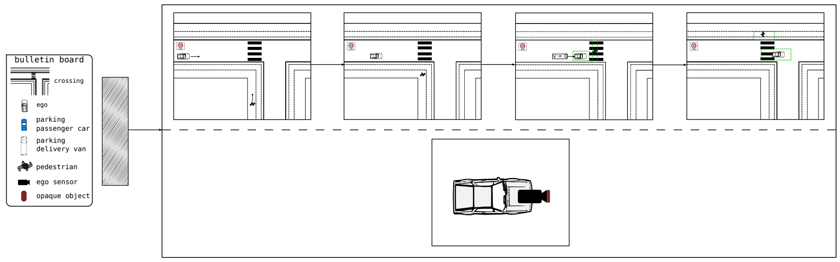


Fig. 28: Functional Use Case 2-4: Urban junction with pedestrian crossing and sensor blockage.

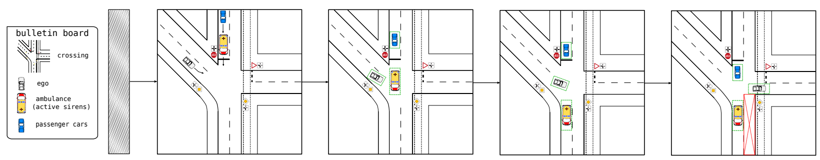


Fig. 29: Functional Use Case 3-3: Leaving priority road with active sirens emergency vehicle.

SCENARIO formats as to enable subsequent simulation [28, 31]. The toolchain also features initial versions of scenario variation methods and associated measures of variation quality. While the TSC2OpenX toolchain is currently limited to highway scenarios, its extension for urban environments is ongoing work.

5.3 Criticality Analysis using Evolutionary Algorithms in Logical Scenario Classes

Another option is to avoid abstract scenarios entirely, going directly from a functional to a logical scenario. In this case, the engineer seeks to represent all scenario parameters that are not single-valued as real-valued intervals or as subsets of the integers. However, as previously mentioned, the size of the parameter space scales exponentially in the amount of parameters of the logical scenario. Thus, complex scenarios with many parameters require obscene amounts of effort to explore their entire parameter space if the exploration is done naïvely. As to circumvent this issues, the literature has demonstrated several times that optimization methods using criticality metrics as fitness functions can be beneficial in comparison to naïve searches.

Fehnker considered evolutionary algorithms and a self-validated criticality metric named PCI to demonstrate an approach of finding the most critical scenario clusters in a given scenario space [17]. As to evaluate the validity of this criticality metric, Fehnker carried out a small self-study for three different logical scenarios. In each of them, they rated several concrete realizations based on their impression regarding the scenarios' criticality and, later on, compared the

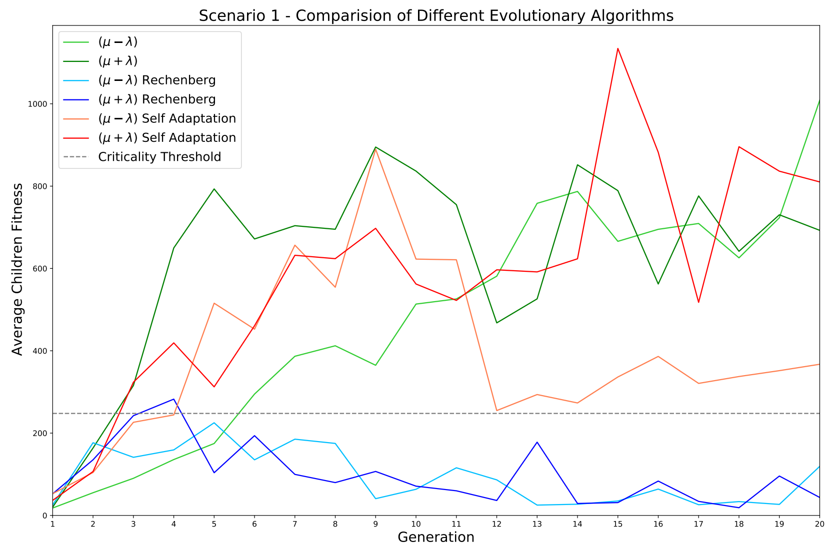


Fig. 30: Comparison of the number of critical scenarios generated by different variants of evolutionary algorithms [17].

results with criticality as measured by the PCI. Relative to this ground truth, Fehnker was able to derive target values for the binary classification of scenarios in critical resp. uncritical using criticality metrics. Additionally, using the concept of ROC-curves, cf. Figure 11, several criticality metrics and their optimal target values can be compared against each other. Further, evolutionary algorithms were used to optimize the measured criticality using the most valid criticality metric according to the comparison, e.g. the PCI with a threshold of 247.79 Joule for a logical scenario corresponding to FUC2-3. Their performance in finding critical scenarios is depicted in Figure 30, with target value for scenario separation as explained above. Even though the clustering of the critical scenarios seemed problematic, Fehnker found a way to cluster and name-tag groups of Note that, even though the validation approach worked on a rather simple example in one scenario, it did fail in 2 other scenarios, thus the other rwo scenarios considered in the work are left out in the optimization study and their performance is not analyzed, since it would not been possible to claim valid results [17].

5.4 Usage of the SOCA Model to Structure Scenarios

The SOCA method, cf. [26], has been developed to be used for a behavior driven structuring of the decision space in which an autonomous system operates. The two key ingredients used to address this task are a topological abstraction of the environment called *zone graph* and a morphological analysis based on *Zwicky boxes*. Using these two tools a situation based analysis is carried out. Figure 31 shows the zone graph and Figure 32 shows a simple set of Zwicky boxes for an exemplary model of the Functional Use Case 2-3: Occlusion of Bicyclist through Parking Cars, cf. Figure 18.

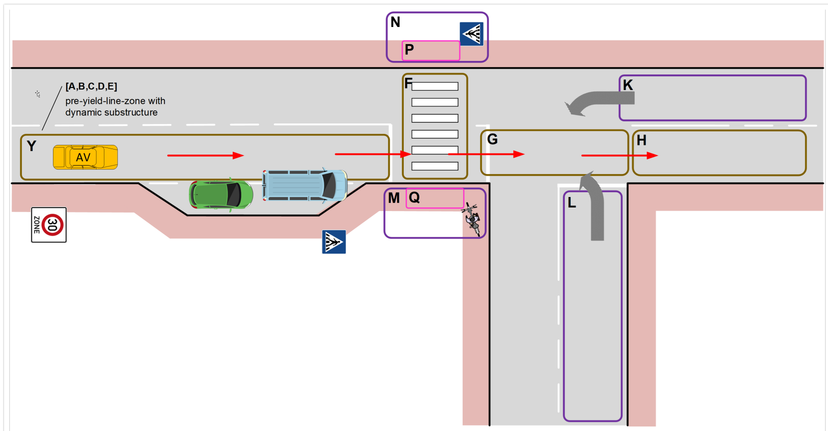
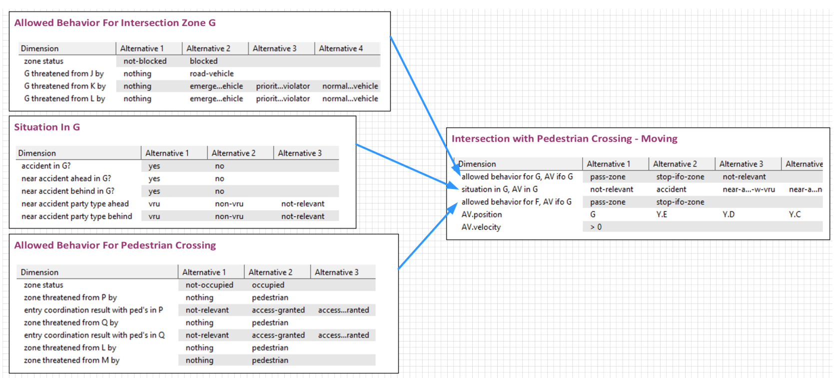


Fig. 31: Zone Graph with underlying Scenery Sketch of Functional Use Case 2-3.



Allowed Behavior For Intersection Zone G

Dimension	Alternative 1	Alternative 2	Alternative 3	Alternative 4
zone status	not-blocked	blocked		
G threatened from J by	nothing	road-vehicle		
G threatened from K by	nothing	emerge...ehicle	priorit...violator	normal...vehicle
G threatened from L by	nothing	emerge...ehicle	priorit...violator	normal...vehicle

Situation In G

Dimension	Alternative 1	Alternative 2	Alternative 3
accident in G?	yes	no	
near accident ahead in G?	yes	no	
near accident behind in G?	yes	no	
near accident party type ahead	vru	non-vru	not-relevant
near accident party type behind	vru	non-vru	not-relevant

Allowed Behavior For Pedestrian Crossing

Dimension	Alternative 1	Alternative 2	Alternative 3
zone status	not-occupied	occupied	
zone threatened from P by	nothing	pedestrian	
entry coordination result with ped's in P	not-relevant	access-granted	access...ranted
zone threatened from Q by	nothing	pedestrian	
entry coordination result with ped's in Q	not-relevant	access-granted	access...ranted
zone threatened from L by	nothing	pedestrian	
zone threatened from M by	nothing	pedestrian	

Intersection with Pedestrian Crossing - Moving

Dimension	Alternative 1	Alternative 2	Alternative 3	Alternative
allowed behavior for G, AV ifo G	pass-zone	stop-ifo-zone	not-relevant	
situation in G, AV in G	not-relevant	accident	near-a...-w-vru	near-a...n
allowed behavior for F, AV ifo G	pass-zone	stop-ifo-zone		
AV.position	G	Y.E	Y.D	Y.C
AV.velocity	> 0			

Fig. 32: Zwicky Boxes of the SOCA model for the Functional Use Case 2-3.

This model covers a large number of scenarios, which can occur within the modeled functional use case. With the help of a language based on regular expressions a restriction to, or a selection of, interesting sequences can be formed. Figure 33 depicts an example for such a sequence. The information in the blue boxes shows the constraints on the sequence of topological positions and AV behavaior specified. The boxes below are the derived conditionals from the analysis.

We would like to draw your attention to the red entries there, which correlate with CP linked to the specific situation. Integrating the criticality phenomena to the Zwicky boxes enables the analysis of the used combinatorics within a modeled situation.

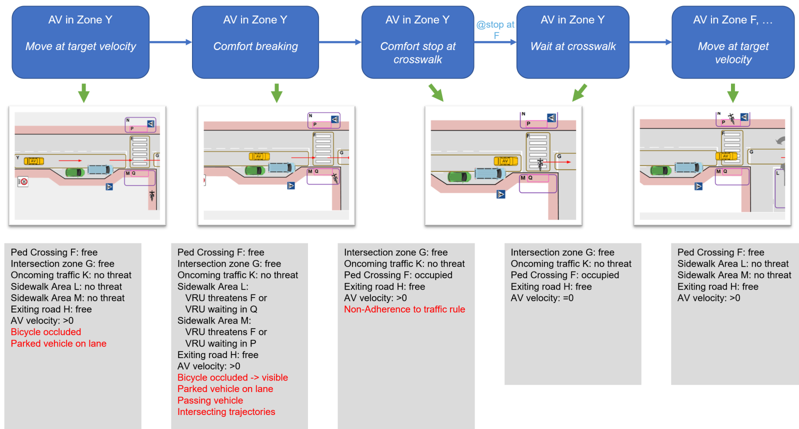


Fig. 33: Derived sequence of situations based on the SOCA model of Functional Use Case 2-3.

Such a sequence of situations can be used e.g. as an abstract scenario formulation. The model artifacts Zone graph, the topological abstraction of the scenery as well as the Zwicky box describing the relevant decision dimensions related to the ADS's behavior can, among others, be used as an input for the derivation of a formal behavior specification, see Salem et.al. [32], or the creation of a phenomenon-signal model (PSM) [33].

6 Downstream Usage of Results

As the criticality analysis is located at the very beginning of a development process, it is paramount to show how its artifacts can be used therein. Therefore, in this section we elaborate on how criticality phenomena, causal relations, and abstract scenarios can be used for the development, verification, and validation of automated driving systems.

6.1 Hazard Analysis, Risk Assessment, and Behavior Specification

After a criticality analysis, the subsequent step is (typically) a hazard analysis and risk assessment, from which safety requirements – in our case, on the behavior level, are derived. We now sketch how the central artifacts of the criticality analysis can be used in both the HARA and requirement derivation.

6.1.1 Usage of Criticality Phenomena Representing abstract classes of danger, cf. Remark 1, the collection of CP provided by the criticality analysis is a source of information for a subsequent hazard analysis and risk assessment (HARA). A HARA is generally conducted for a concrete system as specified in the item definition, but CP are elicited with respect to a whole class of systems. Hence, before hazards can be derived from CP, a concretization step is necessary. Therefore, it needs to be checked whether the CP are relevant for the ADS and ODD under consideration. Depending on the hazard model that is used, the relevant CP can contribute to hazards as possible causes or, when relating them to a specific damage, they can constitute hazards themselves.

Note that the hazard model used within the VVM project is based on ISO Guide 51, cf. [11, Figure 1]. The hazard analysis contributes to the risk-based behavior specification refinement which, naturally, is in correspondence with the behavior specification. Thus, CP are, albeit indirectly, used for the specification of ADS behavior.

6.1.2 Usage of Causal Relations A causal relation explaining a CP transcends the associative nature of the CP by explictly modeling the underlying causal links. Once a causal relation is stable and has been plausibilized through the incorporation of real-world data, it can be used in a variety of ways. Naturally, such information is also of high value within a HARA, e.g. when identifying causal chains leading to a harm in a fault tree analysis [34,35]. Even more importantly, as safe ADS behavior necessitates a causal understanding of criticality, causal relations an serve as a basis for behavior requirements.

6.1.3 Usage of Abstract Scenarios Within a criticality analysis, abstract scenarios are used to encapsulate CP and their causal relations. Abstract scenarios give context to these artifacts and convert them to a machine-readable format which enables their recognition in data or their analysis within a simulation.

Regarding the development of a safety concept, they lend themselves as valuable input for ADS validation on the level of behavior. Instantiating abstract scenarios featuring a plausibilized causal relation can e.g. be used to create a *phenomenon-signal-model* (PSM) [33].

6.1.4 Usage of Safety Principles Safety principles, once established, reduce either the exposure to a CP or the severity of its consequences, cf. Section 3.3. When conducting a HARA, as explained before in Section 6.1.1, CP can be used as a source to derive hazards (or hazardous events) from. For such CP-derived hazards, the potential applicability of associated, established safety principles may offer a possibility to the safety engineer to estimate the *controllability* these hazards.

Moreover, as their effectiveness can be formally verified, safety principles can be seen as abstract classes of behavior specification (similar to rules taught in driving school). During the specification of safety requirements, a requirements engineer can use these abstract safety principles as an initial list of behaviors that have been shown to reduce criticality. Due to their abstract level, safety principles have the advantage of re-usability over multiple systems (again, similar to rules taught in driving school).

6.1.5 Usage of the Ontology The role of the ontology in the criticality analysis is to glue together knowledge and data. As such, it is not a main artifact of the criticality analysis, but it should be considered by anyone who incorporates CP, causal relations or abstract scenarios coming from the criticality analysis. By doing so, a consistent semantic basis is ensured, and downstream artifacts (e.g., hazards) can be linked to the meaning of the criticality analysis artifacts (e.g., the CP). Providing a formal basis for these artifacts hence enables a unified understanding across stakeholders and, therefore, the ontology becomes an indispensable tool for their unambiguous interpretation downstream.

6.2 Contributions of a Criticality Analysis to a Safety Argumentation

Naturally, the question arises how the artifacts of a criticality analysis can be explicitly referenced within a safety argumentation?

Firstly, it is important to state the two *fundamental assumptions* under which a criticality analysis operates, namely that the number of relevant phenomena is manageable (A1) and that they, in principal, are identifiable in the information basis (A2). However, the exemplary conduction within the VVM project has shown that, while these assumptions can not be proven formally, they are quite likely to hold in some form. Of course, this depends on the analyst's notions of 'manageability' and 'relevance'.

Let us consider what we can possibly claim for a CP that underwent the process of the criticality analysis, cf. Figure 2. We know that

1. the CP has been identified and formalized using the information basis,
2. it is associated with increased criticality as measured by a suitable criticality metric,
3. this association is estimated to be sufficient based on available data,
4. the causality underlying the CP's association has been modeled, analyzed, and plausbilized, resulting in a causal relation
5. abstract scenarios that feature the CP and its causal relation have been specified and can used in verification and validation activities downstream.

Note that CP can be concretized to triggering conditions in the sense of the ISO 21448 as soon as the OD becomes an ODD, i.e. when a item definition is available. Compliance with the ISO 21448 is a definite regulatory requirement [36] and a cornerstone of the VVM assurance framework [37].

Therefore, depending on the usage of CP in the development process, the following claims are possible within a safety argumentation

- the CP (or the triggering conditions related to the CP) have been identified, formalized and underwent causal analysis,
- the CP and its causal relation have been incorporated in the development of a safety concept, e.g. as input to a HARA or capability gap analysis,
- the CP and its causal relation have been used for ADS behavior specification,
- abstract scenarios featuring the CP and its causal relation have used to derive test cases,
- the criticality metric associated with the CP has been used to specify test requirements for these test cases,
- the evidences for these claims have been collected, documented and are attached to the safety argumentation.

Let us remark that we do not claim that it is possible to achieve *completeness* of a CP-catalog. However, there seems to be a *saturation effect* regarding the identification of novel CP relative to a growing the information basis: When knowledge and data keeps piling up, but there are no new, sufficiently strong, unexplained criticality associations to be found, the process of the criticality analysis becomes *saturated* w.r.t. the available information basis. Note that such a saturation effect has already be hypothesized by Damm and Galbas [6] and preliminary data for maneuvers exists as well [38].

Claiming a saturation effect regarding CP in a safety argumentation can greatly enhance its rigor. Of course, as the open context of the traffic world changes, e.g. through the introduction of new vehicles or regulations, previously unknown CP might emerge. Therefore, as to keep the claim of being *saturated*, the information basis needs to be updated and monitored for criticality associations regularly.

6.3 Requirement Elicitation for Simulation

For various tasks within a criticality analysis the use of simulation can be advantageous, e.g. for the concretization resp. abstraction of CP, initial plausibilization of causal relations, or the evaluation of safety principles. Each of these use cases comes with certain requirements on the simulation environment, employed actor models and their validity. Therefore, these requirements can used as a secondary artifact of the criticality analysis in order to enhance and further develop the employed simulation software. For more details on the use cases of simulation within a criticality analysis and associated requirements we refer to the deliverable 'Requirements on Simulation for Criticality Analysis' of the VVM project [39].

6.4 Requirement Elicitation for Data Acquisition

Within a criticality analysis, data are required for several process steps. In particular, for estimating a CP's criticality association, plausibilization of causal relations, and the validation of criticality metrics real-world data are required. A criticality analysis can therefore be used for guiding the acquisition of data.

6.4.1 Estimation of Criticality Association Once a potential CP is identified and before moving into causal analysis, the criticality analyst seeks to estimate its association with criticality as measured by apropriate criticality metrics [5], cf. process step 'Estimation of Criticality Association'.

For this, many possible data sources are conceivable such as accident databases (cf. Section 3.1.2), drone data (cf. Section 3.1.3), stationary measurements [40,41] or sensor data from test vehicles, among others. Depending on the CP under consideration, requirements on the data acquisition arise. For example, the abstract CP Adverse Environment Condition and its concretizations, such as Hail or Heavy Rain are unlikely recorded using a drone. Likewise, criticality associations regarding Special Traffic Infrastructure, e.g. Intersecting Tram Rails or Building for Unpredictable Road User Near Road, can hardly be established using stationary measurement.

From these considerations, it is evident that a criticality analyst requires many data sources to get a representative sample that allows a valid estimation of a CP's criticality associations. This requires the data to, at least, be documented whether it is representative or not. The GIDAS database is an example of a data source that claims to be representative, cf. [4]. Of course, there are limitations here as well, as the GIDAS data are a) only recorded after an actual accident with damage to persons, b) the (subjective) memory of the involved persons plays a key role in data collection. Moreover, near-misses, and critical situations that human drivers could handle but may be difficult for the ADS are missing as well. Stationary measurements spanning several weeks or even months, e.g. using AIMATS [41], can be representative at least for the selected measurement area and the selected time frame. The same is true for drone data, such as the inD or highD data sets, but for these the recording time is usually limited. Therefore, to achieve representativeness for a given area using drones repeated measurement flights over the same area are required.

A necessary requirement for a CP to be recognized in a data set is that the CP can be formally expressed in terms the data scheme of the respective data set or, at least, that the formal expression of the CP can be evaluated on an augmented version of the data set. This, in turn, means that data has to be recorded in a way that the relevant CPs are measureable, i.e. the criticality analysis needs to clearly lay out these requirements, and prerecorded data sets may be problematic for some CP [4, Section 4.1]. An obvious example of a CP that can not be reversely engineered by simply looking at the data is reduced coefficient of friction. Moreover, besides the recognition of the relevant CP, the evaluation of potentially interesting criticality metrics imposes further requirements on measurements.

In any case, in order to avoid the problem of ambigious formalizations of a CP, a data-independent formalization using ontologies is recommended [3].

6.4.2 Plausibilization of Causal Relations The plausibilization of causal relations can also be used to guide the collection of data. For this, we summarize the respective section from Koopmann et al. [2, Section 5.1]

In order to answer the causal queries **Q1** and **Q2**, cf. Section 3.2, during the plausibilization of a causal relation in the method branch of a criticality analysis, certain causality indicator functions, cf. [2, Section 3.3], have to evaluated repeatedly. These causality indicators involve interventional quantities in the sense of Pearl [16], i.e. the do-operator, as well as purely stochastic quantities. The estimation of such quantities from data naturally imposes requirements on the data acquisition for both real-world and synthetic data. However, the iterative modeling and plausibilization of causal relations (cf. steps 4.-7. in Figure 10) will have to be done using real-world data, since demonstrating the validity of simulation environments remains an open problem. Once the causal model is established and plausibilized using real-world data and validly represented in a simulation environment, the effectiveness of SPs can be evaluated therein (cf. steps 8.-11. in Figure 10).

While requirements on simulation are covered in the VVM deliverable D09 [39], we focus on the real-world case here. Once an initial expert-based causal relation has been established, it is necessary for iterative plausibilization (steps 4.-6. in Figure 10) of the model to collect data for the relevant variables.

Table 6: Measurability of an exemplary adjustment set for the effect of the criticality phenomenon Reduced Coefficient of Friction on a criticality metric $\mathrm{agg}(\mathrm{STN}_{\mathrm{DT}}, \mathrm{BTN}_{\mathrm{DT}})$ based on the causal structure of Figure 9 [2, Table 4].

Adjustment variable	**Measurability**
Ego vehicle tire temperature	In-vehicle measurement with a sensor.
Planned steering	Can be obtained from the planner component.
Ego vehicle longitudinal wheel slip	Is provided by the electronic stability control (ESC).
Ego vehicle tire wetgrip	Can be inferred from the tire imprint.
Ego vehicle tire type	Can be inferred from the tire imprint.
Planned acceleration	Can be obtained from the planner component.
Ego vehicle tire pressure	In-vehicle measurement with a sensor.
Ego vehicle forward velocity	May be obtained using global navigation satellite system (GNSS) sensors.
Ego vehicle slip angle	Is provided by the electronic stability control (ESC).

Depending on which causal effect is to be determined, there exists a set of possible subsets of the variables, called adjustment sets, that allow for the correct computation of this effect, cf. [2, Section 2.1]. The choice of an adjustment set therefore determines which variables ought to be measured. Let us remark that the opposite is also a feasible option: an adjustment set can be chosen depending on which measurement methods exist and are economically feasible to implement. For the causal relation of the Reduced Coefficient of Friction, depicted by Figure 9, the left column Table 6 lists a possible adjustment set for the effect of Reduced Coefficient of Friction on a generic criticality metric $\mathrm{agg}(\mathrm{STN}_{\mathrm{DT}}, \mathrm{BTN}_{\mathrm{DT}})$. The right column elicits how these variables can be measured, hence defining requirements on real-world data acquisition.

Table 7: Measurability of the CP *Reduced Coefficient of Friction* and a criticality metric $\mathrm{agg}(\mathrm{STN}_{\mathrm{DT}}, \mathrm{BTN}_{\mathrm{DT}})$ based on the causal structure of Figure 9 [2, Table 5].

Exposure and outcome	**Measurability**
Coefficient of friction	Can be approximated from values provided by vehicle sensors.
Aggregate of BTN and STN	Computed based on BTN and STN.
Break threat number	Computed from the required and the minimal available longitudinal acceleration.
Steer threat number	Computed from the required and the minimal available lateral acceleration.
Required longitudinal acceleration	Computed based on ego vehicle forward velocity, planned acceleration and planned steering.
Required lateral acceleration	Computed based on planned acceleration and planned steering.
Maximal available longitudinal acceleration	Computed based on the ego vehicle longitudinal acceleration provided by the physical model, the ego vehicle tire type, the ego vehicle tire wetgrip and the ego vehicles maximal braking torque.
Maximal available lateral acceleration	Computed based on the ego vehicle lateral acceleration provided by the physical model.

Note that in Section 6.4.1 we had to measure the CP of interest and the necessary inputs for criticality metrics (CM). For the plausibilization of causal relations, we have to additionally measure the the variables in the adjustment set (AJ). Whereas before we required representative measurements for CP + CM, now we need the same for CP + CM + AJ. Table 7 lists the corresponding variables for the example *Reduced Coefficient of Friction*.

Note that each causal relation is valid only within a given *context* – a set of statements about the existence of and constraints on individuals in a given domain ontology [2, Definition 1]. This context also imposes requirements on the data collection process in terms of the location, existence of other traffic participants and the setting in general.

7 Dissemination

In this section, we analyze how the various works related to the criticality analysis for ADSs from the VVM project has impacted standardization activities and research in the field of scenario-based safety assessment for automated driving.

As both projects, VVM and SET Level[10], are successors to the well-known PEGASUS project[11], there are significant interconnections between the two projects. While the criticality analysis as a method has been developed and conducted as a sub-project in VVM, it presented a *simulation use case* in SET Level [42]. As such, the VVM criticality analysis has been present at both the SET Level mid-term presentation [43] and the final presentation [44].

On the side of standardization activities, ASAM OpenSCENARIO 2.0 (OSC2) seeks to provide a unified, declarative scenario description language that breaks with the XML-format of OpenSCENARIO 1.x [27]. In particular, one of OSC2's central goals is to enable the specification an subsequent instantiation of abstract scenarios, for which they explicitly reference the scenario qualification, as provided in [1, Figure 14]. Moreover, as another publication from the criticality analysis, the 6-layer model, cf. Section 4.2.2, is mentioned as an approach to define constituents of a scenario.

Moreover, the criticality analysis is featured in the Annex E of the recently published ISO 34502 standard which is´ named *Road vehicles — Test scenarios for automated driving systems — Scenario based safety evaluation framework* [45]. In particular, the Annex E is titled *Derivation and structuring of scenarios using criticality analysis.*

8 Conclusion

As we have seen in the course of this document, a plethora of work has been put into the further development of the criticality analysis as a procedure and, through conduction of its process steps, towards its industrial applicability.

On the one hand, there are many process steps that reached a quite high degree of completion. This includes the identification of CP, their formalization and recognition in data, the creation of an ontology that formalizes an urban OD, the gathering of knowledge on existing criticality metrics, the engineering of novel ones, and the methodical background for the modeling and analysis of causal relations.

On the other hand, process steps such as the estimation of the criticality association of CP, modeling and plausibilization of causal relations, calculation of causal effects, and specification and automated processing of abstract scenarios have only been realized for few examples.

We aim to further invest resources into these promising topics. These open ends of the criticality analysis provide many possibilities for future research and may be considered as a source of ideas for upcoming research projects.

[10] https://setlevel.de/en
[11] https://www.pegasusprojekt.de/en/

References

1. Christian Neurohr, Lukas Westhofen, Martin Butz, Martin Herbert Bollmann, Ulrich Eberle, and Roland Galbas. Criticality Analysis for the Verification and Validation of Automated Vehicles. *IEEE Access*, 9:18016–18041, 2021.
2. Tjark Koopmann, Christian Neurohr, Lina Putze, Lukas Westhofen, Roman Gansch, and Ahmad Adee. Grasping Causality for the Explanation of Criticality for Automated Driving. *arXiv preprint arXiv:2210.15375*, 2022.
3. Lukas Westhofen, Christian Neurohr, Martin Butz, Maike Scholtes, and Michael Schuldes. Using Ontologies for the Formalization and Recognition of Criticality for Automated Driving. *IEEE Open Journal of Intelligent Transportation Systems*, 2022.
4. Stefan Babisch, Christian Neurohr, Lukas Westhofen, Stefan Schoenawa, and Henrik Liers. Leveraging the GIDAS Database for the Criticality Analysis of Automated Driving Systems. *Journal of Advanced Transportation*, 2023, May 2023. Publisher: Hindawi.
5. Lukas Westhofen, Christian Neurohr, Tjark Koopmann, Martin Butz, Barbara Schütt, Fabian Utesch, Birte Neurohr, Christian Gutenkunst, and Eckard Böde. Criticality Metrics for Automated Driving: A Review and Suitability Analysis of the State of the Art. *Archives of Computational Methods in Engineering*, 30(1):1–35, 2023.
6. Werner Damm and Roland Galbas. Exploiting learning and scenario-based specification languages for the verification and validation of highly automated driving. In *2018 IEEE/ACM 1st International Workshop on Software Engineering for AI in Autonomous Systems (SEFAIAS)*, pages 39–46. IEEE, 2018.
7. Marij Van Strien. On the Origins and Foundations of Laplacian Determinism. *Studies in History and Philosophy of Science Part A*, 45:24–31, 2014.
8. Stefan Babisch, Christian Neurohr, Lukas Westhofen, Stefan Schoenawa, and Henrik Liers. Supplementary Material for 'Leveraging the GIDAS Database for the Criticality Analysis of Automated Driving Systems', March 2023.
9. Tino Brade, Birte Kramer, and Christian Neurohr. Paradigms in Scenario-Based Testing for Automated Driving. In *2021 International Symposium on Electrical, Electronics and Information Engineering*, pages 108–114, 2021.
10. Nina Kauffmann, Felix Fahrenkrog, Ludwig Drees, and Florian Raisch. Positive risk balance: a comprehensive framework to ensure vehicle safety. *Ethics and Information Technology*, 24(1):15, 2022.
11. Nayel Fabian Salem, Thomas Kirschbaum, Marcus Nolte, Christian Lalitsch-Schneider, Robert Graubohm, Jan Reich, and Markus Maurer. Risk Management Core–Towards an Explicit Representation of Risk in Automated Driving. *arXiv preprint arXiv:2302.07715*, 2023.
12. Lina Putze, Lukas Westhofen, Tjark Koopmann, Eckard Böde, and Christian Neurohr. On quantification for sotif validation of automated driving systems. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–8, 2023.
13. Arlena Wellßow. *Entwurf eines Prozesses zur Formalisierung sicherheitsrelevanter Phänomene für das automatisierte Fahren.* Master's thesis, Carl-von-Ossietzky Universität Oldenburg, 03 2022.
14. Julian Bock, Robert Krajewski, Tobias Moers, Steffen Runde, Lennart Vater, and Lutz Eckstein. The inD Dataset: A Drone Dataset of Naturalistic Road User Trajectories at German Intersections. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1929–1934. IEEE, 2020.

15. Maike Scholtes, Michael Schuldes, Hendrik Weber, Nicolas Wagener, Michael Hoss, and Lutz Eckstein. Omegaformat: A comprehensive format of traffic recordings for scenario extraction. In *Proc. FAS-Workshop*, pages 195–205, 2022.
16. Judea Pearl et al. Causal inference in statistics: An overview. *Statistics surveys*, 3:96–146, 2009.
17. Anselm Fehnker. *Application of Evolutionary Algorithms to Analyze Criticality in Urban Traffic Scenarios*. Master's thesis, Carl-von-Ossietzky Universität Oldenburg, 05 2022.
18. Patrick Schneider, Martin Butz, Christian Heinzemann, Jens Oehlerking, and Matthias Woehrle. Towards threat metric evaluation in complex urban scenarios. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 1192–1198, 2021.
19. Tobias Merk, Andreas Linnemann, Martin Bollmann, and Mladjan Radic. MerLin - a Potential-based Criticality Measure to Quantify Danger and Risk in Traffic Scenarios, 11 2022.
20. Maike Scholtes, Lukas Westhofen, Lara Ruth Turner, Katrin Lotto, Michael Schuldes, Hendrik Weber, Nicolas Wagener, Christian Neurohr, Martin Herbert Bollmann, Franziska Körtke, Johannes Hiller, Michael Hoss, Julian Bock, and Lutz Eckstein. 6-layer model for a structured description and categorization of urban traffic and environment. *IEEE Access*, 9:59131–59147, 2021.
21. Till Menzel, Gerrit Bagschik, and Markus Maurer. Scenarios for Development, Test and Validation of Automated Vehicles. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1821–1827, 2018.
22. Lukas Westhofen and Christian Neurohr. Functional Use Cases-Characteristic Scenarios for the Evaluation of Urban Driving Automation. VVM Project Midterm Event, 2022.
23. Association for Standardization of Automation and Measuring Systems. ASAM OpenDRIVE, 2020.
24. Bogdan Cojocaru and Johannes Daube. Criticality Identification System (CriSys). VVM Project Midterm Event, 2022.
25. Florian Bock, Christoph Sippl, Aaron Heinz, Christoph Lauer, and Reinhard German. Advantageous Usage of Textual Domain-Specific Languages for Scenario-Driven Development of Automated Driving Functions. In *2019 IEEE International Systems Conference (SysCon)*, pages 1–8, 2019.
26. M. Butz, C. Heinzemann, M. Herrmann, J. Oehlerking, M. Rittel, N. Schalm, and D. Ziegenbein. Soca: Domain analysis for highly automated driving systems. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, 2020.
27. Association for Standardization of Automation and Measuring Systems. ASAM OpenSCENARIO V2.0, 2022.
28. Jan Steffen Becker, Tjark Koopmann, Birte Neurohr, Christian Neurohr, Lukas Westhofen, Boris Wirtz, Eckard Böde, and Werner Damm. *Simulation of Abstract Scenarios: Towards Automated Tooling in Criticality Analysis*, pages 42–51. Zenodo, 2022.
29. Werner Damm, Eike Möhlmann, Thomas Peikenkamp, and Astrid Rakow. A Formal Semantics for Traffic Sequence Charts. *Principles of Modeling: Essays Dedicated to Edward A. Lee on the Occasion of His 60th Birthday*, pages 182–205, 2018.
30. Jan Steffen Becker. Partial consistency for requirement engineering with traffic sequence charts. In *Software Engineering (Workshops)*, 2020.

31. Vincent Kalwa. *TSC2OpenX – Realisierung einer Werkzeugkette zur Simulation abstrakter Verkehrsszenarien.* Bachelor's thesis, Carl-von-Ossietzky Universität Oldenburg, 09 2021.
32. Nayel Fabian Salem, Veronica Haber, Matthias Rauschenbach, Marcus Nolte, Jan Reich, Torben Stolte, Robert Graubohm, and Markus Maurer. Ein beitrag zur durchgängigen, formalen verhaltensspezifikation automatisierter straßenfahrzeuge, 2022.
33. Hans Nikolaus Beck, Nayel Fabian Salem, Veronica Haber, Matthias Rauschenbach, and Jan Reich. Phenomenon-Signal Model: Formalisation, Graph and Application. *arXiv preprint arXiv:2207.09996*, 2022.
34. Birte Kramer, Christian Neurohr, Matthias Büker, Eckard Böde, Martin Fränzle, and Werner Damm. Identification and Quantification of Hazardous Scenarios for Automated Driving. In Marc Zeller and Kai Höfig, editors, *Model-Based Safety and Assessment*, pages 163–178, Cham, 2020. Springer International Publishing.
35. Eckard Böde, Matthias Büker, Werner Damm, Martin Fränzle, Birte Kramer, Christian Neurohr, and Sebastian Vander Maelen. Identifikation und Quantifizierung von Automationsrisiken für hochautomatisierte Fahrfunktionen. Technical report, OFFIS e.V., 2019.
36. United Nations Economic Commission for Europe (UNECE). UN Regulation No. 157: Uniform provisions concerning the approval of vehicles with regard to Automated Lane Keeping Systems, 2022.
37. Jan Reich and Marcus Nolte. VVM Assurance Argumentation - How to systematically organize the approval concerns for safe AD systems in a structured framework? VVM Project Midterm Event, 2022.
38. Lukas Hartjen, Robin Philipp, Fabian Schuldt, and Bernhard Friedrich. Saturation Effects in Recorded Maneuver Data for the Test of Automated Driving. In *13. Uni-DAS e.V. Workshop Fahrerassistenz und automatisiertes Fahren*, 2020.
39. Christian Neurohr, Lukas Westhofen, Tuan Duong Quang, Tjark Koopmann, Roman Gansch, and Stefan Schoenawa. Requirements on Simulation for Criticality Analysis. Technical report, VVM Project, 2023.
40. Sascha Knake-Langhorst and Kay Gimm. AIM Research Intersection: Instrument for traffic detection and behavior assessment for a complex urban intersection. *Journal of large-scale research facilities JLSRF*, 2:A65–A65, 2016.
41. Christian T Erbsmehl, Thomas Lich, and Jorge Mallada. How to Link Accident Data and Road Traffic Measurements to Enable ADAS/AD Simulation? *Journal of Mechanics Engineering and Automation*, 2019.
42. Arun Das, Günter Ehmen, and Thomas Platzer. Simulation Use Case 1 – A Traffic Simulation for Criticality Analysis. SET Level Mid Term Presentation, 2021.
43. Mark Schiementz, Christian Neurohr, and Henning Mosebach. VVMethods and SET Level – Towards a Comprehensive Framework for AD Safety Ensurance. SET Level Mid Term Presentation, 2021.
44. Roland Galbas, Christian Neurohr, Philipp Rosenberger, and Gerhard Schunk. Application of SET Level results in the VVMethods Project. SET Level Final Presentation, 2022.
45. International Organization for Standardization. ISO 34502: Road vehicles – Test scenarios for automated driving systems — Scenario based safety evaluation framework, 2022.