Deliverable D13

Scenario-based Model of the ODD through Scenario Databases

Version 0.1

Editor Michael Schuldes, ika

Project coordinator Robert Bosch GmbH und BMW AG

Due date 30.12.2023

Creation date 19.12.2023

Publication 30.12.2023

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

Supported by:

Federal Ministry for Economic Affairs and Climate Action

on the basis of a decision by the German Bundestag

Document Information

Authors

Christoph Glasmacher, ika – RWTH Aachen

Michael Schuldes, ika – RWTH Aachen

Prasankumar Topalakatti, AVL

Petar Hristov, IT Center – RWTH Aachen

Hendrik Weber, RWTH Aachen

Lutz Eckstein, ika – RWTH Aachen

Reviewer

TP8

Contact

Michael Schuldes	Christoph Glasmacher
ika	ika
52074 Aachen	52074 Aachen
Deutschland	Deutschland

Phone: +49 241 80 27010

Email: Michael.schuldes@ika.rwth-aachen.demich

Revision log

Version	Date	Comment	Author	Partner
0.1	01.11.2023	Initial document	Michael Schuldes	Ika
0.2	08.12.2023	Add content	TP8	
0.3	19.12.2023	Review Stage	Michael Schuldes	Ika

Table of content

List of Figures

List of Tables

1 Introduction

The introduction of higher automation levels of automated driving systems (ADS) inhibits the potential of increasing the efficiency and safety of traffic. To move the responsibility from the human driver to the automated driving function, the validation and verification of the system are crucial. The evaluation of such systems in real-world traffic proves to be challenging due to the high complexity and variability of the open context. An alternative, already suggested by the Pegasus project[1] is scenario-based testing. Here the traffic is decomposed into distinct scenarios, which can be systematically evaluated.

This document, Deliverable 13, provides a methodology (Sec. 2) of how to derive a set of scenarios that represent an operational domain and utilize data to define their characteristics (Sec. 3). Further, the generation of simulation instructions on distinct levels from these scenarios and the management and provision of scenarios to users are described. Next to this core topic, the practices required to acquire data are explained. This contains data acquisition and labeling practices. During the development, it can already be required to exchange data between different stakeholders. For this, a development database is utilized which is described in Sec. 3.1.

1.1 Relation to the VVM Overall Methodology

As suggested by the Pegasus project, scenario-based validation and verification of ADS is a promising approach. In VVM, this approach was extended to cover the urban use case and show the advantages of scenario-based testing already in the specification and design stages. Key in this approach is the modeling of an Operational Design Domain (ODD) through a set of scenarios. Sets of scenarios that can cover the ODD and are non-overlapping are called core scenarios (Figure 1).

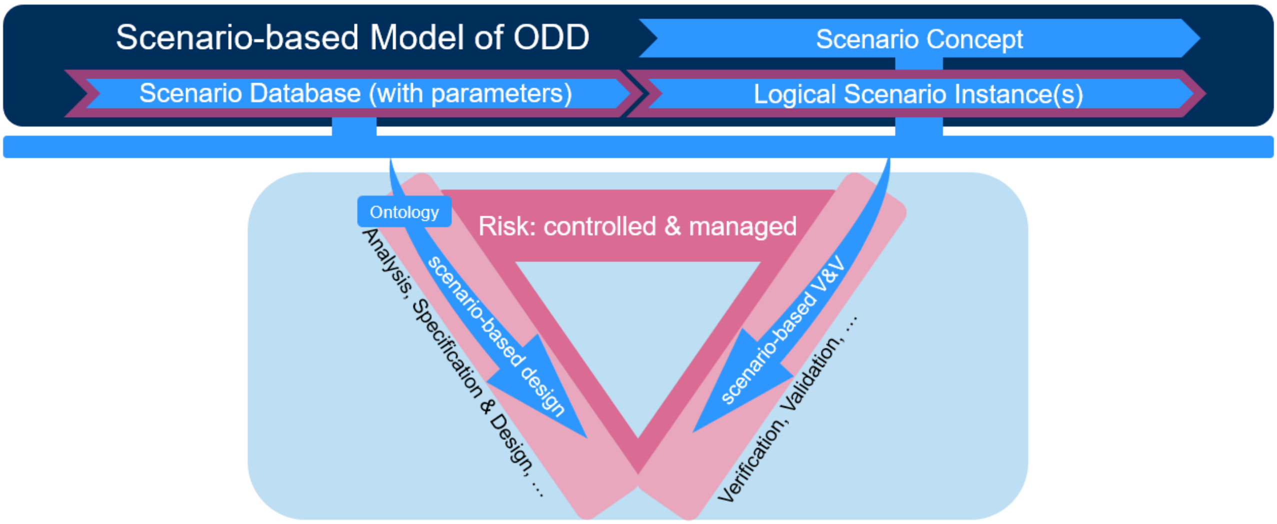


Figure 1: Scenario-based Model of the ODD

[1] Pegasus Project Consortium, "Pegasus Method – An Overview", www.pegasusprojekt.de, 2019

1.2 Method and Data Flow

The goal of the scenario concept is to define a set of scenarios, that together cover the operation domain (OD). The OD is considered for this to not limit the concept to a specific function. To achieve this, a systematic procedure is necessary. The basis of the abstraction is the 6-layer model. It is based on the in the Pegasus project introduced 5-layer-model and structures and defines the ADS system's relevant parts of traffic into six layers: road, roadside, temporary modifications of layers 1 and 2, dynamic objects, environmental conditions, and digital information. Based on this model, an ontology is defined. This can be used to derive a scenario concept. The specific concept suggested by the deliverable is the “Holistic Urban Scenario Concept for Urban Traffic” (Sec. 2.2.3.2). Logical scenarios are created by defining parameters, which are used to define simulation instructions from the scenarios. Through direct knowledge or the analysis of data, parameter distributions approximating the distributions found in the OD can be derived. This functionality is provided by scenario databases.

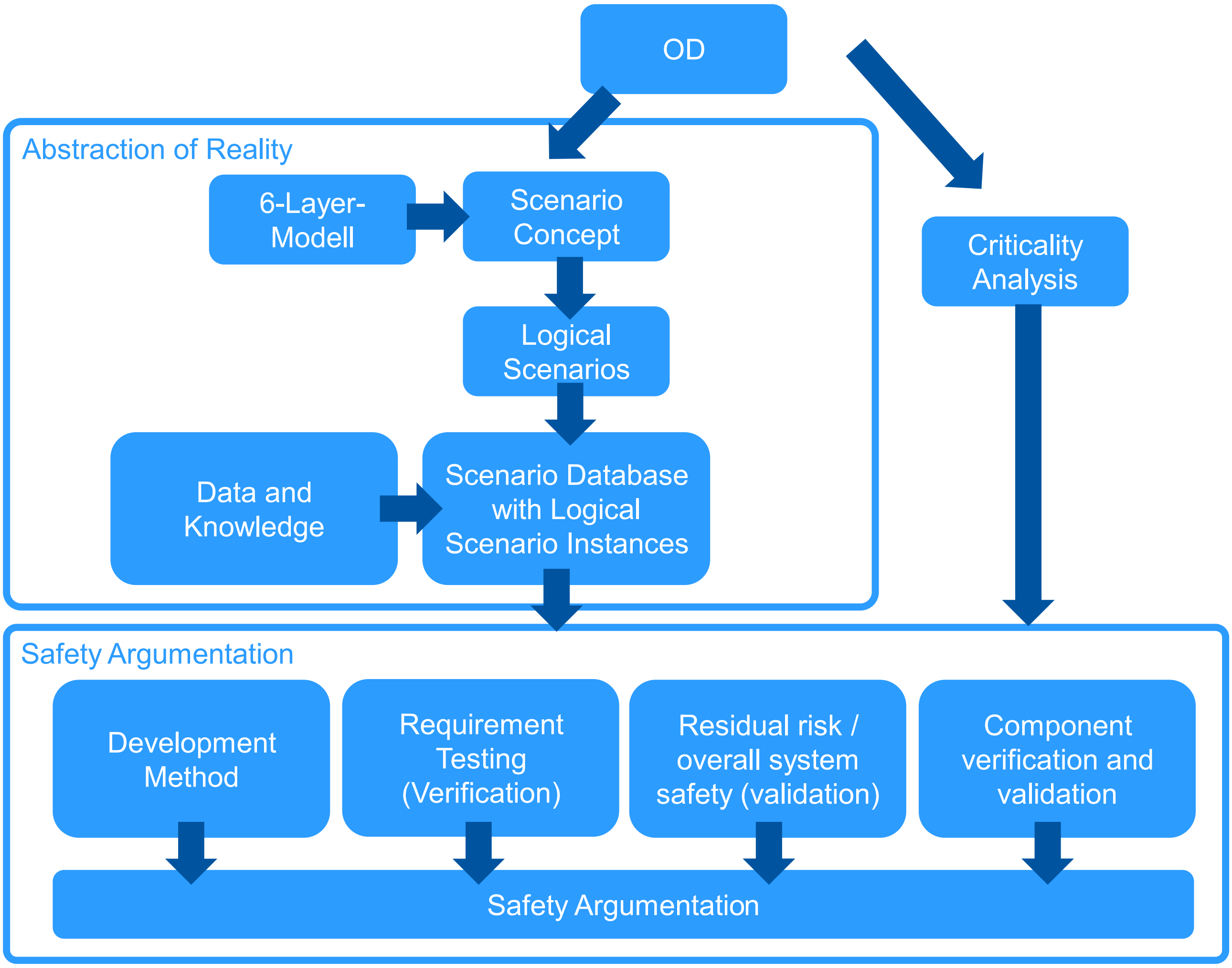


Figure 2: Overall Method

Such scenario databases can be utilized to manage and provide scenarios for testing and therefore for the validation and verification of ADS. Depending on the specific use case, these scenarios are used in:

- The development of ADS,
- the requirement testing (verification)
- the estimation of residual risk and overall system safety (validation)
- and component verification and validation.

Together, these steps feed into an overall argumentation, arguing for the safety of the ADS. Such an argumentation is streamlined by the usage of a single scenario concept through all aspects of the argumentation.

The approach can be separated into the underlying methods, which are explained in Sec. 2 “Scenario Methodology”, and the tools necessary to apply data to the methodology, detailed in Sec. 3 “Scenario Data Flow”.

2 Scenario Methodology

For the validation and verification, and also the development of automated driving systems, it is desired to model real world traffic in a structured and systematic manner. One way to achieve such an abstraction of reality is through the development of a comprehensive scenario methodology, which can be used to define and generate scenarios. A scenario is thereby defined as a sequence of scenes. The implementation may include the usage of actions and triggers. Additionally, an ego vehicle is assigned in each scenario, from whose point of view the scenario is perceived.

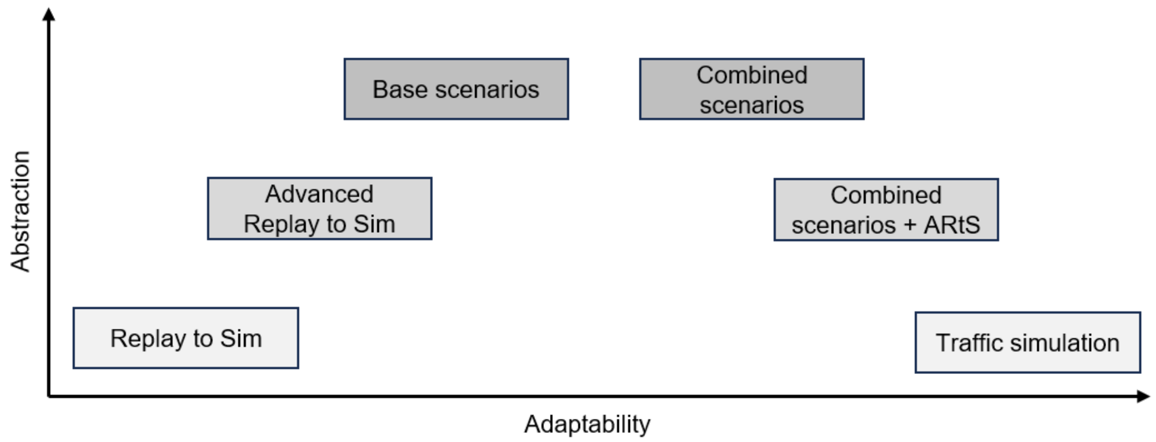


Figure 3: Simulative scenario types of safety assurance

Within the concept of scenario-based testing, different requirements are set to use scenarios for testing. This can on the one hand can be different abstraction levels of scenarios or different level of structuring and parametrization (see Figure 3). The suitability to use a representation for testing highly depends on the needs for testing, availability of models and availability of data. Therefore, it is distinguished between different types:

- **Replay to Sim (RtS):** A replay to simulation is an unchanged replay based on data from recording. The replay is normally as accurate as possible depending on the available data. Thereby, no changes are induced besides deviations caused by interpretation in a simulation. Though it represents the data accurately, the issue of this method is the low flexibility and the divergence of validity if the ego vehicle behaves different to the recording.

- **Advanced Replay to Sim (ARtS):** An advanced replay to simulation approach utilizes a RtS but induces changes to increase the validity due to ego behavior changes. That can include the usage of driver behavior models or synchronizations of other dependencies in a simulation. So, it is more robust against deviations.

- **Base Scenario (BS):** A base scenario is a scenario abstracting the real-world with a predefined set of parameters describing elementary scenarios. The representation with a relatively small set of parameters leads to a higher degree of abstraction but also a higher flexibility and the possibility to generate scenarios out of it. So, it is well-suited for systematic testing.

- **Combined Scenario (CS):** combined scenarios are combinations of base scenarios to manage the complexity of real-world traffic. Thereby, multiple base scenarios as well as additional factors can be included. Those are needed to reflect also more complex constellations systematically.
- **Combination of ARtS and CS:** Parametrized scenarios and replay approaches can be combined to work with a relatively small parameter set but creating detailed scenarios. This approach combines both: the availability to model certain situations accurately and to control them as well as adding multiple other factors as seen in the real world. So, a high level of detail can be combined with structured scenarios. This is especially helpful adding further traffic to a combined scenario to confront a system under test with more potential influences.
- **Traffic Simulation:** Traffic simulations build the most complex opportunity since larger road networks can be tested if suitable road user models are available. However, a system is not directly confronted with controlled scenarios, but those occur in the simulation depending on the behavior of the system under test and the road user behavior models. Although this approach provides the highest adaptability, it relies highly on the used models which have to be available and validated.

To process and store the data, scenarios are divided into different abstraction levels[2]. Whereby a logical scenario is often one important level, here it is split up to make the distinction within the scenario concept more clearly. Therefore, the scenario levels are defined as follows:

- **Functional scenario**: A functional scenario is a behavior-based natural language description of a scenario on a semantic level.
- **Abstract scenario**: An abstract scenario is a formalized, machine readable, declarative description of a traffic scenario with the focus on representing complex relationships.
- **Logical scenario class**: A logical scenario class declares attributes and parameters for abstract scenarios. It also contains parameter limits, dependencies, etc. which are already clear before or filling by data and declare the logical scenario. Actual distributions are not part of the logical scenario class.
- **Logical scenarios instance**: The logical scenario instance is a filled logical scenario class. An instance belongs to a class and fills all declared parameters with concrete distributions by knowledge or data. These distributions can also be trivial distributions and thus represent semi-concrete or even concrete scenarios.
- **Concrete scenario**: A concrete scenario arises from the instantiation of a logical scenario instance by filling it with concrete values.

Based on classifications, a scenario concept is developed to refine scenarios from abstract to concrete. The goal is to define a set of scenarios of limited size that represent an ODD. The scenarios

[2] C. Neurohr, L. Westhofen, M. Butz, M. Bollmann, H. Martin, U. Eberle, R. Galbas „Criticality Analysis for the Verification and Validation of Automated Vehicles", IEEE Access 2021

in this set should be mutually exclusive. In VVM scenarios of such a set are called core scenarios. The above-mentioned base scenarios are one possible variant of such core scenarios.

To fill the concept and map it to the real-world, the process is data-oriented (see Figure 4). Traffic data is recorded and processed from object list level on (see Sec. 3.4). Thereby, a longer ride is cut down into temporal and spatial logical connected enveloping scenarios (see Sec. 2.2.1). On this basis, scenarios are analyzed in multiple ways: On the one hand, the original data is stored and processed to an Advanced Replay to Sim (see Sec. 2.2.2). Furthermore, attributes are annotated, events and base scenarios are detected. An essential input for this analysis is the underlying scenario concept (Sec. 2.2.3.2). For each ego-vehicle, the concept analyzes the real scenarios and splits them into base scenarios. In the next step of abstraction, these annotated scenarios are transformed into a parameterized representation. Thereby, parameterized representation consists not only of unrelated parameters, but also of the causal relations between the parameters.

The abstracted parameters are stored as multivariate distributions and their relations in the database and linked to the original data. During the query, these parameters have to be concretized for the simulation. For this purpose, the scheduled set of logical scenarios instance is taken, and a sufficient set of parameters is chosen. Those can be concretized to translate the concrete parameters in an OpenSCENARIO[3] file for simulations using a sampling method. Optionally, this can be enriched by a suitable Advanced Replay to Sim coming from original data (see Sec. 2.2.2).

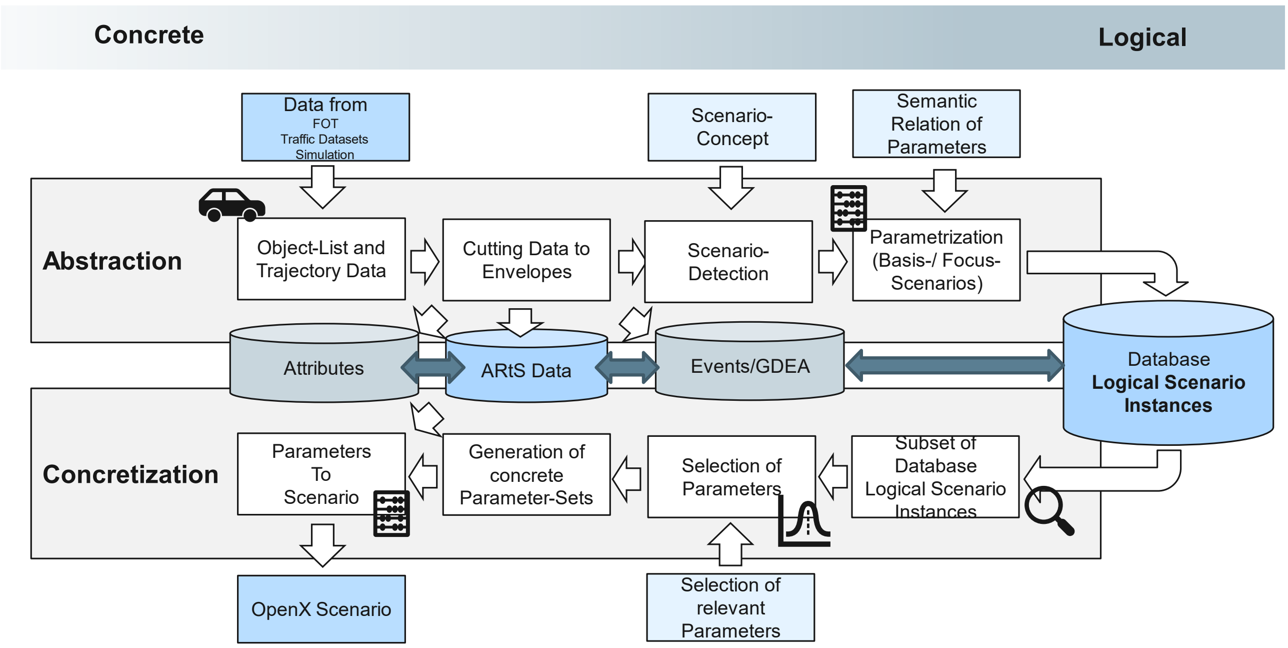


Figure 4: Flowchart from data to scenarios

[3] https://www.asam.net/standards/detail/openscenario/

2.1 6-Layer Model[4]

The 6-layer model defined in VVM is an extension of the 5-layer model defined in Pegasus to adapt to the needs of urban settings. Key task of the 6-layer model is to decompose the relevant aspects of traffic to be able to better talk and reason about the setting. The six layers are defined as follows:

- 6. Digital Information
- 5. Environmental Conditions
- 4. Dynamic Objects
- 3. Temporary Modifications of Layer 1 and 2
- 2. Roadside Structures
- 1. Road Network and Traffic Guidance Objects.

An overview of the layers is shown in Figure 5. *Layer 1*, road network and traffic guidance objects includes everything that is necessary for the navigation and guidance of a road user. *Layer 2*, roadside structures fill the scenery with entities that can have an influence on perception and reasonable occupancy of objects. In *layer 3*, temporary modifications of layers 1 and 2 are defined. The modification is seen as constant throughout the scenario but cannot be expected to be present in subsequent scenarios on the same intersection. Examples are the presence of construction sides and temporary lane markings. The definition of a separate layer for these constructs can be justified by the increase in reusability of representations throughout the lifetime of the traffic infrastructure. In *layer 4*, the dynamic objects and their trajectories are defined. The joined representation of all dynamics in one layer eases the dealing with the complex interactions in time variant descriptions. The influences of weather are defined in *layer 5*, environmental conditions. *Layer 6* includes all digital information, such as vehicle-to-x (V2X) information and the dynamic state of traffic lights and roadside objects.

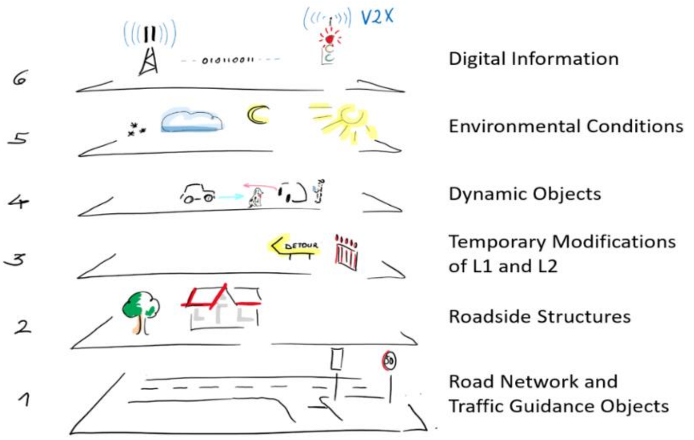


Figure 5: Illustration of the 6-Layer Modell

---

[4] Based on M. Scholtes, L. Westhofen, L. Turner, K. Lotto, M. Schuldes, H. Weber, N. Wagener, C. Neurohr, M. Bollmann, F. Körtke, J. Hiller, M. Hoss, J. Bock and L. Eckstein „6-Layer Model for a Structured Description and Categorization of Urban Traffic and Environment”, IEEE Access, 9, 2021.

2.2 Scenario Concept

To describe traffic systematically, a modular scenario concept for the description, detection and generation of driving scenarios is developed (see Figure 6). It defines logical scenario classes and the relations between them. Firstly, comprehensive traffic is cut into different enveloping scenarios (see Sec. 2.2.1) to logically subdivide intersections, roundabouts, or straight road sections. Furthermore, within an enveloping scenario an ego vehicle is assigned to cut traffic temporally. The description of traffic within those enveloping scenarios is based on the 6-layer model (see Sec. 2.1). On all six levels attributes and parameters are distinguished:

- **Attributes** are used to describe a recorded situation and are purely descriptive in nature. Therefore, they primarily focus on potentially interesting information for the user to characterize the scenario. Duplications of phenomena description and dependencies between attributes are allowed since the same instance can always be referenced and described from different perspectives.
- **Parameters** are used for scenario generation. In this function they serve less the understanding for detecting and searching but need other properties for the actual generation. They should either be independent of each other or the relations between them should be clearly described, so that there are no conflicts in the scenario generation.

According to this division into description and generation, parts of the concept are also divided. This allows for easier comprehension and better usability. In the following, the primary focus is placed on layer 4, since it involves considerable complexity, especially in an urban context. Attributes are therefore divided into events and a semantic decomposition for layer 4. Events are temporally singular events that provide information about the driven sequence (see Sec. 2.2.3.1). The sequence of layer 4 movements can be found in the semantic decomposition and are described in base scenarios.

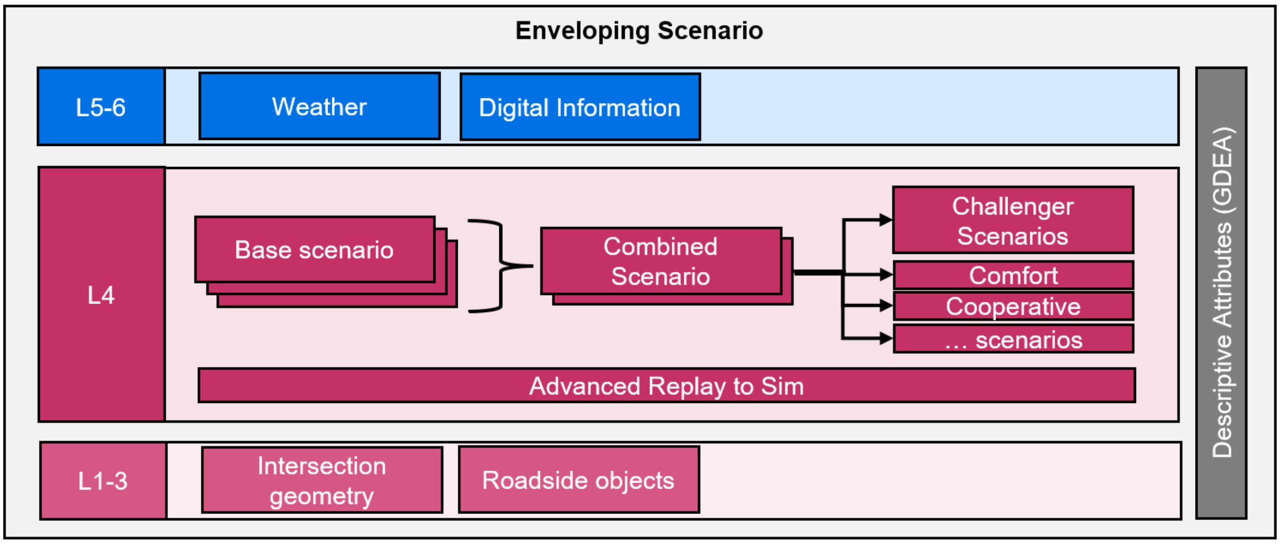


Figure 6: Scenario concept structure

In addition to the description of scenarios, another core aspect of the scenario concept deals with the generation of scenarios. Within the generation, the environment as well as the initial conditions and a goal of the ego vehicle is defined. So, the ego behavior within the scenario is not predefined, but is controlled by the system under test. Since this brings a certain degree of freedom and uncertainty in the scenario, it cannot be ensured that the outcome or timing of the scenario (as e.g., constellations or predefined TTCs) happens as detected in the base scenario or designed by parameters. However, the parameters are designed to lead to a certain constellation as detected and described by attributes so that a situation as e.g., a merging conflict with the challenger vehicle driving in front can be achieved. Despite generating scenarios based on random initial conditions without thinking about potential outcomes, this leads to a more streamlined scenario generation with more relevant scenarios.

Due to different possibilities of generation and usage in simulations, a further distinction is made here. Flexibility in scenario generation is important to vary certain aspects in a scenario or to calculate distributions. To do that efficiently, the scenario must be parametrized. In addition to the abstracted parametric description along the layers, a description using an Advanced Replay to Sim (ARtS) approach is provided (see Sec. 2.2.2). The combination allows a reduction of the described parameter space to essential aspects, which are to be focused and varied if necessary. Still, further aspects can be inserted via the ARtS. Parametrization can be extracted from base scenarios (see Sec. 2.2.3.2). Since only the constellation between two road users is described in base scenarios, more complex scenarios are build using combined scenarios (see Sec. 2.2.3.3). A subset of those can thereby be challengers. However, other types of scenarios with different focusses can be generated.

Although attributes and parameters are stated separately, they have to be linked. Linkage is essential for the scenario concept so that parameters and probabilities of occurrence can be determined for the scenarios to be used for simulation based on a real data analysis.

2.2.1 Enveloping Scenario

An enveloping scenario is the spatially and temporally meaningfully limited canvas on which a scenario takes place. It is used to split longer rides or complex traffic spatially and temporally. The split through traffic movements is thereby spatially oriented and performed so that logically connected infrastructure elements are not separated. For example, an enveloping scenario describes a whole intersection, a traffic circle, or a stretch of road. In addition, the enveloping scenario already focuses on a dedicated vehicle for which the scenario is detected or generated. This vehicle is called ego vehicle. Temporally, therefore, the enveloping scenarios is bounded by entry and exit of the ego vehicle of the tailored environment.

On the one hand, this definition allows to divide a longer trip into several logically separated enveloping scenarios and to describe them separately. On the other hand, this division allows a simplified clustering of similar scenarios since the possible sequences are limited in time. Thus, distributions on similar enveloping scenarios can be combined and distributions for the generation can be calculated in a structured way. In addition, spatial delimitation creates a defined framework for the generation in which individual scenarios can be tested in a meaningful way.

The cut of the respective enveloping scenario is primarily directed at elements of layers 1-3. It is performed in such a way that logically connected sequences are not separated. This means that, for example, an intersection is not cut directly around the actual intersection area, but also includes the entrances and exits to the intersection. These must be considered, since, for example, logically the approach to an intersection must already be assigned to the element intersection and depends, for example, directly on the intended turning maneuver. If, for example, the road user is about to drive through a T-intersection but has a conflict with a cyclist crossing the intersection, he will already slow down when approaching the intersection. This behavior is thus essentially linked to that of the intersection and is considered within an envelope's scenario. So, the used timespan to avoid collisions and the ego vehicle can adjust and act in the situation is given and set by the size of the Enveloping Scenario.

2.2.2 Advanced Replay to Sim

Within the scenario concept, two different options are used for generation: an explicitly parameterized (see Sec. 2.3.4) and non-parametrized description based on the most accurate reproduction of the real-world data. The parameterized representation simplifies to allow for meaningful analysis in the parameter space. In contrast, the non-parametrized description does not explicitly abstract. It allows for a more accurate and relatively detailed scenario without further need for fundamental analysis and additional data processing.

The second is realized via an Advanced Replay to Sim approach (ARtS). Similar to Replay to Sim (RtS), all recorded information is stored without loss in the ARtS. In RtS, all road users follow the trajectories assigned in the originally recorded scenario. An exception can be the ego road user. Since this vehicle includes a system under test, it is the only vehicle that deviates from the originally recorded trajectory and follows a different path. However, this can lead to conflicts since the behavior of the remaining agents do not adapt or react to the deviating behavior of the ego within the simulation. In another development[5], the agents are therefore steered by controllers above a certain threshold of a matric to avoid conflicts. However, the presented approach is not feasible for an OpenSCENARIO-based simulation due to the need of additional driver models which cannot directly be specified in the file and thereby depend on the simulation environment.

The Advanced Replay to Sim approach builds on this approach but ensures compatibility with the OpenSCENARIO standard and enriches the concept. It includes the adaptive behavior of the road users and, additionally, counters the problem of potential settling behavior of automated driving functions. For the adaptive behavior, each road user is assigned an OpenSCENARIO interpretable driver model in addition to its trajectory utilizing actions and triggers. This model is based on criticality metrics as well as speed regulations derived from input data. However, the metrics are not checked for all potential vehicles, but based on a pre-selection based on an analysis made on the original data. In this way, prioritized objects are defined, which each individual agent must consider. Irrelevant objects can be neglected, thus keeping the system real-time capable. This pre-selection takes simple traffic rules into account and ensures that obvious conflicts that are not caused by the misbehavior of the ego vehicles are prevented. Additionally, the pre-selection allows a simpler interpretation for simulation programs since the complexity of the OpenSCENARIO file is reduced.

The driver model itself focuses on modeling the speed of the road users along its originally driven path. It does this based on existing acceleration values from the literature as well as analyzed values from the originally recorded scenario. So, scenario-typical behavior of the road user as measured in the data is considered. A slow pedestrian will thus remain slow in the ARtS approach, although a

---

[5] N. Weber, C. Thiem, U. Konigorski "A Needle in A Haystack - How to Derive Relevant Scenarios for Testing Automated Driving Systems in Urban Areas", arxiv, 2021.

generic controller would have increased the speed due to general simplifications of a driver behavior model. The path stays the same relative to the infrastructure since road users normally do not adapt the path significantly but focus on speed adaption[6]. Thus, on the one hand the characteristic of the agents is taken over and on the other hand the compatibility with different simulation environments is guaranteed by the exclusive use of the OpenSCENARIO standard.

In addition to agent modelling, a settling time can be considered to take a potential setting behavior of the system under test into account. Since the system under test may have components that have a certain settling behavior within the test, this should not influence the scenario itself. The road users are synchronized before relevant aspects of the scenario occur so that the constellation of interest can be investigated reducing unwanted influences (see dots Figure 7). This allows potential transient events to be disregarded and still to test the original wanted scenario.

For more complex and long scenarios further synchronization points for road users can be implemented (see stars, Figure 7). Therefore, e.g., interactions after the intersection are synchronized after the initial start to still get a realistic behavior although the ego road user may have behaved differently when entering the intersection. This synchronization is thereby possible for two reasons: Either the scenario is temporal or spatial relatively long and the road users in the end should be synchronized in between or the behavior of road users are important for the ego and should be synchronized afterwards. This second synchronization has to be handled cautiously since potential unrealistic behavior could arise due to the interaction between initially synchronized and later synchronized road users due to the time shift in their trajectories.

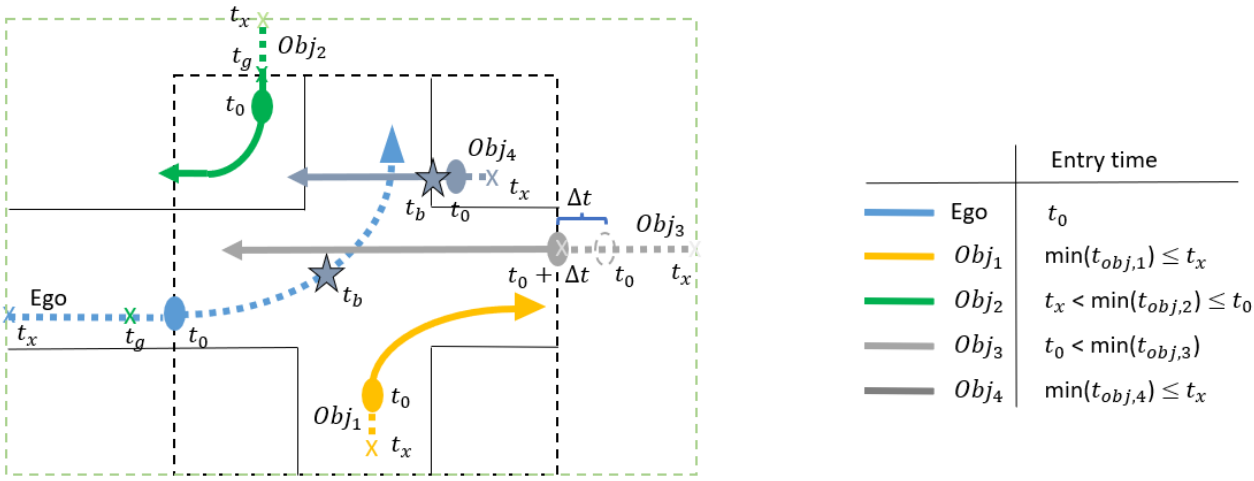


Figure 7: Schematic Advanced Replay to Sim trajectory chart including the boarder of the relevant scenario (black), extrapolation (green), handling of synchronization and starting points with different time steps (dots, crosses and stars), predefined trajectories (solid) and adaptive trajectories (dashed).

To define such a settling time for the ego within the scenario, elements of the scenario has to be extrapolated over the initial replay for the scenario to not limit the scenario itself. Therefore, two boundaries have to be defined: the original boarder of the scenario wherein the system under test should be tested and an extrapolated boarder for the settling time. Depending on the data and road user behavior, it is thereby hard to extrapolate a proper behavior of other road users. So, this is only

[6] H. Weber, L. Eckstein, F. Beringhoff and J. Josten "Towards Modelling Driver Performance Within Crash-Relevant Scenarios As Virtual Reference for the Safety of Automated Vehicles". ESV. Yokohama, 2023.

done if such information is available. If such information is available e.g. because of a longer time of recording of those road users before the relevant scenario, those are extrapolated according to the settling time t_x and controlled so that they reach their initial point and speed of the relevant scenario when the ego enters the boarder of the relevant scenario (black).

Both, the potential adaptive behavior and the consideration of settling behavior of a system under test thereby helps to make a replay to sim more robust and useful for simulation. Anyhow, it sticks to individual recorded situations, so that an exclusive usage makes it hard since deviations from real-world situations are not easily applicable. Therefore, a more systematic description is proposed in Sec. 2.2.3.

2.2.3 Systematical description

Next to the unparametrized description, a systematical description of scenarios is an important addition. It offers the opportunity to describe and vary a scenario systematically. This brings the opportunity to structure the scenario space and sample from it.

Following, we structure the scenario space within the Enveloping Scenarios according to the 6-layer model.

2.2.3.1 Layer 1, 2 and 3

Layers 1, 2 and 3 are tasked with the description of the road network, traffic guidance objects and roadside structures. Also included are the temporary modifications of those (static over duration of the scenario). The parametrization of layer 1, 2 and 3 can be structured into the description of the network itself with information on the lanes and their properties and the positioning and properties of objects. A design criterion of the parameters is that intersections which invoke a similar behavior of the ego vehicle should be described with similar parameter values. This necessitates that the parameters are defined relative to the ego vehicle that is described in the enveloping scenario. The enveloping scenario already divides and structures a drive according to features of layer 1 of the 6-layer model. Therefore, the description and parametrization of layers 1 to 3 can be performed regarding only the local intersection, roundabout or other local characteristics. Since the description is a-posteriori the whole path of the ego vehicle can be used to define the relation.

An intersection, roundabout or section of road is defined through its center point. The orientation is set relative to the lane the ego vehicle enters the section defined by the enveloping scenario. This leads to a semantic definition of the lane network relative to the ego (see Figure 8). The goal is an independence of the actual road definition through a reduction to the semantics. With this, a similarity measure and comparison between different road networks can be achieved.

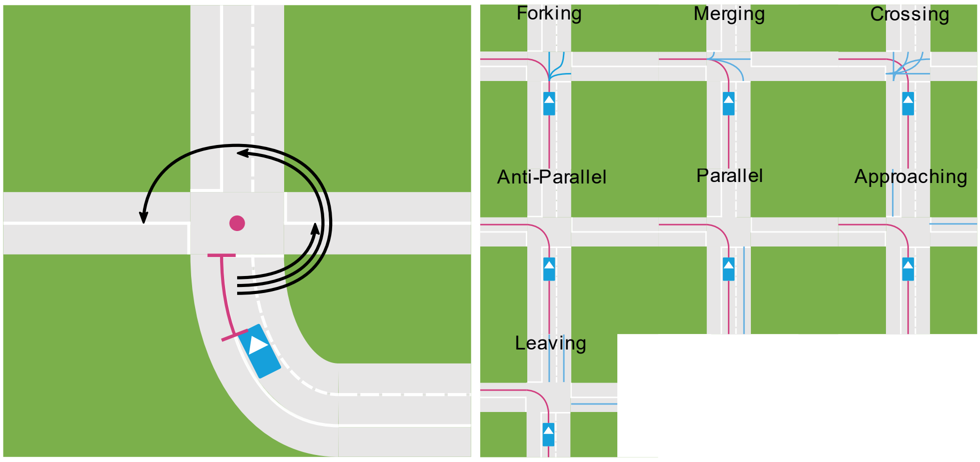


Figure 8: Semantic description of road network in relation to an ego vehicle

Road Objects are positioned through a road coordinate system, similar to that of OpenDRIVE, ensuring that the position adapts when changes to e.g., the angle of the road are made. Object types and characteristics, as well as the road and lane characteristics a defined through an ontology.

2.2.3.2 Layer 4 Concept – Base Scenarios[7]

For the exact description and later playout of Layer 4, two types of scenarios are defined within the concept: base scenarios and combined scenarios. Base scenarios are the elementary building blocks for the description of a road user and for the description of the relation between an ego and another road user. For this purpose, base scenarios are created based on defined concepts via an ontology (see Figure 9).

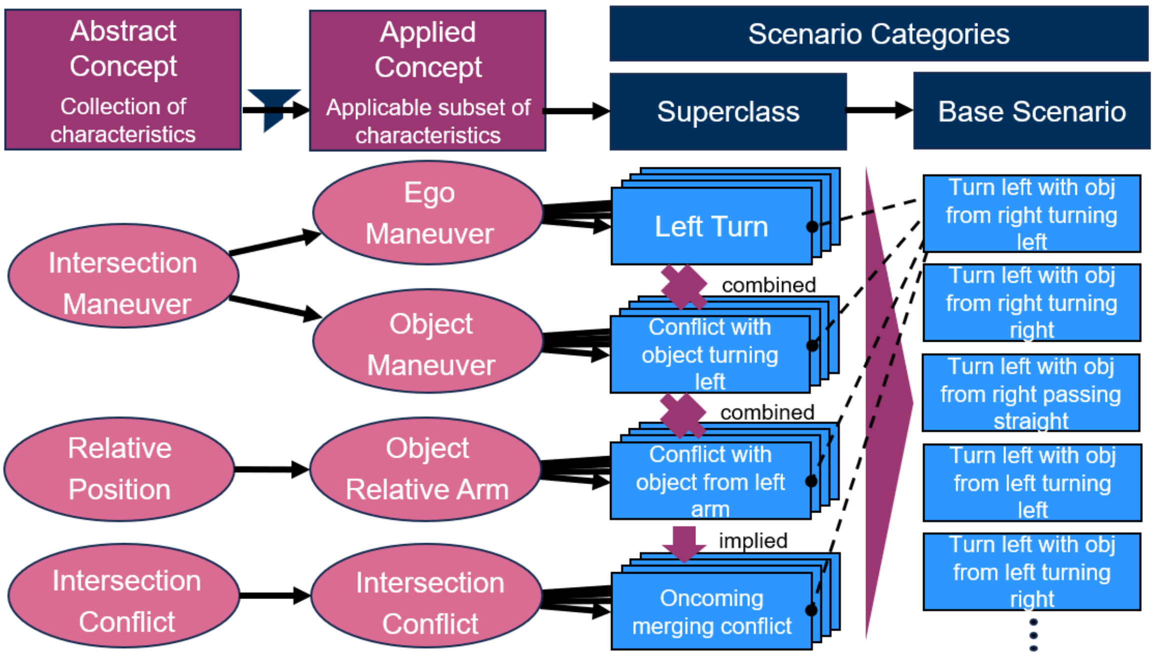


Figure 9: Systematic derivation of base scenarios

[7] Based on H. Weber, C. Glasmacher, M. Schuldes, N. Wagener and L. Eckstein “Holistic Driving Scenario Concept for Urban Traffic”, Intelligent Vehicle Symposium, Anchorage, 2023.

These concepts are divided into individual, bilateral and global concepts depending on the number of road users considered. On this basis, the concepts are assigned to road users and are logically combined, traced by means of an ontology, so that in total 273 scenarios as leaves of the ontology can represent the entirety of constellations in the entire structured traffic space.

In order to represent more complex scenarios on the basis of these basic scenarios, they can be combined (see Figure 10). This combination is possible both sequentially, provided that the base scenarios consider different objects, and sequentially with different or also the same road user.

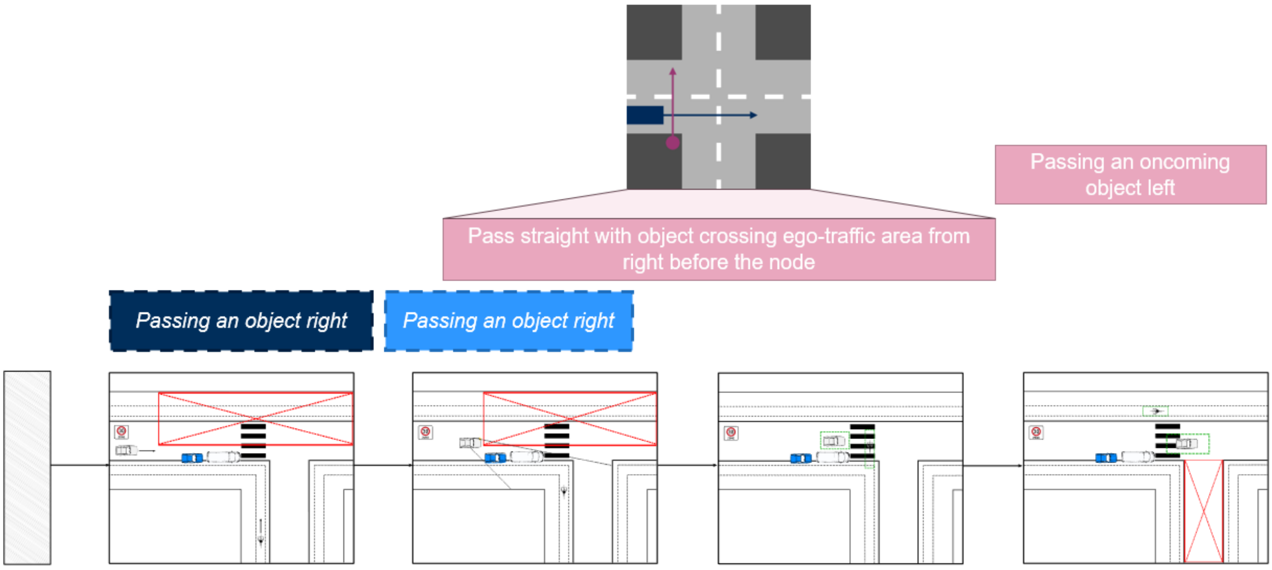


Figure 10: FUC 2-3 Base scenario description

Whereas this elementary form is used for description of existing traffic scenarios, it can be transferred to executable scenarios for simulations (see Figure 11). Those may have a different character since the ego behavior is not specified in advance. So, a recorded base scenario cannot be executed one to one with the guarantee to get the same scenario as recorded. If e.g., the ego decides to brake initially, there may be no influence on a pedestrian so that no intersecting conflict can be detected after simulation. Anyhow, scenarios are generated in a way that it is designed for the desired outcome to represent the real-world. For this, parameters are extracted from the recorded base scenarios and transferred to build the executable scenario.

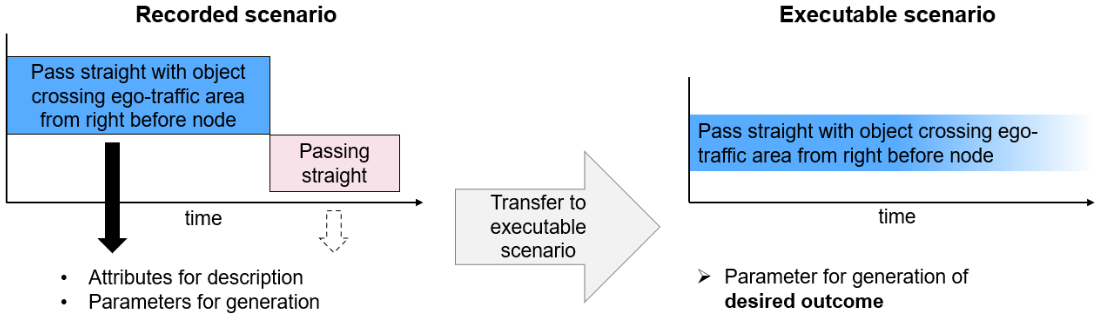


Figure 11: Transfer to executable scenario

2.2.3.3 Layer 4 Concept – Combined Scenarios and Scenario Modifications

Traffic can be more complex than relatively easy Base scenarios. Therefore, Base scenarios can be combined or adapted to represent more complex scenarios. While a simple sequence of base scenarios is sufficient for the description of situations, for the generation, the structure of the combined scenarios is necessary to take design elements of generation into account.

Different groups of combined scenarios and scenario modifications can be distinguished. An important category is the definition of challenger scenarios[8]. In challenger scenarios, there is a need for action on the part of the ego road user in order to avoid a collision. A translation of base scenarios into challenger scenarios can be represented from the sequence of base scenarios. E.g., a challenger type A in an intersection with the start on different intersection arm can be represented as a sequence of a pass straight merging following conflict following an approaching scenario. On a highway, a challenger type A can be presented by a pure approaching base scenario. However, the challenger A can be extended by a following scenario followed by an approaching scenario, so that more flexibility is possible in scenarios. In addition, scenarios that include multiple challengers can also be constructed. This includes, for example, the combination of merging and intersecting conflicts, which occur relatively often with a combination of pedestrians and cars on intersections. Besides the modeling of the challenger conflict, action constraints can be modeled as an overlay similarly to multiple challengers. E.g., an action constraint by a vehicle next to the ego could be modeled as a passing depending on the type of constraint of the vehicle. Occlusion can then be modeled in combination with a proper parametrization in such combined scenarios to define action constraints.

Besides challengers, other combined scenarios and modifications are conceivable. Relevant are e.g., scenarios that consider the attention on other road users without a direct conflict or cooperative driving. Another relevant type are scenarios that do not include a direct challenger, but a conflict occurs due to misbehavior of the ego. In addition to those, comfort scenarios are conceivable that consider regular crossings without a challenger or constraints. Those types of combined scenarios may be differentiated and need different descriptions for generation and efficient testing. Without a dedicated purpose or if the purpose is already met, the base scenario (sequence) according to the combined scenario can be used for generation.

2.2.3.4 Layer 4 – Event-based Storyboard description

The scenario definition described in the previous sections have the limitation, that their concepts have to be translatable to simulation instructions. This limits the possibilities of realization. To define the OD and to aid the search for relevant scenarios, not all descriptions need to be translatable to simulation instructions. Therefore, to provide a description and query language that expresses the sequence of scenes over time, events are defined and arranged into a storyboard. The description of this kind is inspired by how people would describe accidents in an event-based manner. Events have no extent in time but are a fixed single point in time. The following events are defined:

- Status Event (e.g., traffic light change)

---

[8] Hendrik Weber, Julian Bock, Jens Klimke, Christian Roesener, Johannes Hiller, Robert Krajewski, Adrian Zlocki & Lutz Eckstein (2019) A framework for definition of logical scenarios for safety assurance of automated driving, Traffic Injury Prevention, 20:sup1, S65-S70, DOI: 10.1080/15389588.2019.1630827

- Driving/Travelling Direction
- Driving/Object Dynamics
- Geometric Events
    - Passing Event (relation change to static object)
    - Overtaking Event (relation change to dynamic object)
    - Occlusion Event (change in visibility from perspective of ego)
    - Lane Change Event
    - Intersection Entering Event

All events are defined for every object in the scene, except for the occlusion events. These are only described from the ego perspective. This is done to reduce computational resources and does not necessarily have to be the case. Furthermore, geometric events are split into sub-events, since defining a single point in time where e.g., a lane change takes place is difficult. The sub-events are started, majority and complete, combined with a flag for the edge direction. This enables the search for e.g., aborted lane changes and overtaking.

The positioning of objects and their relation are only defined at each event through the construct “dynamics of objects at event’. The positioning is performed in a semantic manner in a lane coordinate system analog to Sec. 2.2.3.1. The hierarchy of the concepts are shown in Figure 12. Additional to the positioning further metrics describing the dynamics at the point in time are stored.

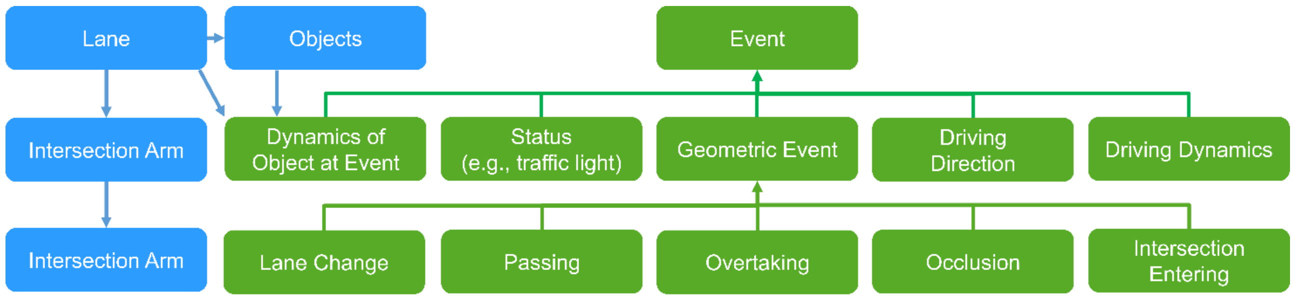


Figure 12: Event hierarchy of the storyboard

2.2.3.5 Layer 5 and 6

Layer 5 of the 6-Layer Model describes environmental factors such as the weather. Because the duration of enveloping scenarios is small, compared to the dynamics in e.g., weather, the description of layer-5 characteristics is simplified to being static. The definition of parameters and attributes is derived from the availability of data in common sources like the Federal German Weather Service (DWD). Examples are the precipitation in mm per minute or the wind speed in m/s.

Layer 6 inhibits digital information. This, for one, includes the state of traffic lights and dynamic traffic signs. These are represented through the status events of the storyboard described in Sec. 2.2.3.4. V2X information is not yet part of the description because it was not in the scope of VVM, but it can be included by e.g., the introduction of virtual objects.

2.2.4 Combination of unparametrized and parametrized description

The two presented concepts of parameterized description (see Sec. 2.2.3) and ARtS (see Sec. 2.2.2) each have individual advantages and disadvantages. While the parameterized description can be varied flexibly, probabilities can be calculated and is easy to use for argumentation, the ARtS offers the possibility to generate large structures without a dedicated parameter description in an uncomplicatedly inflexible way and as in reality. Thereby, it does not abstract reality significantly and can easily handle situations which are not parametrizable.

To provide the opportunity to precisely define a focused scenario and to still have additional road users for a realistic environment, the two concepts can be combined: ARtS and parameterized description are super positioned. However, to allow a combination two problems have to be solved: Potential explicit conflicts between scenario elements as well as non-matching behavior of ARtS and behavior of ARtS and parameterization may occur when combining randomly.

If the parameterization and the ARtS are based on the same initial situation and the parameterization does not deviate too much, the road users described from parametrization can simply be deleted from ARtS both scenarios super positioned (see Figure 13). The result would be a slightly modified ARtS. Moreover, by robustly adapting the road users of the ARtS, potential conflicts by variations can be reduced.

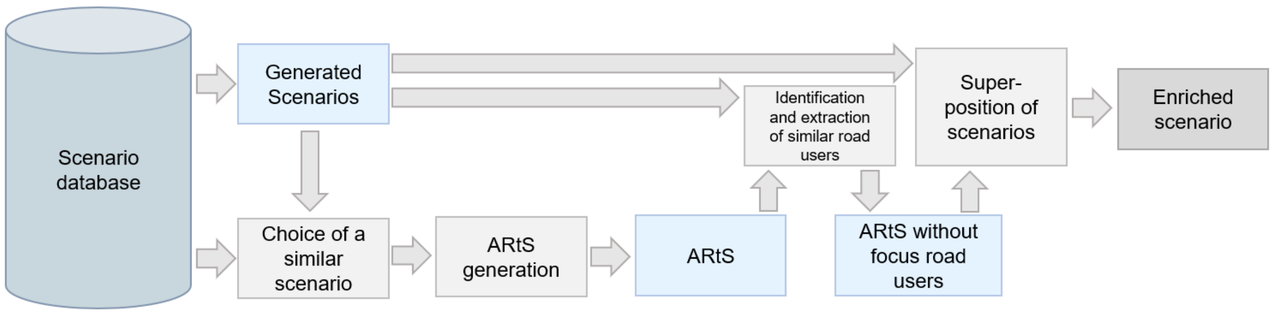


Figure 13: Merging of parametrized and unparametrized scenario elements

If a scenario is generated based on distributions, the parametrized generation is no longer assigned to a dedicated situation for which one can create an ARtS. Instead, ARtS and generated scenarios have to be matched. Since the focused scenario is described with parameters, a proper ARtS for a given generated scenario is chosen. Therefore, it has to be considered that the ARtS behavior should be compatible with the generated scenario. For this, a similarity analysis between scenarios has to be performed. For simplification, this is done on predicted trajectories.

The similarity analysis of the scenarios is based on two points of view: a top-down approach and a bottom-up approach. The top-down approach considers the general and relative relationships of elements within the scenario while the bottom-up approach considers the similarity based predictive trajectories.

The top-down approach adopts the annotation of base scenarios, so that constellations can be compared in the context of the underlying scenario concept. It considers the relative description between scenario elements in a hierarchical way: At the lowest level, base scenarios are compared with each other matching the individual concepts with suitable weighting. For a longer sequence of base scenarios between ego and one other vehicle dynamic time warping[9] is used. For this purpose, time is discretized, and the base scenario is annotated to the time stamps for each ego-object relation. Comparing different sequences of base scenarios, costs can be calculated based on the individual comparison and evaluated over time using dynamic time warping. If multiple relevant road users occur in the scenario, the procedure can be performed for all, and the normalized combination can be used as a comparison score for similarity. Thereby, potential weighting due to importance of relations may be included. So, abstract scenario sequence can be compared.

In the bottom-up approach, the determined relations between the road users are not considered explicitly. The approach focuses purely on raw trajectories. These are compared with the corresponding similar trajectory of the other scenario using dynamic time warping and Fréchet distance. Anyhow, further scenario elements such as infrastructure parameters can be added to the similarity comparison using additional weighting.

Performing both, bottom-up and top-down approach of the similarity analysis, scenarios can be classified in three categories: similar, dissimilar, and distinct (see Figure 14).

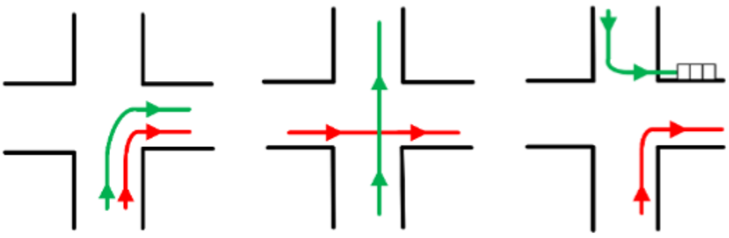


Figure 14: Types of similarity

A combination of ARtS and parameterization is possible in two cases: If the scenarios are similar, parametrized road users can be deleted within the ARtS. The remaining road users then react to the similar parameterized trajectories. Another technical possibility of superposition arises if the scenarios are independent of each other. In this case, no road user has to be deleted, but the benefit is doubtful since the road users do not interact significantly with each other. In the dissimilar classification, such a combination is not possible. However, in both positive cases, no one to one relationship between ARtS and parameterized scenario is assumed. Accordingly, it is conceivable that many matching ARtS can be found for a parameterized scenario, each of which provides a valid background.

2.3 Logical Scenario Classes

Logical scenario classes are a concretization of abstract scenarios. Within those, abstract scenarios are enriched with parameter and attribute declarations. Those scenario classes structure the scenario space and can be understood as a blueprint for logical scenario instances (see Sec. 2.4) and afterwards concrete scenarios. So, parameter design of scenarios and the character of the scenario is defined within the scenario class. Therefore, attributes, parameters and their relations are defined within the logical scenario classes. The definition is structured along the six layers.

---

[9] M. MÜLLER, “Dynamic time warping. Information retrieval for music and motion”, 2007, S. 69-84

Whereas most layers allow a nearly independent and simple parametrization and attribution, layer 4 has a potential of variety and complexity and is particularly focused in the following.

2.3.1 Infrastructure Parametrization

Based on the abstract scenario concept, parameters and attributes are defined to concretize them to logical scenario classes. While the parameters on layers 1-3 (Sec. 2.2.3.1) as well as layer 4 (this chapter) require special explanation, Layer 5 (Sec. 2.2.3.5) and Layer 6 are orthogonal to them. They may still have an influence, but this is reflected in the distributions and relations within the logical scenario instances (see Sec. 2.4).

2.3.2 Base Scenarios[10]

The attribution and parametrization of base scenarios must be done systematically. Due to the potential amount of different base scenarios, attribution and parametrization for each scenario individually would lead to a high workload and potential inconsistencies. So, the strictly hierarchical structure of the scenarios is used for formal derivation of attributes and parameters of base scenarios. Due to the ontology, the attributes and parameters result directly from the definition of the characteristics of the concepts. Defined concept attributes and parameters are super positioned in base scenarios (see Figure 15). While this can be done without any restrictions for attributes, this process implies requirements for the parameterization, since these must be independent by definition or must not contradict each other to guarantee a conflict free and meaningful generation.

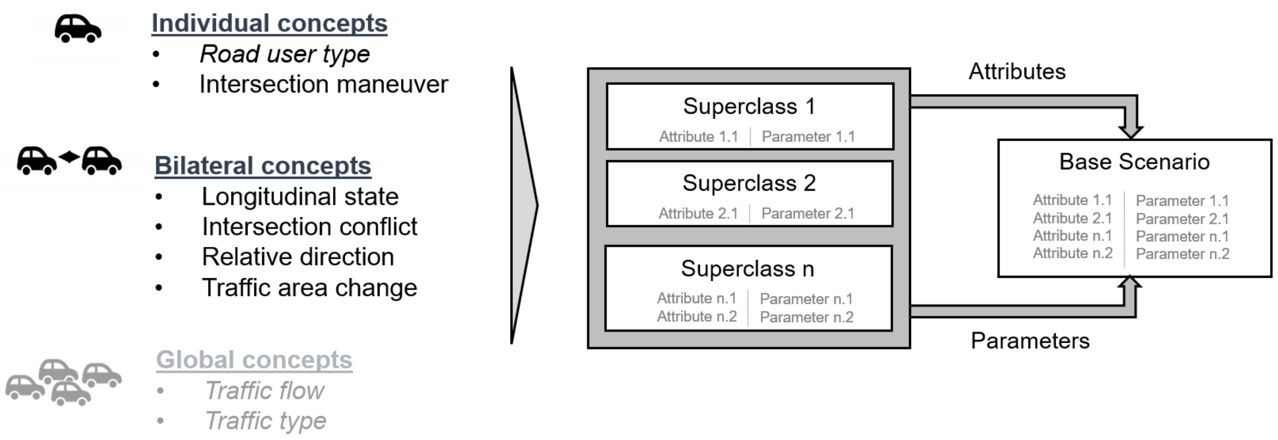


Figure 15: Hierarchical parameter declaration from concepts to base scenarios

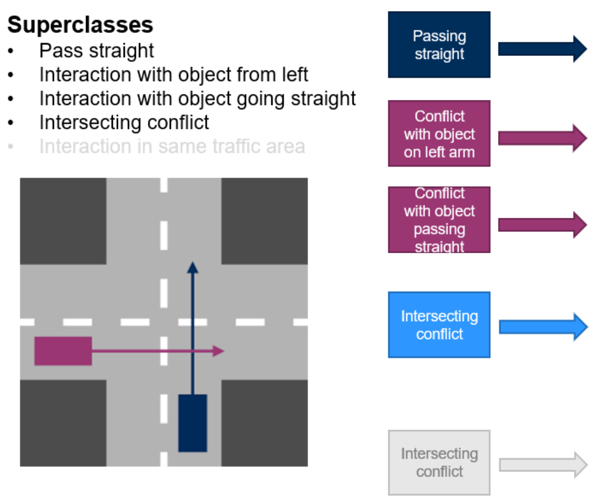
Using this kind of parameter and attribute derivation, a base scenario can be described comprehensively in a systematic way. Thereby, both, parameters and attributes focus on the core characteristics of the base scenarios respectively its underlying concept.

If e.g., a conflict is modeled in a base scenario, the main characteristics of the conflict are reflected in attributes and parameters. Those then include the description of the conflict zone, the behavior of

---

[10] Based on H. Weber, C. Glasmacher, M. Schuldes, N. Wagener and L. Eckstein "Holistic Driving Scenario Concept for Urban Traffic", Intelligent Vehicle Symposium, Anchorage, 2023.

road users in the conflict zone and the relation of road users at this conflict zone. Furthermore, for parametrization, approaching the conflict zone is particularly interesting for generation. Since there are multiple ways to model it, constraints and probabilistic relations are set additionally to other concept parametrizations to ensure a robust and realistic generation process. The constraints considered thereby depend on the purpose the scenario is created for, on mathematical dependencies and probabilistic causal relations between parameters (e.g., due to behavior of road users). Similarly to the conflict, considerations hold true for other concepts so that the amount of all concepts gives a comprehensive picture (see Figure 16).



Attribute	Parameter
Pass Straight • Ego Velocity*, Ego lon. acceleration*, …	Pass Straight • Initial ego velocity • Initial ego position (s, t, road, lane) • Ego destination (road, lane)
Interaction with object from left • Obj. Velocity*, Obj. Lon. acc* • Relative heading* • Initial intersection arm	Interaction with object from left • Initial object velocity • Initial object position (s, t, road, lane)
Interaction with object going straight • Final intersection arm	Interaction with object going straight • Final object velocity • Final object position (s, t, road, lane)

Intersecting conflict
• TTC, PET, predicted time gap
• Conflict angle, velocity
• Priority Level
• Conflict point
	Intersecting conflict
	• Velocity of object at point based on:
	• Constraining parameters:
	  • Predicted timegap @ start
	  • Predicted Priority Level
	  • Predicted Conflict Point

Interaction in same traffic area • Prior traffic area type • Traffic area type	Interaction in same traffic area • Point of changing traffic area • Velocity at point

Figure 16: Concrete parametrization of base scenario

Thereby, the parametrization within each concept describes the concept itself sufficiently. That can lead to overlaps or additional constraints between the parametrizations of each concept. Therefore, constraints and probabilistic relations are established by and emerge from the combination of those concepts (see Figure 17, left) and are described according to Sec. 2.3.4. Additional constraints and relations have to be set. E.g., if two concepts describe the velocity of a certain road user at different timestamps at least a probabilistic relation between those different velocities should be modeled to ensure a proper causal behavior chain of the road user (see Figure 17, right).

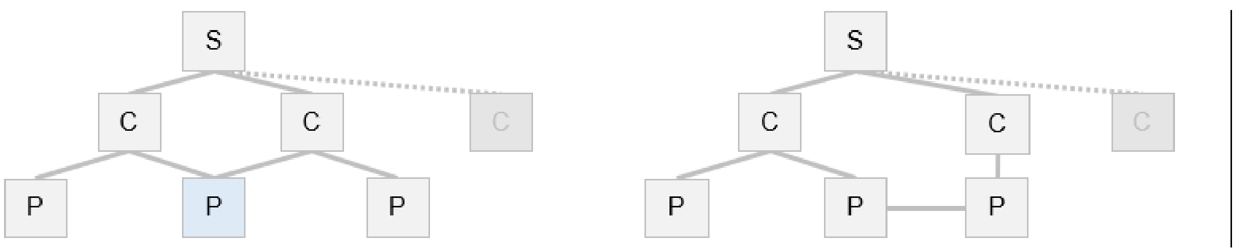


Figure 17: Graph based relations to prevent inconsistency due to multiple parameterization (left) and dependencies (right) between parameters (P) derived from scenario classes (C) within a scenario (S).

Besides dependencies between existing parameters and attributes of concepts, emergent properties can arise from the combination of different concepts. Those have to be described directly in the ontology since they arise from the combination of those concepts. To structure parameters and attributes efficiently, both are structured within an ontology to allow different abstraction layers of parametrization depending on the use case.

Since there is not only one correct parameterization, as an example an abstraction is used based on a trajectory reconstruction using Cubic Hermite Splines as shown in Figure 16. Cubic Hermite Splines thereby sets a spline not only based on two dimensional control points, but considers the direction and velocity within those points. This allows a relatively robust generation by using control points. In this way, characteristic points are derived from each of the concepts, based on their core elements. For example, conflicts are defined by the characteristic point entering the conflict zone. The maneuvers are defined by points at the entry and exit of the intersection. The four-dimensional points are defined in each case in the Frenet space. This space is locally defined and follows curvatures of road and lanes, so that is makes use of the infrastructure definition for an easier representation.

To ensure that the parameters and concepts established represent traffic, trajectories of the inD dataset[11] were annotated within the enveloping scenarios and the parameters were extracted. The reconstruction using Cubic Hermite Splines was compared to the original trajectories. This shows a small deviation in most of the trajectories (see Table 1). Regardless of this, the parameterization ensures that the region of interest in particular is mapped realistically (see Table 1, Figure 18).

Table 1: Comparison of real and reconstructed trajectories on inD dataset

Metric	Average deviation	90% Quantile	99% Quantile
RMSE velocity [m/s]	0.1443	0.2273	0.3697
RMSE position [m]	0.4940	0.8097	1.0169

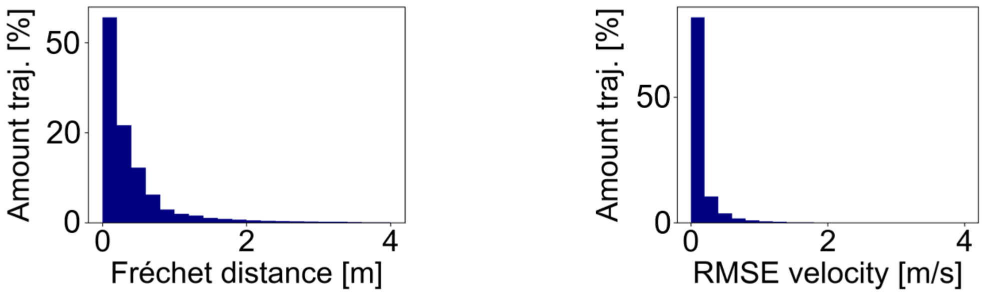


Figure 18: Histogram of root mean squared error between reconstructed and real-world trajectories within the inD dataset.

However, different parameterizations from open to closed loop are also conceivable within the proposed method. The parametrization depends on the purpose the scenario is used for and the availability of data. For example, the speed control can be set adaptively by acceleration values extracted via the distribution function of the real accelerations driven. Another possibility is a fully parameterized driver model that responds completely to the ego. Therefore, a problem is the missing but necessary knowledge of a comprehensive parameterizable driver model.

[11] https://levelxdata.com/ind-dataset/

2.3.3 Combined Scenarios and Scenario Modifications

Similar to the base scenarios, the parameters of the combined scenarios are derived hierarchically from the concepts. However, a simple superposition is only possible in very few cases as the proposed control points from Sec. 2.3.2. It is also not necessarily meaningful for all combined scenarios since emergent properties cannot be considered when simply super position individual parameters. Therefore, depending on the type of combined scenario, a case distinction must be made, and parameterization has to be adapted. That can include the neglection or restructuring of parameters as well as adding new parameters because of emergent properties. Therefore, a distinction is made between adaptation, sequential and parallel linking of basic scenarios. The parametrization is thereby still related to the base scenario parametrization to link those and to give and use the advantage of hierarchical derivation of parameters and concept description.

2.3.3.1 Sequential combination

A sequential combination involves the temporal combination of at least two base scenarios with a time gap of $\Delta t \geq 0$ (see Figure 19). A simple superposition of the parameters of the two base scenarios under the synchronization to a common start time is only possible with an independent choice of parameters, since the ego vehicle is not controlled in the scenario itself and can decide itself how it behaves.

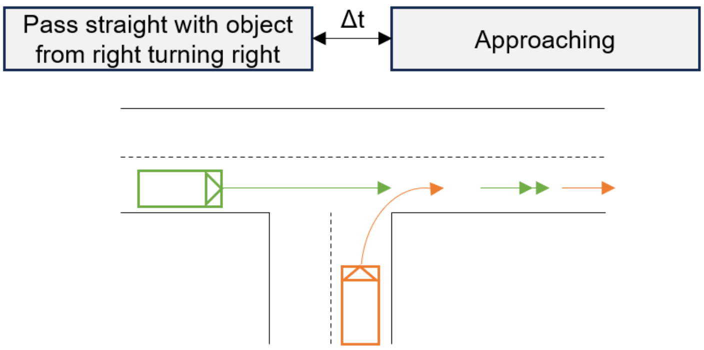


Figure 19: Sequential combined scenario with intra dependency in a relation between objects

This affects the parameterization of both scenarios but becomes more relevant the longer the scenario lasts since deviations can increase and conditions of the second scenario are not regarded anymore.

The first one can be modeled relatively independently, since there are no further influences starting the scenario. For one or multiple following, the independent setting of initial conditions without any restrictions is not possible due to changes in ego behavior. So, the parametrization of the second scenario must fit to the first scenario. Two cases must be distinguished:

- In case of a direct temporal connection, if parametrizing start conditions of the following base scenario, the restriction must be set such that the states correspond to the end conditions of the previous base scenario. This is not possible for parameters that are influenced directly or indirectly by the system under test. Those could change according to the ego behavior and therefore would cause a conflict or jump in velocities and positions. So, it is preferable to not force starting conditions as the first scenario, but to set a target state to reach within a control problem.

- In the case of a time gap between base scenarios, a transition phase between scenarios can be established to meet the conditions of the scenario. Thus, starting points can be set under the constraints of a physically reasonable control. Still, this should be done relative to the ego vehicle since the deviation of the ego vehicle at the starting point of the second scenario may be significant.

However, depending on the parameterization of the scenario, it cannot be ensured that the desired sequence is seen in the simulation. This is because although the scenarios for the situations are designed, the behavior of the ego may differ. For example, if the ego in Figure 19 accelerates depending on the start conditions in such a way that it crosses the intersection before the object, an approaching would no longer be observed. For longer sequences, this problem is amplified. This still has to be allowed since the ego vehicle behavior should be tested. This can include avoiding certain situations. A possibility to face this issue is an adaptive parametrization and scenario modeling can be used. Since distributions for parameters including the ego state are included in the logical scenario instances, parameters of consecutive scenarios may be adjusted according to the ego state at runtime. If e.g., the ego decelerates, the distribution for velocities or decelerations of the object vehicle may be tilted towards lower speeds than initially expected, so that this parameter may be adapted while the system is tested in the scenario.

2.3.3.2 Parallel combination

Similar to the sequential combination of base scenarios, a parallel combination of base scenarios is only possible with certain parameterizations. One possibility is the direct description of predefined trajectories according to the originally recorded scenario. However, this is only possible because these trajectories already implicitly contain further information such as the interaction of the other two vehicles.

Deviating descriptions require further parameters or constraints depending on the constellation of the scenario to handle emergent characteristics. Necessary distinctions to handle those dependencies are therefore made based on a potential spatial overlap and temporal offset:

Spatial inter dependencies between base scenarios

If not only one road user constellation between two road users is considered but two or more different should be modelled, emergent characteristics have to be taken into account. Especially in the case of spatial overlap, a separate or sequential consideration of the individual base scenarios is no longer possible. Beside the relations already described in the base scenarios, dependencies between other non-ego road users have to be considered within the creation and parameterization. In the case of Figure 20 both the orange vehicle and the blue vulnerable road user could be described by the base scenario intersecting conflict. Since the interaction between blue and orange is not focused, an extensive description between those is not necessary. However, conflicts and unrealistic behavior which affects the ego road user must be avoided. To account for this, the addition of parameters to describe the conflict zone as well as the object that should cross it first becomes necessary for the description.

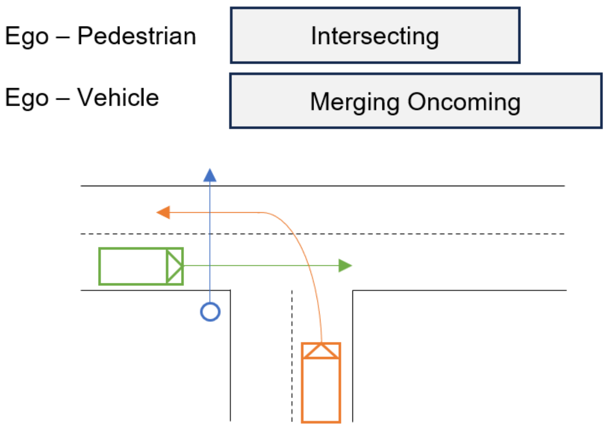


Figure 20: Parallel Combined scenario with temporal and spatial influence

The description of the dependency depends on the combination of the base scenarios. For systematization, this can be modeled in an ontology like the parameters themselves.

Temporal inter dependencies between base scenarios

If an ego along with two other road users should be modelled in a scenario, two base scenarios can have a temporal offset, so that they occur but do not start at the same time. Therefore, similar considerations have to be made compared to the sequential description (see Sec. 2.3.3.1). The ego cannot be described again in the second scenario but behaves according to the influence of the system under test. Nevertheless, the initial state of the second object must be set. However, the synchronization to the ego is decisive for the generation of the desired scenario. To achieve the scenario in the best possible way, the choice depends on several factors. These are the current state of the first base scenario, the relation to the infrastructure as well as the temporal course of the scenario. Depending on the objective, it is therefore possible to synchronize the start of the second base scenario according to the location of the ego, the time since the start of the simulation or the start of the first scenario as in the ARtS (see Sec. 2.2.2). Due to the localized structure of intersections, local synchronization is chosen for intersection conflicts. However, especially in stationary longitudinal traffic a deviating synchronization can be useful.

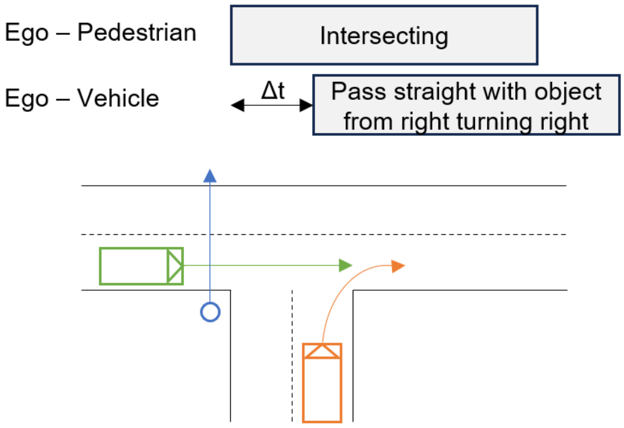


Figure 21: Parallel Combined scenario with only temporal aspects

2.3.3.3 Adaption of Base Scenarios

As stated in Sec. 2.3.2, different parametrizations are possible within base scenarios. Different parametrizations may be necessary since the respective purpose of the use of the scenarios can be different and must be parameterized differently accordingly. The parameterization depends on the statement to be made by the test, but also on the system to be tested.

If, for example, the avoidance of a conflict with another road user is to be tested, the parameters must be set differently than when testing the passing of children at the roadside. In the case of the children, if the children are not paying attention, it can be assumed that they are unaffected by the ego road user and, accordingly, can be modeled independently. In the case of conflict, different parameterizations can be useful depending on the system to be tested. For example, if the avoidance of a conflict is to be tested by means of a predictive system, the parameters must be set differently than for a system with low prediction time, since the predictive system must be given the chance to solve the conflict beforehand. Also, parameters like color of the road user are needed if a camera is used, but not when only a lidar is. The selected parameterization of the base scenarios, thereby, offers a tradeoff, which has to be chosen depending on the application.

In the case of the challenger scenarios, for example, a deviating parameterization of the velocity may be appropriate. While the path is kept constant, the challenger provokes a collision by adapting its speed to that of the ego vehicle within the framework of recorded limit values for acceleration and speed. With such a parameterization, however, it must be ensured that the original behavior is not too alienated in order to continue to be able to make statements about the probability in the sense of a risk-balance calculation.

2.3.4 Modeling of Parameter Dependencies[12]

In addition to the declaration of the parameters, relations between parameters are assigned in logical scenario classes and are reflected in the parameter ontology. Those relations can be represented by simple distribution functions like copulas[13]. Thereby, correlations between all parameters would be considered at the same time. Consequently, a significant amount of data is needed to generate an adequate fit in high-dimensional parameter spaces. Therefore, an alternative modeling is used, combining causal probabilistic edges with constraint edges. In spite of usual multivariate distribution functions as copulas, only causal and physically meaningful relations are described to narrow down the needed amount of data for fitting. Furthermore, the description of only meaningful relationships prevents spurious correlations.

To describe causal relations between parameters which cannot been described easily with constraints, a probabilistic model is used: causal Bayesian edges are used (see Figure 22).

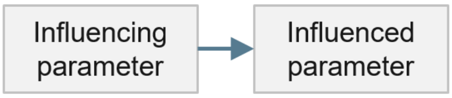


Figure 22: Causal relation between parameters

Parameters which are modeled with those edges can be related to behavior sequences of road users, but also include e.g., infrastructure influences.

In addition to the representation of causal relationships, however, there are also constraints that do not directly represent causal relationships but must hold in a valid scenario. These can include mathematical relationships or physical constraints. In addition, design decisions can be modeled as constraints. However, these must be consistent with the scenario concept used and must not unjustifiably constrain the space. For modeling, the graph structure of a constraints graph is used[14] and stored within an ontology. A distinction can be made between constraints that are spanned between several parameters and those that refer to one parameter (see Figure 23).

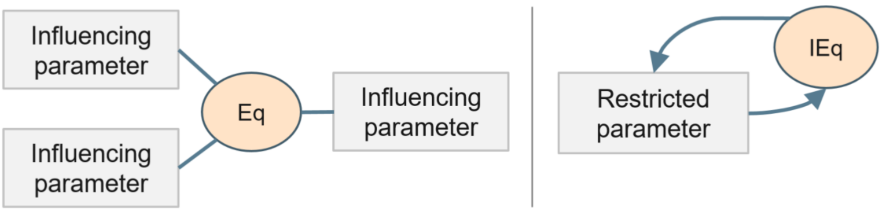


Figure 23: Constraints between parameters

Typical parameters described by such constraints can be prerequisites of location to ensure the existence of a scenario. Another example is ensuring the correctness of mathematical equations for criticality metrics for dependencies within the parametrization. Furthermore, in combination with

---

[12] Based on C. Glasmacher, H. Weber, M. Schuldes, N. Wagener and L. Eckstein "Generation of Concrete Parameters from Logical Urban Driving Scenarios Based on Hybrid Graphs", VEHITS, Prague, 2023.
[13] K. Lotto, T. Nagler, M. Radic "Modelling stochastic data using copulas for application in validation of autonomous driving" 2021
[14] G. Friedman. P. Phan. "Constraint Theory", volume 23. Springer International Publishing 2017

combined scenarios, temporal logics can be modeled with these constraints if there are temporal dependencies between parameters.

The advantage of this graph-based modeling over an independent list is the easier understanding of a visualization, as well as modeling complex relationships within a scenario. Furthermore, characteristics of causal Bayesian networks as well as constraint graphs can be used. So, e.g., adjustment sets can be calculated which builds a valid set of parameters within a scenario. Using this analysis also allows a dependent parametrization unless one valid set is used to create OpenSCENARIO files. Furthermore, the description of influences becomes easier describable e.g., using the do operator.[15]

2.4 Logical Scenario Instances

Logical scenario instances are logical scenario classes filled with values and distributions. As such, they represent the traffic space within the abstracted and knowledge-based parameter space of the scenario. In addition to the set parameters, the predefined relations are also fitted to constrain the multidimensional parameter spaces easier using knowledge-based information. Although that allows a description which needs less data for fitting, it has to be elaborated how to calculate probability distributions (see Sec. 2.4.1) and how sampling is possible out of those logical scenario instances (see Sec. 2.4.2).

2.4.1 Estimating Distributions

An essential requirement for the construct of logical scenario instances is the determination of occurrence probabilities of parameter constellations in real traffic under certain conditions. This conditional probability is determined by the definition of the parameter relations to each other. Therefore, the graph structure established in Sec. 2.3.4 is used. The graph structure coming from the logical scenario class is filled with probabilities by the current values.

Due to the differentiation in causal relations and constraints, two types must be distinguished:

- Directed edges denote causal relations and can be treated as edges of a Bayesian network. As such, the conditional probabilities can be determined.

- Constraint edges cannot be fitted but are always binary. Provided that the constraint is fulfilled, the probability is 1. However, if it is not satisfied, the value drops to 0 and the total probability of the network is 0. Thus, provided that the network is filled with valid input data and the set for probability calculation is valid, the constraints can be neglected for the calculation. This is not the case for newly generated parameter sets. For these, the constraints must be checked. If a constraint is not fulfilled, the probability drops to 0 here as well.

Once probabilities are calculated for each edge within the network, the probability of the whole scenario can be calculated as the joint probability P for n causal Bayesian nodes (X) and m constraint edges (C) (see Equation Equation 1). A constraint is thereby fulfilled, if the mathematical representation holds true for given parameters.

---

[15] J. Pearl, S. Russell, "Bayesian networks" UCLA, Department of Statistics Papers 2000.

Equation 1: Probability estimation network

$$(X_1,\ldots,X_n) \; = \prod_{i=1}^{n} P(\,X_i \mid Parent(X_i)) * \prod_{j=1}^{m} \begin{cases} 1, & C_j \; is \; fulfilled \\ 0, & else \end{cases}$$

2.4.2 Sampling from Logical Scenario Instances

In order to create concrete scenarios from the logical scenario instances, it is necessary to sample from the logical scenario instance. While a discussion about sampling methods is not part of this deliverable, the handling of the defined logical scenario instance consisting of parameters and their relations as described in Sec. 2.2.3 is discussed below. Generally, different methods are conceivable to sample parameters from the previously defined graph structure: Analytic solution, brute force from multi variate distribution or initial solution via multivariate distribution in combination with necessary adjustments.

However, since the analytical solution can only be calculated for simple systems[16] and the brute force method is a relatively unstructured and slow method due to a large amount of invalid generated scenarios, a combined approach is chosen: Thus, based on an initial copula solution with implicit consideration of the constraints, an initial sampling is done. Based on this, the constraints are iteratively checked, and parameters are adjusted accordingly. Finally, to ensure that the solution is still valid, the probability is calculated according to Sec. 2.4.1.

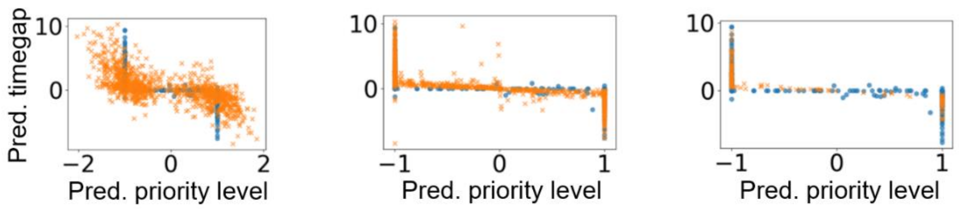


Figure 24: Sampling from logical scenario instance with real data (blue) and generated data (orange)

This solution offers the possibility to represent and sample even high-dimensional parameter spaces with a limited amount of input data (see Figure 24). While a multivariate copula is purely data-based and does not consider limitations (left) and a simple adaptation of the parameters does not correspond to the original distribution (middle), the combined approach offers the possibility of a realistic and physically reasonable parameter distribution even from high-dimensional parameter spaces (right).

2.5 Scenario Concept Completeness and Coverage

For the safety assurance of automated driving, the importance of completeness or complete or sufficiently complete coverage is a relevant research topic. As those scenarios are stored in a database, the database itself should give hints about the completeness of included scenarios as a significant quality measure. Therefore, three types are identified as relevant for a database and underlying concepts. Those are split in completeness and coverage. Completeness is the full

---

[16] According to Friedman, G. J. and Phan, P. (2017). Constraint Theory, volume 23. Springer International Publishing, Cham.

representation of relevant concepts and is generally understood in binary terms. For example, a system is completely defined if no component is not defined. Accordingly, completeness itself can be applied primarily to systems and concepts, but not to unknown parameter ranges. Coverage is introduced as a counterpart to this. It describes the degree of coverage of a population. In contrast to completeness, it is not binary, but contains a continuous range of values.

For the reality abstraction and the collection of scenarios in a database, three types are to be distinguished:

- Completeness of the scenario concept
- Completeness of scenario parametrization
- Coverage of scenarios in the database

In addition, further completeness and coverages can be defined, such as the completeness of the test selection or the sampling. Since these go beyond the scope of the database, they are not considered further below. The three relevant concepts for a database will be covered in the following sections.

2.5.1 Completeness of Scenario Concept

As a basis for the storage of scenarios and reality abstraction, the completeness of the scenario concept must be investigated. A scenario concept has to be sufficiently complete to cover all aspects to structure traffic in the respective ODD which are relevant for the system to be tested. So, the completeness depends on both, the system under test for which the completeness should be proven and its respective ODD. If e.g., a concept shall only cover highway scenarios and the respective driving function should only be confronted with such, the scenario concept may be complete for this use case. If such a use case is not defined a priori and the ODD is not clear, the completeness would have to be tested to the maximum needed. If a new function exceeds those limits, the completeness has to be tested again for these extended requirements.

To check the completeness of a scenario concept with regards to a function and ODD systematically, a Goal Structuring Notation (GSN) -based argumentation structure based on concerns is established. This structure approaches the problem of completeness in two ways: on the one hand, the scenario concept must be completely defined, and on the other hand, the definitions must comprehensively cover the ODD and elements needed for a potential system under test.

For the proof of the complete definition, the scenario concept is broken down into trivial definitions of the components of the concept. For this purpose, an alternating structure of claims and argumentations is established (see Figure 25). In addition to that, examples can be given to explain the underlying elements and to make the structure more understandable for stakeholders.

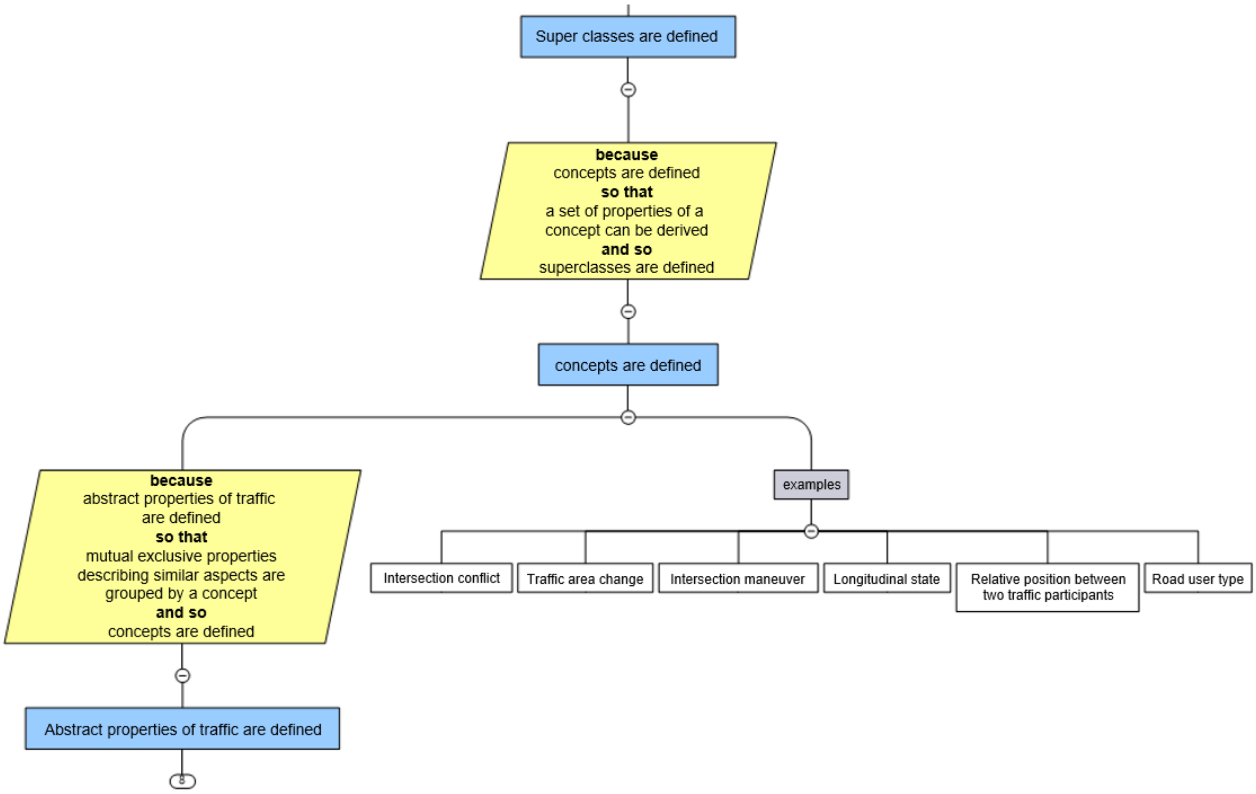


Figure 25: Part of GSN argumentation structure from super class definition to abstract properties of traffic

For the argumentation of the comprehensive coverage, on the other hand, a top-down and a bottom-up approach is pursued. In each case, the coverage of the concept must be argued to support the claim of sufficiently complete coverage. Thereby, this can be proven by three ways:

- Data can be used to prove that no more relevant element is found in the real world.
- It can be shown combinatorically that there is no other element or state possible which would have to be considered.
- It can be argued that another state is either not in or not reasonable for reasons specific to the concept.

With the proof that the concept is defined completely and the elements cover the ODD sufficiently, the argumentation can be made that the scenario concept is sufficiently complete.

2.5.2 Completeness of Parametrization

A parameterization represents the abstraction of reality into a simplified model for an abstract scenario. Accordingly, the question of the completeness of the parameterization is related to the completeness of the logical scenario classes within the concept. This question goes along with the level of detail a parametrization should include for a sufficiently defined scenario.

However, this question cannot be answered without considering the purpose of the model and the intended usage, since a model always represents the world only sufficiently for a certain context. This context is determined by the effects that have a direct or indirect influence on the system under test. Direct influences can be the speed of a vehicle in front, which causes the ego to brake. Indirect influences, on the other hand, have an impact on other components in the scenario, such as the speed of the vehicle driving in front of the one just described.

Accordingly, parameterization is sufficient if the concepts of the base scenarios describe them in a sufficient level of detail for the system under test and further detailing does not lead to any relevant deviation in the output of the scenario.

Since there is no specific driving function observed in the project, the answer cannot be given comprehensively here. For further validation we assume that a potential system under test only reacts on a trajectory of other vehicles which do not imply a significant counter action which is relevant for the outcome of the scenario. This may be especially the case if the interaction of the vehicles with the ego vehicle is rather small in terms of intensity, time and/or impact. Under those assumptions, the completeness of parametrization can be shown by comparing original trajectories with trajectories reconstructed by derived parametrizations (see Figure 26). If the deviations between those are small as proven for the inD dataset and the presented scenario concept, it can be argued that the parametrization is sufficient for the presented assumptions.

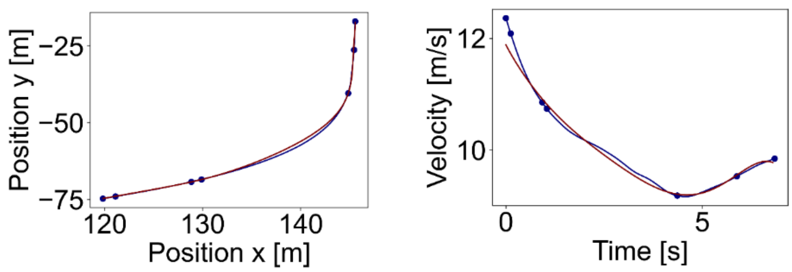


Figure 26: Reconstruction of trajectories (blue) from parameters (points) from real trajectories (red) using Cubic Hermite Splines

If the assumptions with regards to the trajectory parametrization do not hold because of a different purpose of the concept or behavior of the system under test, the proof of complete parametrization has to be done differently. Potential opportunities are simulations and comparing the behavior to realistic situations. Another option is to vary the parametrization and observe the influence in the outcome of a simulation. If the deviation is not relevant for the assessment of the simulation or deviations can be controlled, argued, and corrected, the parametrization is sufficient.

2.5.3 Coverage[17]

The coverage of the scenario space and the subsequent derivation of concrete scenarios form the third pillar. While the first two consider completeness on a conceptual level, the third focuses on the coverage of filled values and representation of the real-world according to the boundaries of a scenario concept. Accordingly, logical scenario classes are assumed as defined and based on this, it is checked to what extent the logical scenario instances sufficiently represent the traffic.

[17] Based on C. Glasmacher, M. Schuldes, H. Weber, N. Wagener and L. Eckstein "Acquire driving scenarios efficiently: A Framework for Prospective Assessment of Cost-Optimal Scenario Acquisition", Intelligent Transportation Systems Conference, Bilbao, 2023.

Practically, this means to what extent the real limits of individual parameters such as initial speeds, abstract constellations or positions correspond to those of the real traffic (Equation 2). The relative coverage of the space is therefore defined as a fraction of the covered area with reasonable scenarios by the approximated population of the occurring scenarios.

Equation 2: Relative coverage of a data set compared to the real world

$$Relative\ coverage\ = \frac{Coverage\ of\ data\ set}{Approximated\ coverage\ of\ real\ world\ set}$$

The coverage can thereby be separated into two hierarchical layers:

- Coverage of abstract concepts in logical scenario instances
- Coverage of parameter ranges within logical scenario instances

On the first layer, it should be considered if a given set of scenarios contains all abstract constellations. Although the theoretical number of base scenarios can be calculated, it is not clear that all base scenarios can be found in a given ODD in real-world scenarios. An even bigger deviation between theoretical and real-world amount of different found abstract scenarios can be arise when having a look at combined scenarios since those can combine different base scenarios to sequences.

On the second layer, the parametrization within given scenario classes is considered. This can include discrete variables such as road user types (pedestrian, bicycle, e-scooter, car, etc.) or continuous variables such as vehicle speeds or locations. Compared to the first layer, a theoretical maximum of different characteristics cannot be set, but just be approximated.

In order to perform calculations to assess the coverage the covered set has to be assessed and the occurrence in real-world has to be approximated. Thereby, a distinction has to be made between discrete and continuous variables. To use both in a calculation, a volume within a multidimensional parameter space is calculated to approximate each coverage (see Figure 27). Since a volume cannot be calculated for a point, it is assumed that the space around each point is covered by the point approximating an ellipsoid for continuous variables and fixed values for discrete variables. The semi-axes of the ellipsoid and the range for the discrete variables has thereby to be set according to the sensitivity for a further use case. With this definition, the coverage can be defined as the union of covered space of the scenarios.

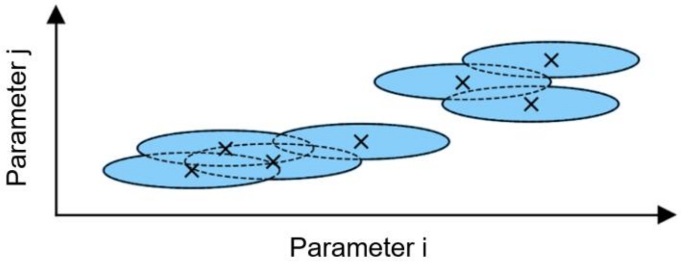


Figure 27: Coverage of points in a parameter space

Another core component for estimating the relative coverage is the determination of the approximated real-world coverage. However, since infinite amount of data cannot not be available, the potential occurrence in real-world is approximated via a bootstrapping method by combining the evolvement of the total volume of coverage over the number of unified scenarios. Fitted to a saturation function, it can be shown that this can reflect the input data and the parameters converge. Thus, the limit value of the saturation function can be approximated (see Figure 28). The underlying assumption that the occurrence of new scenarios is random can be supported by randomizing the input. Still, scenarios that are significantly rarer than others are included implicitly due to the saturation behavior of the function.

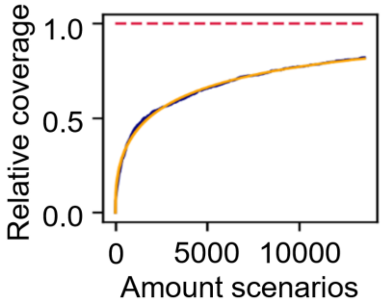


Figure 28: Approximation of real coverage (blue) by function (yellow) for continuous parameters with approximated limit (red)

This is possible for both levels, scenario classes and parametrization within. If all variables are discrete, the coverage approximation can be simplified to the coverage of discrete buckets and no volume has to be calculated. The fitting of the saturation function thereby stays the same (see Figure 29).

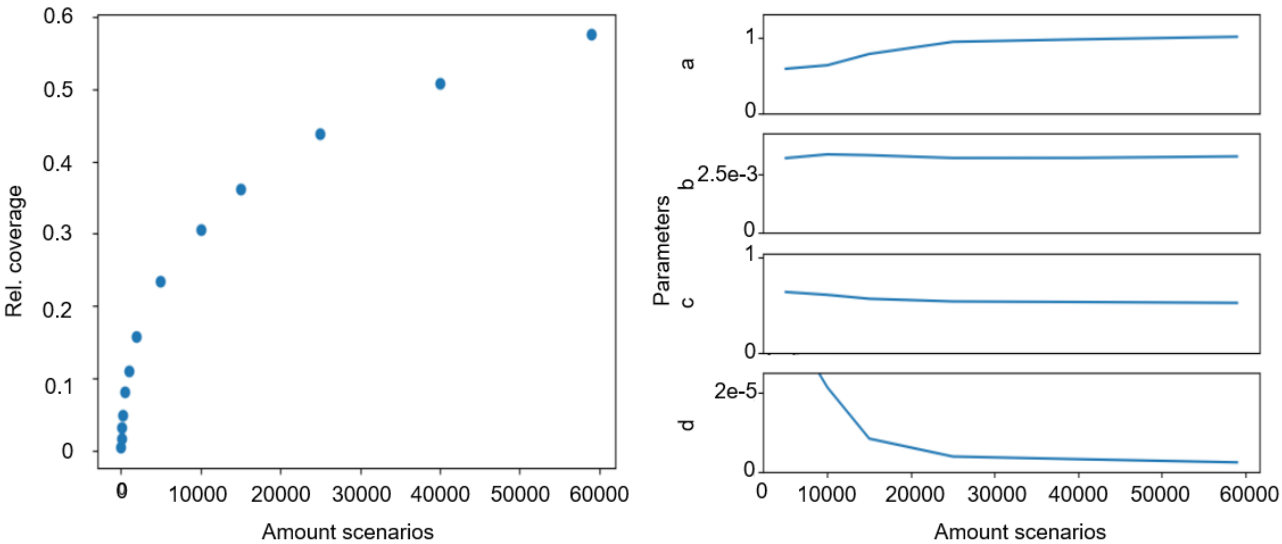


Figure 29: Coverage approximation for logical scenario classes and their converging parameters

However, since this approach is based purely on the recorded input data, biases must be taken into account. For example, if only data of one intersection during daytime are recorded for the approximation, the estimation is only valid for this intersection during daytime. Therefore, further methods to validate the input data should be considered.

The proposed methodology of coverage and coverage approximation is additionally extended to include distributions and the convergence of scenario probabilities. Therefore, not only the union of volumes is used, but the probability at each point is calculated overlaying those volumes. Evaluating the difference between probabilities on different number of scenarios cumulated difference of probabilities can be derived for each pair of different scenario amounts. The difference can thereby be approximated by a saturation function approaching zero, so that difference in probabilities can be approximated over the amount of input scenarios as for the coverage. Nevertheless, significantly more data is needed for this approximation for a convergence of parameters since the probabilities are a new degree of freedom added.

3 Scenario Data Flow

The previous chapter described the methodology for acquiring logical scenario instances for the validation and verification of ADS. In this chapter, the data process required to apply the methods is described. The process is depicted in Figure 30.

The first step is the acquisition of reference data (Sec. 3.2). Real-world driving data is recorded and stored in a raw database. Such raw data includes video and lidar point clouds from the vehicle sensors. It is important that data is secured, and access is restricted because of potential leakage of privacy-sensitive information. These restrictions are managed by the Safety Cage (Sec. 3.3). To label the raw data, the video information is first anonymized. This anonymized data can then be labeled to get reference data.

Based on the reference data, scenarios are extracted. To be compatible with different data sources, an input interface is defined in the form of a data format for object-list-based trajectory data and map information, the OMEGA-format (Sec. 3.4). The reference data in the form of OMEGA-format files is uploaded into the scenario database and stored for traceability in the input database. Then automated algorithms are triggered, that identify scenarios in the data, compute parameters and attributes for these scenarios and store them in the scenario database. This is performed for all available data. Through a query interface, a for a use case relevant subset of scenarios can be selected. This is a set of logical scenarios. Parallel to the scenario identification in reference data, a perception database can analyze object list data from sensors under test and compare them with the reference data in the scenario database. These can additionally be used to identify the relevant subsets of the scenario space. Through the mechanics of the database, these logical scenarios can be translated into simulation instructions like OpenSCENARIO, which is used to define test cases and perform the loop to evaluate ADS.

Already during the development of the above-described systems and methodologies, there is the need to exchange data between involved parties in a reliable and traceable manner. This task can be fulfilled by a development database. Such a database is described in Sec. 3.1.

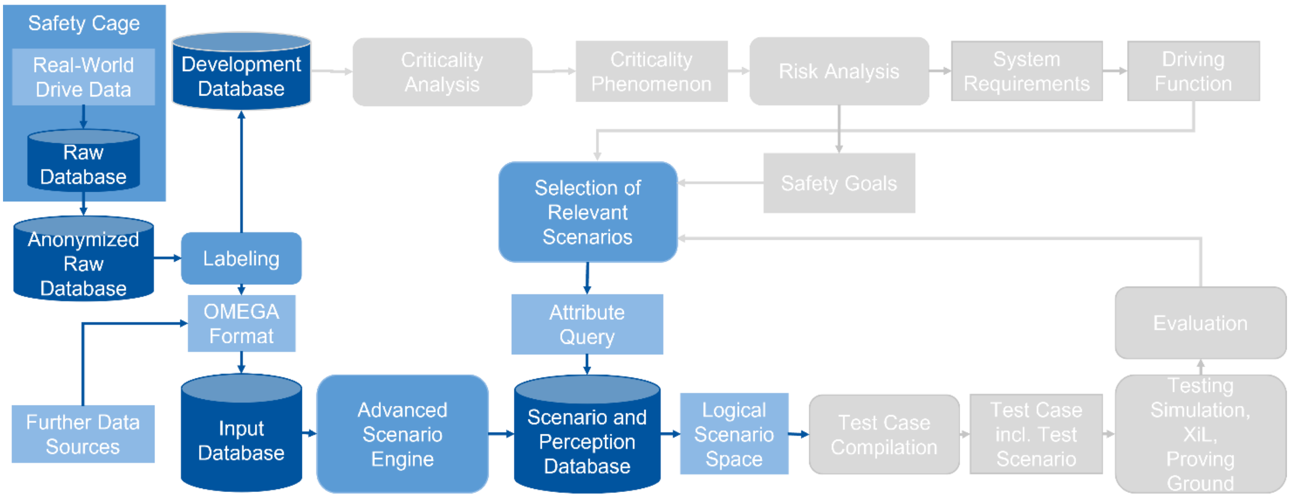


Figure 30: Data flow overview

3.1 Development Database

Coscine, developed by RWTH Aachen University, is a leader in FAIR Research Data Management (RDM), offering a multifaceted platform for data storage, sharing, and archiving. As an open-source and community-driven tool, Coscine continuously adapts and evolves, ensuring it meets the needs of its users. Designed to meet the evolving needs of the scientific community, it is not merely a data repository, but a holistic solution for managing the entire lifecycle of research data. Its role as a tool and development database in the VVM project highlights its effectiveness in handling complex data sets and metadata in accordance with established scientific standards. Throughout the project, Coscine has enhanced its functionalities, evolving to serve as a more comprehensive development database.

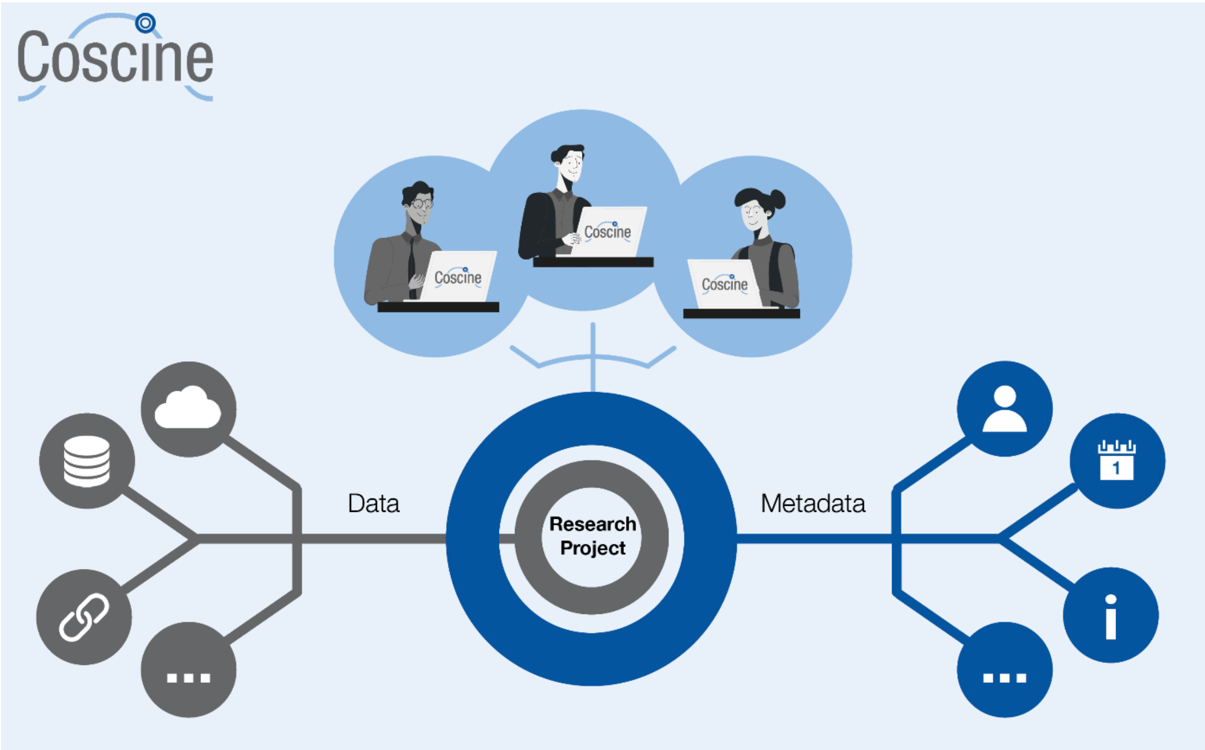


Figure 31: Everything starts with a research project in Coscine

One of the foundational principles of Coscine is its alignment with FAIR (Findable, Accessible, Interoperable, and Reusable) data management principles. As an integral part to modern scientific research, they ensure that data is handled in a way that maximizes its value and utility both now and in the future. Coscine's design and functionalities strongly adhere to these principles, making it a model platform for FAIR data management. The dedication to those principles is evident from the inception of data collection to its eventual reuse. It offers a storage environment that ensures that a significant portion of research data and metadata is not only preserved but also made accessible and reusable. Coscine's technical framework, utilizing W3C standards like RDF and SHACL, aligns with the FAIR principles, particularly in terms of metadata interoperability and reusability. The platform facilitates the assignment of ePIC PIDs to resources, ensuring unique and permanent identification.

A central role in the metadata storage are application profiles – structured metadata forms used for the systematic collection of metadata. These profiles are crucial for organizing and categorizing research data effectively. At their core, application profiles consist of a set of predefined fields that capture specific metadata elements. These elements can include details like data formats, timestamps, identifiers, geographic coordinates, and other relevant descriptive information about the data. The structure and content of these fields are tailored to meet the requirements of particular research projects or disciplines. They facilitate the organized collection of detailed metadata. This comprehensive gathering of information is critical for understanding the context, origin, and characteristics of the underlying data. They help to enhance the indexing and searchability of data within the platform.

3.1.1 Coscine as a Development Database

The VVM project, focusing on the development and implementation of validation and verification processes, presented unique challenges in data management. Coscine's role was integral as the primary development database for this initiative. The "inD VVM application profile" plays a fundamental role in Coscine's integration with the VVM project. This application profile is designed to methodically capture and categorize key data attributes essential for detailed analysis. It includes fields such as Converter Name and Version, Start and End Day Times, Format details, Natural Behavior and Exposure, Recording identifiers, Geographic Reference Points, Scenario Type, Criticality Phenomena and more. Notably, the extensive Criticality Phenomena list provides a thorough enumeration of traffic scenarios and conditions, crucial for researching and understanding the complexities autonomous vehicles will encounter in the future. It is instrumental in storing the meta information of data uploaded to Coscine, facilitating subsequent in-depth analysis.

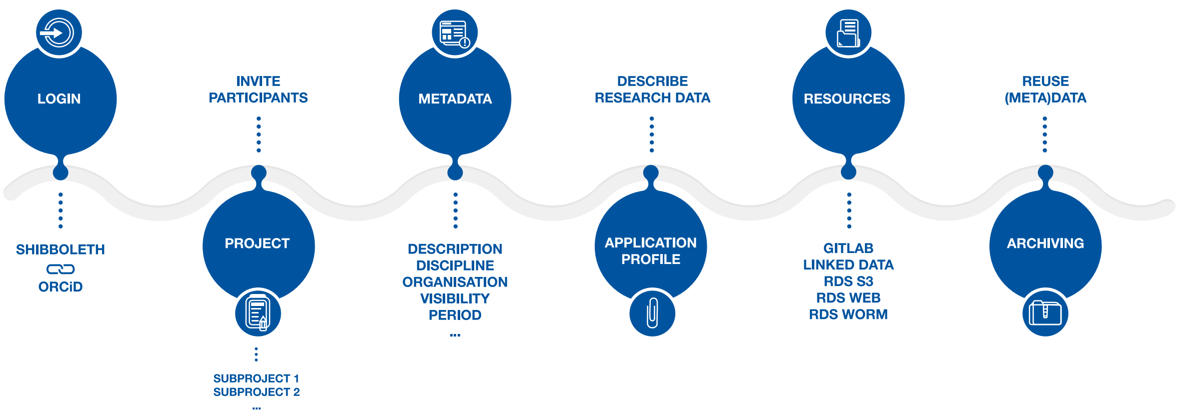


Figure 32: The platform supports researchers during the entire data lifecycle and enables a continuous and easy workflow.

In this capacity, Coscine provided an adaptable and secure environment for managing the extensive data collected in the scope of the initiative. Another significant aspect of the contribution to the VVM project was the provision of enhanced storage capabilities. The RWTH Aachen University facilitated this by allocating 8 Terabytes of high-performance RDS-S3 storage quota in total, enabling efficient storage and retrieval of large data sets, allowing for the archiving and preservation of data for up to a decade of the data after the project ends.

The project's requirements spurred the development of additional structural features in Coscine. This included optimizing file and data upload processes, thereby improving the efficiency and user experience for researchers. A major technological advancement was the integration of ElasticSearch as a technical solution into the platform. This enhancement significantly improved data searchability and retrieval capabilities, allowing researchers to navigate and analyze large datasets with greater ease and precision, building on top of the metadata layer of the platform.

A substantial volume of diverse research data has been uploaded to Coscine, reflecting the extensive scope of the project. To date, the platform houses over 15 000 VVM-related files to date, each contributing valuable insights into the realm of autonomous driving. These files encompass a wide array of formats and content types. A significant portion of the uploaded content includes detailed descriptions of various scenarios. In addition, there is a collection of videos and sensor captures. These visual and sensory data forms are integral to understanding and analyzing the real-world dynamics and responses of autonomous vehicles in diverse driving scenarios. Beyond these files, the initiative has also generated an impressive volume of measurement data. This data, which is instrumental in providing empirical insights into the functioning and performance of autonomous driving systems, is slated for transfer to Coscine. The platform will serve as a repository for this vast data set, ensuring its long-term preservation and accessibility.

3.1.2 Key Insights

In summary, Coscine represents more than just a data storage platform; it is a comprehensive solution for managing research data in accordance with the highest standards of scientific integrity and efficiency. Its role in the VVM project and its contribution to the automotive sector, particularly in advancing autonomous driving technologies and managing complex sensor data, illustrates its utility in practical, real-world applications. As Coscine enters its next phase of innovation, the platform's continued development and application in diverse research scenarios underscore its commitment to a more efficient, responsible, and FAIR approach to research data management.

3.2 Acquiring Reference Data[18]

One of the main challenges of the project is to acquire reference data for the defined FUCs and the respective critical phenomenon in the real world. The analysis of the FUCs and the relevant phenomenon assists in deriving the requirements for a reference sensor system (e.g., like the AVL Dynamic Ground Truth System) and concrete scenario capturing requirements for Real World or proving ground.

Reference sensor system's range and coverage are derived by examining the parameters from the FUCs like the actor behavior, actor classes, actor positioning, infrastructure. These are then examined by AVL and the worst-case scenario for each of them is considered to define the range and coverage of the system. Based on the above-mentioned parameters, a reference sensor system based on the AVL Dynamic Ground Truth was defined as shown in Figure 30.

[18] Based on M. Nestoriuc, H. Walia "Criticality driven data acquisition in autonomous driving – a basis for completeness and safety argumentation", ELIV – International Congress for Automotive Electronics, Bonn, 2023.

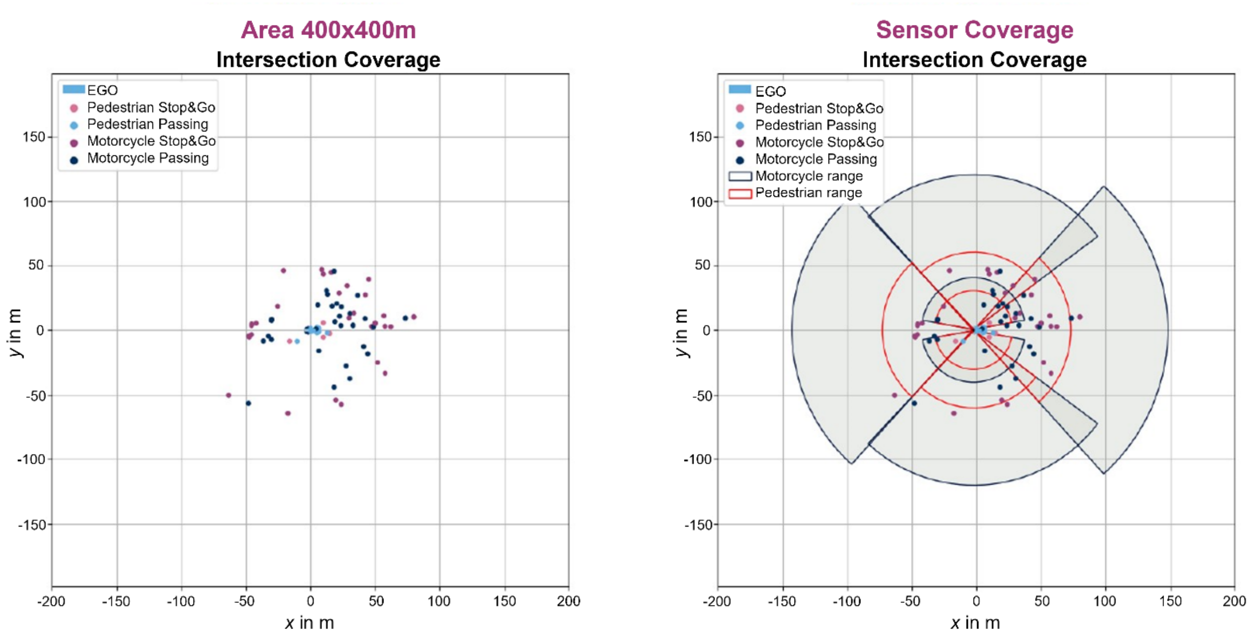


Figure 33: Summarized perception requirements and Sensor Field-of-view

Figure 33 (left) shows the range and area that should be covered to effectively capture the FUCs. The plot on the right shows the actual sensor Field of View. The modified and extended AVL Dynamic Ground Truth System based on the requirements is as shown in Figure 34 below.

Figure 34: Dynamic Ground Truth System

3.2.1 Capturing Data with Privacy in Mind

Data Acquisition and Management workflow is visually represented in Figure 34 below.

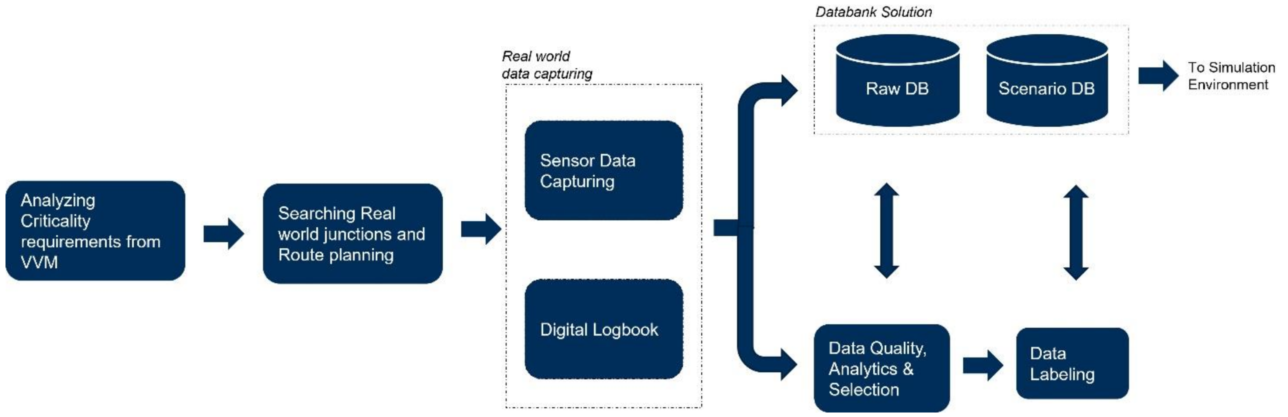


Figure 35: Data Acquisition and Management workflow

Data capturing begins with the requirements that come from the FUCs and criticality phenomenon. Moreover, it is also assessed if the scenario is safe for real-world traffic situations or should be captured in a proving ground for a feasibility check. Next step is to look for real-world traffic junctions where such scenarios could be captured based on the traffic junctions with suitable road, traffic and infrastructure properties.

After the target junctions have been identified, the data capturing phase begins. Here, along with the data from the AVL Dynamic Ground Truth System, the data from additional Sensor Under Test (SUT) units is captured in a synchronized way. These SUT units are automotive sensors supplied by the project partners for the sake of comparison with the ground truth. To link a captured scenario with the corresponding criticality phenomena we have integrated a tool called the ‘AVL Digital Logbook’ with an enriched new feature that helps in filtering and post-processing. Here, the operator can easily tag predefined criticality phenomena during the test drive, which becomes associated with the recordings. Below are the main steps involved in this process.

1. Searching real world junctions: PyJUNCEL (PYthonJUNCtionELection)
2. Location search
3. Location scoring
4. Route optimizer
5. Post-Drive-Analytics

The captured data is then stored in a Raw Database considering the corresponding GDPR requirements. This is done by storing the original data inside a safety cage with role-based restricted access and at the same time providing anonymized data to the project partners.

3.2.2 Labelling of Data

After the captured data is processed and stored on the cloud, all the scenarios are securely shared with the company ‘understand.ai GmbH (UAI)’ which is the labeling supplier for this project. Each scenario is a package containing point cloud data as JSON frames, anonymized images, preview videos of all the cameras, and metadata. Based on the requirements of the project like, the input data, label formats, etc., and the standard labeling guides from UAI for different elements required in the project like, 3D static and dynamic elements, 2D static and dynamic elements, detailed alignments were conducted between VVM partners and UAI to come up with a format that is suitable for further usage in the project.

As per the defined data transfer interface, UAI gets access to the (big) data and labels them as per the defined standards and uploads the labels back to the cloud. The label files for each scenario contain two JSON files each containing dynamic and static labels, respectively. These are then further shared with the VVM partners for further conversion into the desired formats (e.g., OMEGA-format, OSI, etc.).

3.3 Handling of Reference Data[6]

Reference Data is handled in three different steps starting from online in-vehicle processing, lightweight cloud backend, and the raw data backend. The online in-vehicle processing is enabled by an Android app called ‘AVL Digital Logbook’ with a new feature called ‘ADAS Event tagging’. This provides a user-friendly and versatile solution for documenting scenarios and events related to autonomous driving. Before the data is ingested into a full-fledged cloud-based data solution it is critical to perform some pre-processing and filtering steps to optimize the costs associated with the cloud. This is achieved using a lightweight solution called “AVL Data Analytics” software, which is designed to provide quick insights into big data. It uses event detection algorithms and statistical analysis methods to efficiently analyze the data.

After the relevant data is selected based on the above-mentioned process, the raw data can be uploaded to the Raw Data Backend called the “AVL Raw Database Solution”. This is based on the “AVL ADAS/AD Big Data & Analytics Platform (AAP)” capable of managing all kinds of data involved in the development, verification, and validation of critical automated driving functions. Basically, it involves automated data ingest workflows, data management solution, offline-perception, auto-tagger, data search and scenario/tag viewer and data analytics along with custom KPI calculation possibilities as shown in the Figure 36 below.

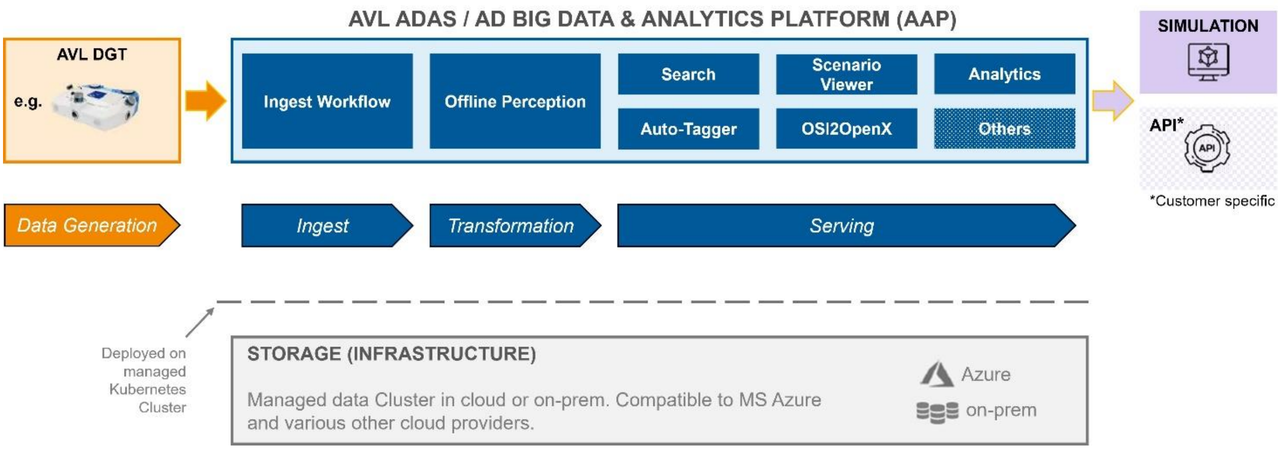


Figure 36: AVL AAP Data Life Cycle

A custom cloud-based solution based on the AAP solution described above is implemented and integrated in the VVM project. Here are the following components of this solution:

1. Raw data ingest of ground truth raw sensor data and SUT data along with corresponding vehicle data and metadata.
2. Automated data ingest workflow for processing the raw vehicle data in the defined formats, and structures.
3. GDPR compliant data storage of un-anonymized data called as ‘Safety Cage’ and anonymized data storage.
4. Role based access structure (e.g., General Project Partner, Labeling Supplier, AVL AAP Admin, etc.) for varying levels of access for the project partners depending on the individual needs in order to optimize costs and reduce unnecessary access to the data.
5. Data search interface to find raw data based on underlying metadata.
6. Scenario data interface to export and connect the processed scenario data with the VVM scenario database.

Figure 37 below depicts the architecture of the raw database solution explained above with the corresponding data flow and all the components, presented during the VVM Mid-term event.

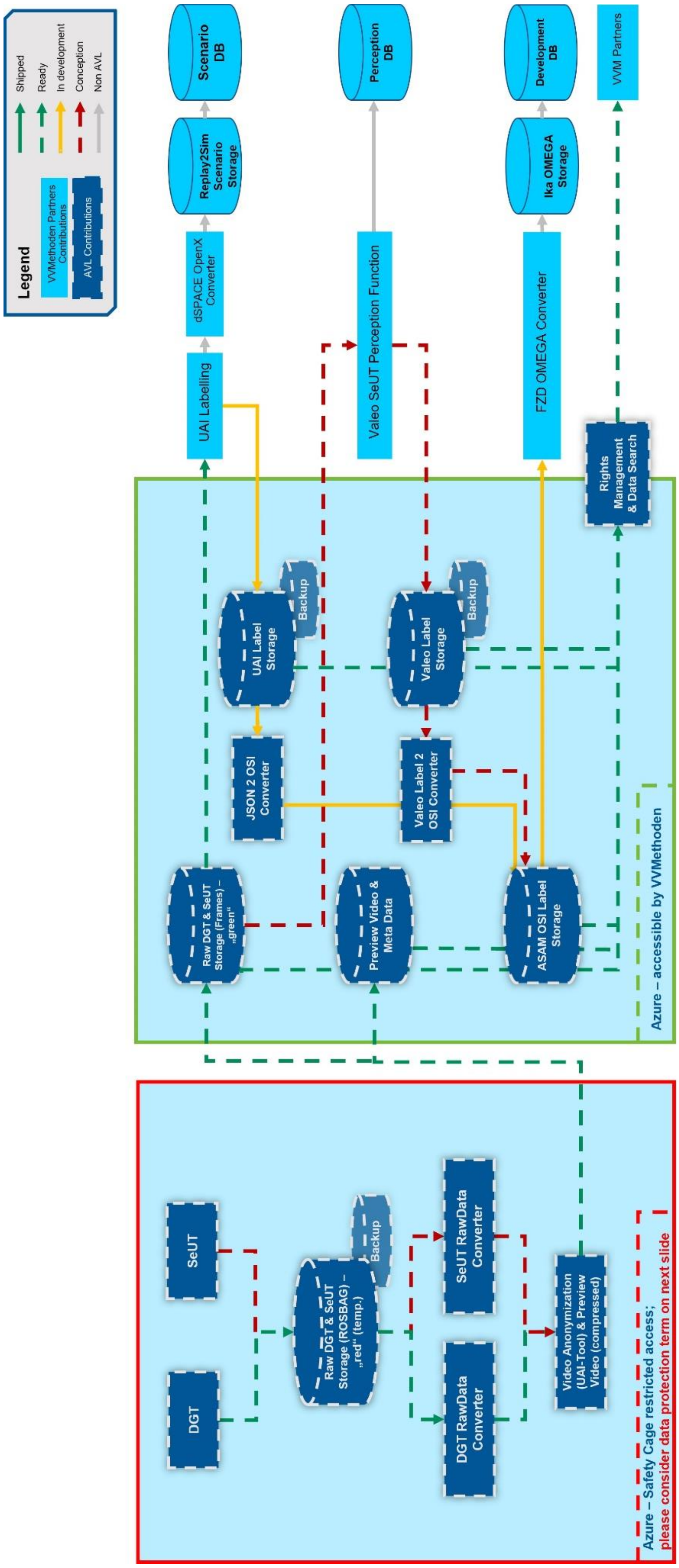


Figure 37: VVM Raw Database Architecture

3.4 OMEGA-format: A Comprehensive Open-Source Measurement Data Format[19]

To achieve a representativeness of scenarios concerning the ODD, a large amount of data is required. Existing data is stored in a multitude of different formats, but a single data format is needed, on which the automated algorithms that identify scenarios can be applied. Therefore, the OMEGA-format was defined to provide a unified data definition. When defining such a format, concepts and entities that are required by the argumentation for validation and verification must be present. To guarantee this, the OMEGA-format was developed in conjunction with the A.U.T.O ontology[20] and based on the 6-Layer Model. Therefore, it contains information about the static environment, dynamic objects, states, and weather. Additionally, it is tracked so that it covers all potential influencing factors of traffic participant behavior, including those of ADS.

On the technical side, the OMEGA-format is derived from the established data formats from Pegasus and the Common Data Format from L3Pilot[21]. Its hierarchical structure is visualized in Figure 38. The definition of road information (Layer 1) combines ideas from the OpenDRIVE standard and Lanelet2[22]. The road network is grouped into roads, which are subdivided into lanes. In contrast to OpenDRIVE s, the geometry of lanes is specified through polylines of the left and right border of the lane. This eases the definition of complex shapes and the conversion from labeling to into the format. The characteristics of the borders of lanes are described in more detail through boundaries. Objects are stored in an object-list-based manner. Besides the specification of what data is needed, requirements on the data quality are set.

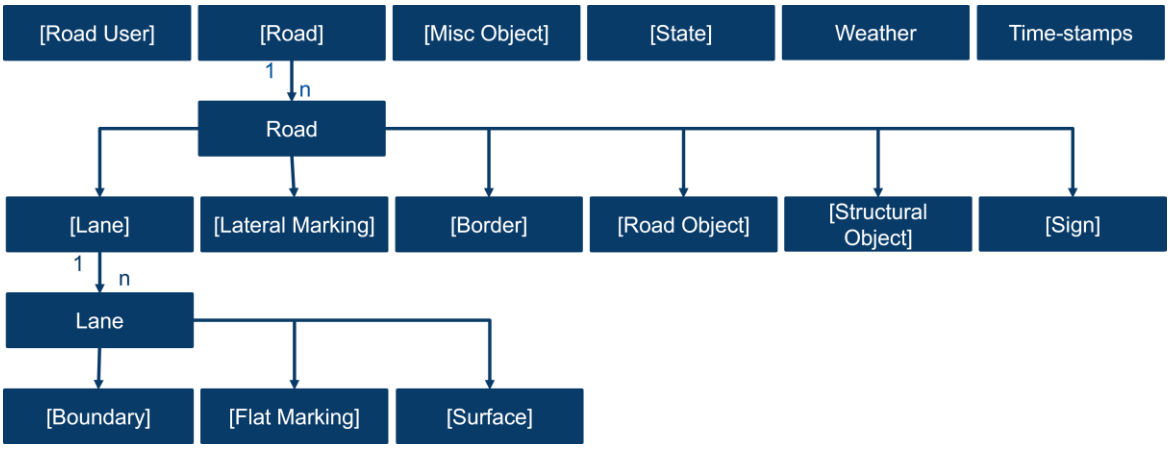


Figure 38: Hierarchical structure of the OMEGA-format

One goal of the OMEGA-format is to be an interface between a multitude of data sources and the automated algorithms of the scenario database. Therefore, multiple converters from existing data formats were developed (see Figure 39). This includes converters from the map formats OpenDRIVE

---

[19] Based on M. Scholtes, M. Schuldes, H. Weber, N. Wagener, M. Hoss, L. Eckstein “OMEGAFormat: A Comprehensive Format of Traffic Recordings for Scenario Extraction”, Uni-DAS FAS Workshop 2022

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

[21] https://github.com/l3pilot/l3pilot-cdf

[22] https://github.com/fzi-forschungszentrum-informatik/Lanelet2

and Lanelet2. Additionally, to the data created in VVM, the inD dataset [23]and the highD dataset[24] are available in the OMEGA-format.

Not all required data points are always available for each a data source. This is especially true for weather information, which can influence sensor performance heavily. The enrichment of such data sources with information from other sources can alleviate the issue. An example is weather information which are provided by the German Meteorological Service. Through a weather station matching of the trajectories in the OMEGA-format, the corresponding historical weather information can be identified and added to the file. Therefore, enriching the data source and enabling a deeper understanding of potential criticality influences.

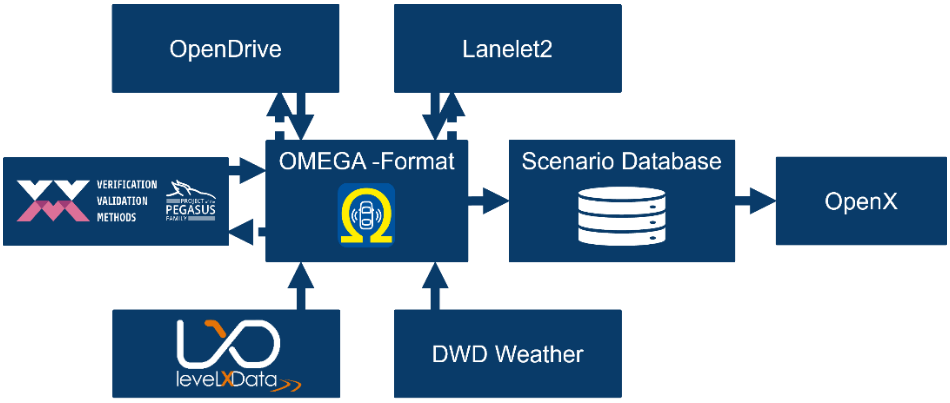


Figure 39: OMEGA-format as an interface between data and scenario database

Not only the understanding of the behaviors is important, but also identifying the source of false behavior. For ADS one main source is failures in perception. So, the analysis of these failures is important. The OMEGA-format is extended with a perception format that can store perception data in an object list-based manner. The data structure is analog to that of the reference data but expanded with fields storing the self-confidence of the perception system. The symmetric definition aids in quantifying the perception performance compared to the reference dat.

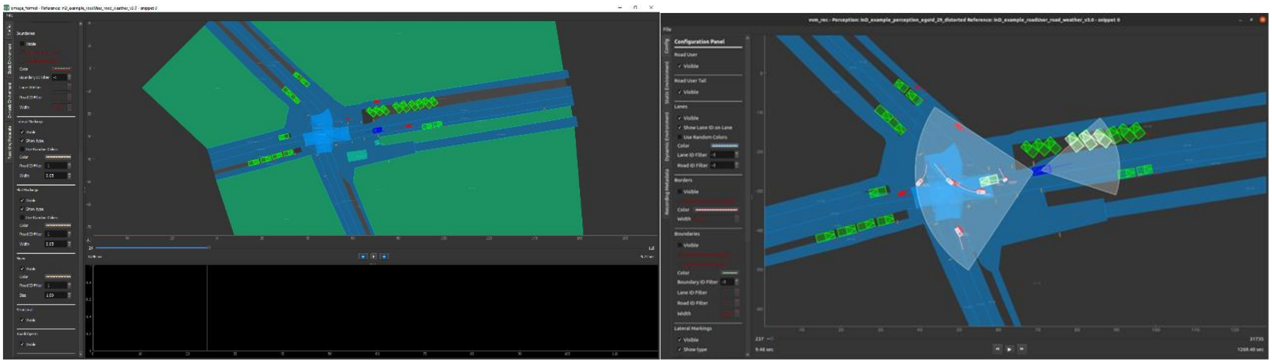


Figure 40: OMEGA-format Visualizer. Left an intersection of the inD-dataset. On the right is a visualization of the OMEGA-format with SUT data as white overlay.

A rich set of tools exists for the OMEGA-format. Python and C++ APIs enable easy creation, reading and importantly, the validation and sanity checking for data in the OMEGA-Format. This tooling is

[23] https://levelxdata.com/ind-dataset/
[24] https://levelxdata.com/highd-dataset/

completed by an intuitive, modular visualization library, that enables the inspection of such files. A screenshot of the visualizer is shown in Figure 40. The tooling and detailed documentation of the OMEGA-format is published under the MIT license on github.com[25]. Free availability and freedom of usage is important to establish it for broader adoption.

3.5 Automatic Extraction of Concrete Instances of Logical Scenarios and Attributes

The acquisition of logical scenario instances, which involve logical scenarios with parameters and their distribution, is facilitated by a scenario database. Figure 41 depicts the process in detail. Starting with reference data, scenarios are identified and extracted. For each identified scenario, parameters, attributes, and events are extracted and stored within the database. Consequently, the database holds a collection of concrete scenarios defined in the terminology of logical scenarios. These concrete scenarios represent a set of points in the parameter space of the logical scenario. By analyzing these points, distributions that approximate the real-world distribution can be estimated. Using these distributions, logical scenario instances are generated. Alternatively, scenarios derived from knowledge, such as insights from experts, can be directly incorporated into the database. This can involve adding either logical scenario instances or concrete scenarios. If concrete scenarios are manually inserted, they must be disregarded when deriving logical scenario instances from data unless special considerations are applied. A query interface enables the selection of relevant subsets of scenarios for specific use cases. The database can convert these logical scenarios or the collection of concrete scenarios into simulation instructions like OpenSCENARIO, which serves as the basis for defining test cases and executing the evaluation loop for ADS.

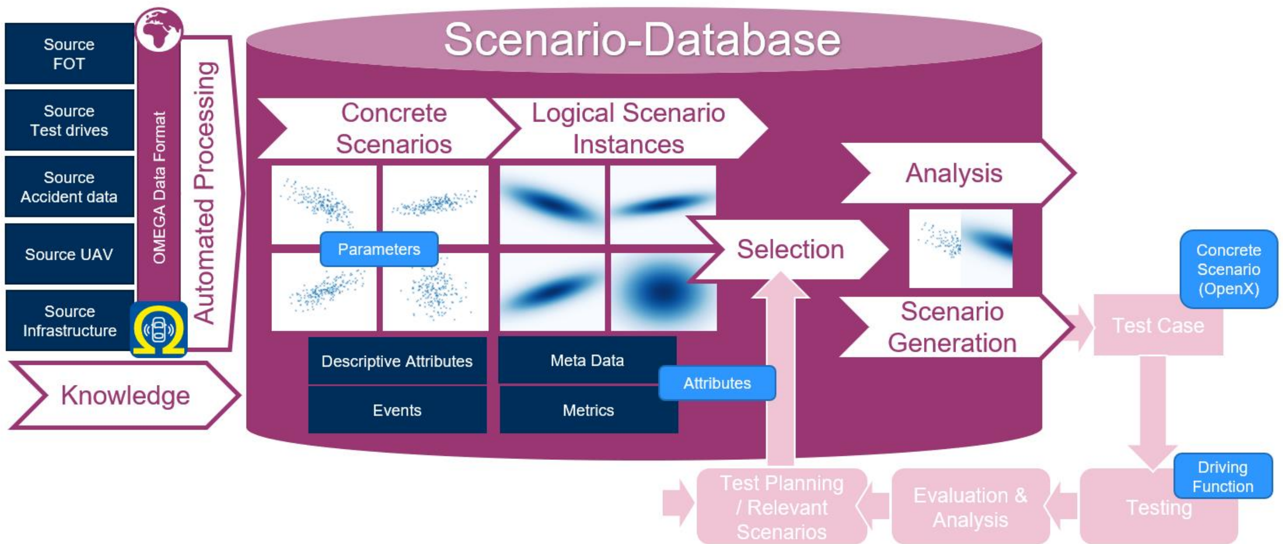


Figure 41: Dataflow through the scenario database

3.5.1 Scenario Identification and Parameter Computation

The scenarios are identified based on the trajectory, shape and map information provided by the OMEGA-format. The initial phase involves the preparation of map information (Layer 1-3). During

---

[25] https://github.com/ika-rwth-aachen/omega_format

this stage, intersections and roundabouts are identified, and their descriptions are derived. The computation involves determining the conflict area and the angles of incoming roads from a calculated intersection/roundabout center. Furthermore, semantic relations between lanes are computed. To ensure a concise and robust description, the lane sections are automatically merged and split. This not only respects the calculated intersection area but also ensures consistency despite possible variations in labeling. All objects are associated with lanes and roads. By utilizing estimated lane and road centerlines, Frenet coordinates corresponding to the position of the objects are computed, similar to OpenSCENARIO. These relative coordinates are crucial for scenarios and parameters to adapt seamlessly to changes in road networks.

After the map and localization information is computed, the temporal information is split according to the enveloping scenarios. Meaning, the perspective of ego vehicle's is taken, and the drive is divided into segments based on infrastructure characteristics. Within each enveloping scenario, various analyses are automatically conducted. To detect scenarios and events, rule-based algorithms based on common metrics are used. Additional metrics are also defined or adapted to enhance the detection of interactions in urban traffic for robust results. An example of such a parameter is given in Figure 42. Here the minimum Time Headway (THW) and Time to collision (TTC) of the scenario “intersecting conflict” is given as empirical cumulative distribution plots. The values are extracted from the inD-dataset through the mechanics of the scenario database. To optimize efficiency, the process is designed hierarchically. This ensures that metrics used multiple times are calculated only once, and information can be detailed as required.

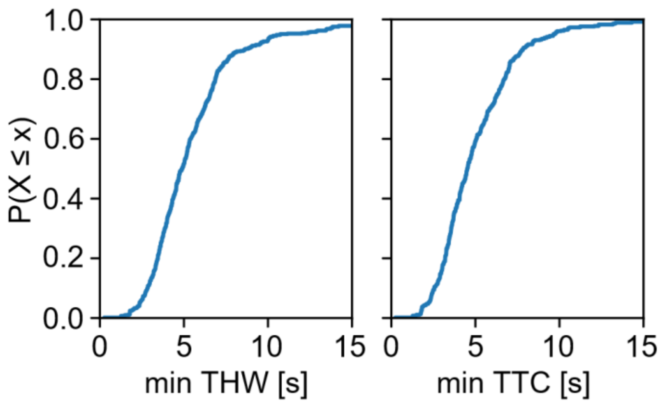


Figure 42: Example Parameter distribution showing the ECDF plot of the minimal THW and TTC value in the base scenario “intersecting conflict”

3.6 Perception Database

The performance of ADS heavily relies on the sensor perception abilities. Therefore, it is important to identify systematic issues. For this, a perception database is developed. Since the reference data is only available in an object list-based manner, the sensor data has to be transformed to object lists too. The data format of choice is the OMEGA-format. As described in Sec. 3.4 , it is included to store object-list based data from sensor perception. Additionally, to each value, the self-confidence of the perception model creating the object-list is recorded. Together with a corresponding reference data file, the data can be uploaded into the perception database (which is integrated into the scenario database). During the analysis numerous object detection and object tracking metrics are computed. The algorithms are provided by the python library py-motmetris[26].

An interface showing the example results of such an analysis are shown in Figure 43. In the shown data, artificial perception data (through the addition of noise) is used since no other data was available at the moment. Here a user can visually inspect the performance of the perception system. Through a breakdown into a time of the recording, problematic timespans can be identified. Through filtering options, the ODD can be specified (e.g., only looking at data with high precipitation), aiding in focusing on the relevant. A correlation with extracted scenarios to identify systematic perception issues caused by scenario and parameter ranges is a promising idea but was not explored in this project.

[26] https://github.com/cheind/py-motmetrics

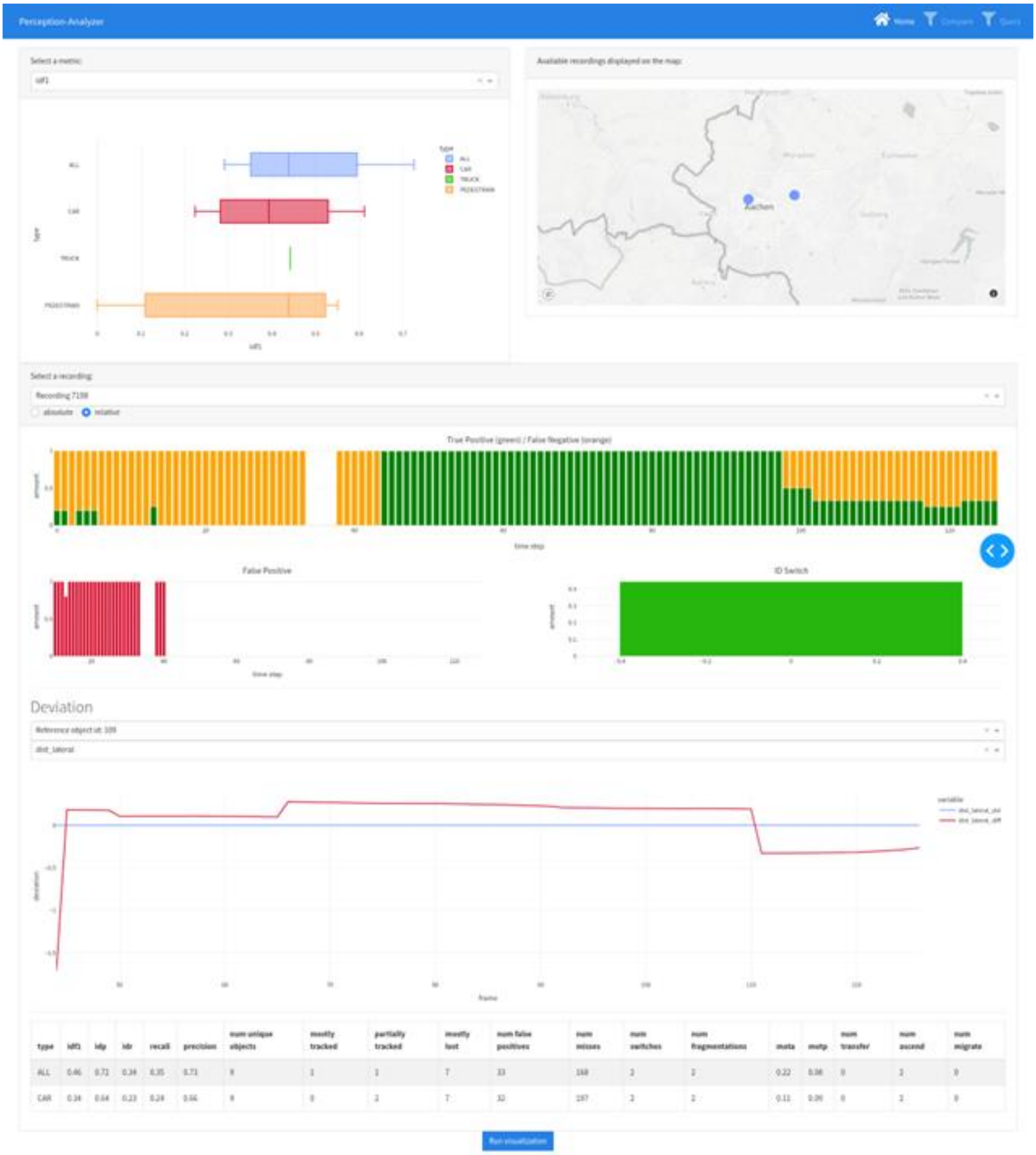


Figure 43: Interface of the perception database

3.7 Scenario Database Implementation

The scenario database is composed of different parts, each taking on different tasks of the overall process. Specifically, it is composed of the following:

- User management to manage access and authorization to the scenario database.
- Input data database (called Upload DB) to store uploaded OMEGA-format data and ensure traceability.

- Worker orchestration and work queue to start the in Sec. 3.5 described automatic identification of scenarios and its parameters and attributes.
- Core of scenario database where the results of the extraction are stored.
- Interface of the scenario database, which provided analysis of data in the database, filtering functions to narrow down to the, for the user, relevant scenarios and scenario generation logic to provide OpenSCENARIO files corresponding to the selection.

For the implementation of these functions Docker is used to provide scalability of microservices. The database definitions and server logic are implemented in python.

Figure 44 displays the possible user interaction with the scenario database. The interaction with the scenario database can be categorized in four groups:

- **Initialization** of the scenario database
- **Uploading and Extraction** of Data
- **Querying** for relevant scenarios
- **Monitoring** the workings of the database

During the initialization, the database system is set up. The worker management is realized with the python library prefect and during setup, the scenario extraction process is defined in a so-called Flow which is registered with work queue. Agents can perform the actual extraction on request. Additionally, user authorization and access rights are defined by an admin. The extraction code itself is tracked on a code version control system such as GitLab. This is necessary for traceability of scenario results which is important for ensuring safety and validity. When the initialization is finished, users can start uploading data in the form of the OMEGA-format. Each file upload triggers an automatic execution of the scenario identification and parameter and attribute extraction engine (Engine). The results are stored in the core database of the scenario database (Scenario DB). When data is uploaded to the scenario database, users can interact with the database through an interface. The interface provides multiple filtering options, tools to analyze and understand the scenarios and the ability to download OpenSCENARIO files with corresponding simulation instructions. This interaction is described in Sec. 3.8. Parallel to these interactions monitoring of the processes is important to ensure reliable functioning.

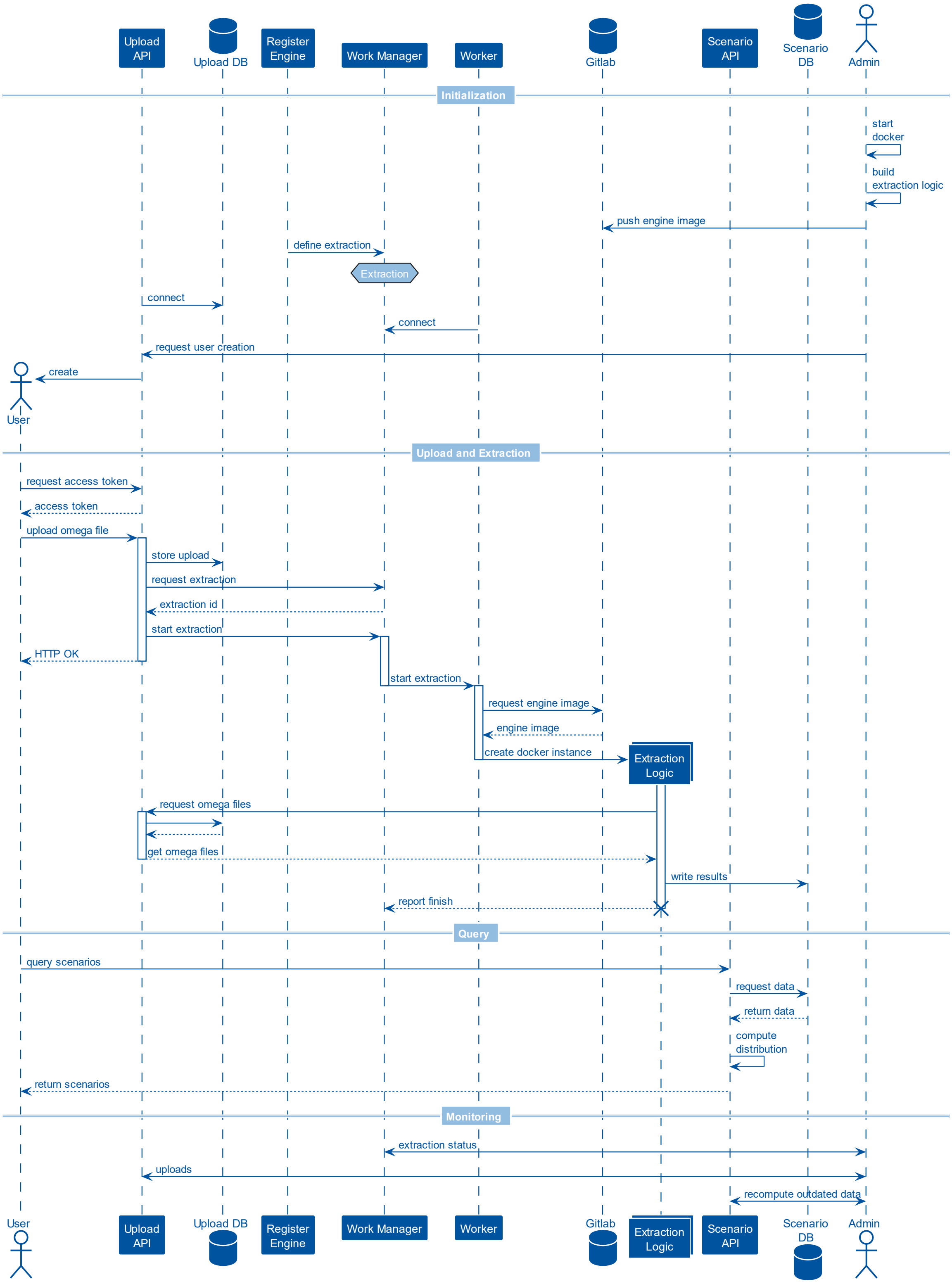


Figure 44: Interaction logic of the scenario database

3.8 Querying Scenarios

When evaluating an ADS, at one time, the tester is not interested in all data at once but wants to focus on certain aspects. This is also true for the development of ADS. Therefore, the database requires a filtering function that enables the user to focus on what is relevant in the situation and for the ODD. The database provides functions for filtering the scenarios based on all layers of the 6-layer model. This includes the scenario classification, parameters, attributes of the scenarios, characteristics of the road network and weather conditions. The filtering function is translated to a database query. In the case of the developed scenario database an SQL statement. For parameters and attributes, the desired parameter ranges can be specified.

It is not only interesting to look at individual base scenarios, but their sequence can be of interest too. This is why the concept of combined scenarios was developed. To search for such combined scenarios, an interface is required. To address this, a graph-based query definition approach has been developed. An example query is shown in Figure 45.

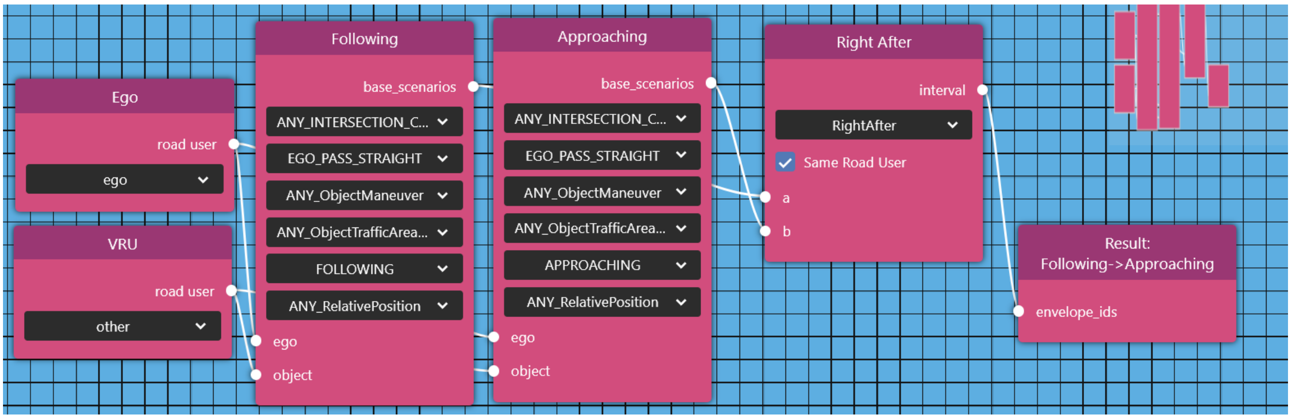


Figure 45: Query Interface for sequences of base scenarios and events

Such an interface can be found in video editing software like Blender, and lets users visually nodes, modify node properties, and connect them to construct complex processes. This adapted approach facilitates the definition of queries for sequences scenarios. To maintain flexibility in the structure and processes that can be built with the tool, fixed interfaces between nodes are crucial. Three types of nodes are defined for this purpose:

- Source nodes: Specify road users and intersections / roundabouts / segments
- Filter nodes: Define filtering criteria such as events and base scenarios
- Result nodes: Specify the output format for final or intermediate results

By connecting these nodes, users can specify specific road users traveling through intersections, experiencing defined events or base scenarios. Through a sequence filter node, multiple events or scenario node can be linked, such as the occurrence of base scenario A immediately after base scenario B. This graph-based query definition simplifies even complex query constructs.

However, a significant challenge arises in translating this query graph into a database query, such as SQL. To translate the graph into a query statement, each node must be translated into a corresponding query component, and the inter-node relationships must be standardized.

Each node separately translated to a statement, which adheres to a fixed output format for consistent filtering by subsequent nodes. Each node refers to the statements of input nodes and applies its filtering criteria to produce an output statement. Result nodes modify the input statement to generate user-specific outputs, such as lists of relevant scenarios or distribution plots.

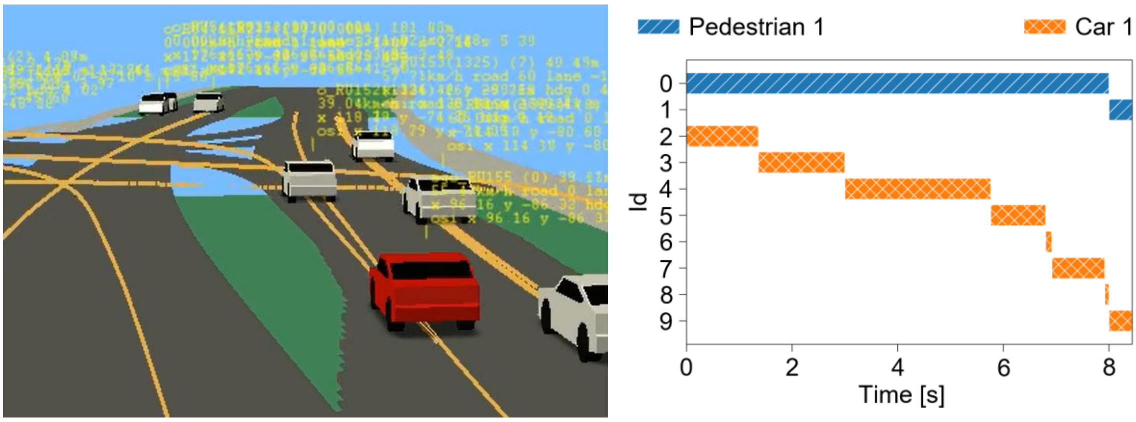


Figure 46: Video and sequence of base scenarios in found enveloping scenario number 614.

The query defined in Figure 45 searches for all enveloping scenarios, in which an ego object is travelling straight through an intersection and is first following a lead object and right after approaching the same object. The object has to be of type car or truck (called other in the interface). Through the compilation of the shown query graph to an SQL statement the corresponding enveloping scenarios can be identified. Two of the identified enveloping scenarios in the inD dataset are scenario number 614 (Figure 46 and Table 2) and scenario number 6067 (Appendix Figure 47 and Table 3). The tables show the breakdown to the actual basis scenarios with the different interacting objects. For scenario 614 the found basis scenarios are basis scenario 2 and 3 and for scenario 6057 they are scenario 7 and 8. This demonstrates that the query functionality works in practice. One can also observe the Gantt charts in Figure 46 and Figure 47 that for each point in time there is only one basis scenario relation between the ego vehicle and one other traffic participant. But there can be interactions between the ego vehicle and multiple other traffic participants at the same time, which illustrates the intended design of basis scenarios described in Sec. 2.2.3.2.

The generation of simulatable scenario closes the tasks of the scenario database. According to the definition in Sec. 2.2.3.2. The simulation can be controlled on different levels. The optimal choice depends on the specific use case.

Table 2: Sequence of base scenarios defined from its concepts found enveloping scenario number 614.

id	Road User Type	Start Time	End Time	Object Traffic Area Change	Relative Object Position	Ego Maneuver	Conflict Type	Object Maneuver	Longitudinal State
0	pedestrian	0	8	crossing	oncoming	pass straight			
1	pedestrian	8	8.44	crossing					
2	**car**	**0**	**1.36**		**same arm**	**pass straight**	**diverging**	**turn left**	**following**
3	**car**	**1.36**	**3**		**same arm**	**pass straight**	**diverging**	**turn left**	**approaching**
4	car	3	5.76		same arm	pass straight	diverging	turn left	
5	car	5.76	6.8	leaving	same arm	pass straight	diverging	turn left	
6	car	6.8	6.92		same arm	pass straight	diverging	turn left	
7	car	6.92	7.92	changing	same arm	pass straight	diverging	turn left	
8	car	7.92	8		same arm	pass straight	diverging	turn left	
9	car	8	8.44				diverging	turn left	

4 Summary

This deliverable showed the developed approach of how to model an ODD through logical scenarios with the help of a scenario database. Through the derivation from the 6-layer model and ontologies a scenario concept that is designed to cover an ODD was created. The outermost structure is the enveloping scenario. It takes the perspective of an object and splits its trajectory based on the characteristics of the road network (layer 1-3 of the 6-layer model). The dynamics in these segments are described through base scenarios. They define the relation and behavior of the ego object to one other object. For each of these base scenarios, parameters are defined, creating logical scenarios. Based on these scenarios a completeness and coverage argumentation are built. More complex and longer scenarios can be created through the combination of the base scenarios, so called combined scenarios. Samples from the logical scenarios can be converted to OpenSCENARIO in different levels. The result of the scenario generation can be used to formulate test cases.

To acquire parameter distributions for the logical scenario instances that reflect the desired ODD, a scenario database is utilized. First, the scenario database requires data captured in the real word corresponding the same ODD. This deliverable presents the process of how to capture such data and transform it to reference data through labelling. Privacy issues are addressed through the concept of a safety cage. The object-list based reference data is stored in the newly defined OMEGA-format, which functions as an interface between scenario database and data sources and is linked to an ontology to tie in with the argumentation. Based on this data, the scenario database can identify scenarios and their parameters from which the logical scenario instances are derived. Additionally, attributes and events are computed, which help in filtering the scenarios to the relevant ones.

During the development of these methods and the creation of scenario databases, the exchange of data between the involved parties is necessary. This issue is tackled through a development database. It enables the exchange of data with metadata in a traceable manner.

In conclusion, the deliverable presents a methodology and demonstrates practically how to define logical scenarios and acquire parameter distributions that model an OD. Therefore, aiding in the development and validation and verification of ADS.

5 Appendix

5.1 Additional Example Found Scenario Sequence

Table 3: Sequence of base scenarios defined from it concepts found enveloping scenario number 6057.

id	Road User Type	Start Time	End Time	Relative Object Position	Ego Maneuver	Object Traffic Area Change	Longitudinal State
0	pedestrian	0	0.28	from left	pass straight		
1	pedestrian	0.28	5.72	from left	pass straight	crossing	
2	pedestrian	5.72	6.08	from left	pass straight		
3	bicycle	0	2.52	from left	pass straight		
4	bicycle	2.52	4.56	from left	pass straight	changing	
5	bicycle	4.56	6.08	from left	pass straight		
6	car	0	0.72	same arm	pass straight	approaching	
7	**car**	**0.72**	**0.92**	**same arm**	**pass straight**	**following**	
8	**car**	**0.92**	**1.64**	**same arm**	**pass straight**	**approaching**	
9	car	1.64	4.72	same arm	pass straight		
10	ego	6.08	9.08				free

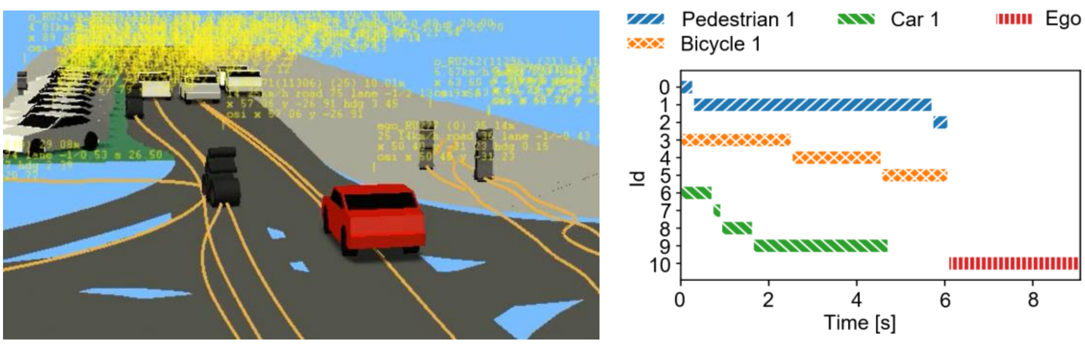


Figure 47: Video and sequences of basis scenarios in found enveloping scenario number 6057.

5.2 Basis Scenario Definition

As stated in CROSS, the derivation of base-scenarios modeled using an ontology. For this, the Web Ontology Language was used within the ontology framework Protégé. Using an ontology, the complex hierarchy between properties, concepts, superclasses and base-scenarios could be modelled in an traceable manner. The class structure was derived manually, while the framework was used to provide checks for consistency and visual exploration of the scenario concept.

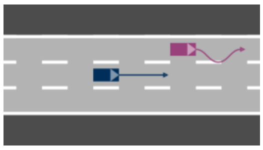
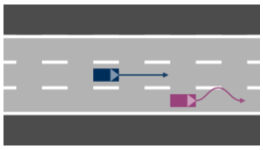
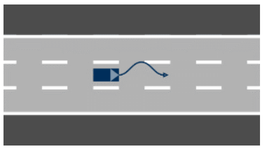
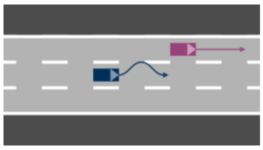
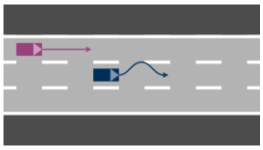
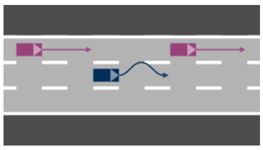
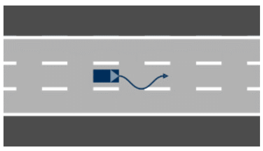
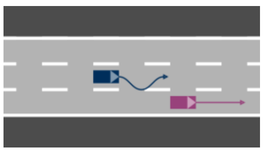
Within the ontology, two main "trees" were modelled independently from each other: One for the abstract properties and concepts, and one for the base-scenarios and their superclasses. Properties are linked with properties using defined "realizes" relation. For bilateral concepts, the "realizes" relation is used directly, while for individual concepts – which can be applied to ego-vehicle and object – "realizes for ego" and "realizes for object" are used.

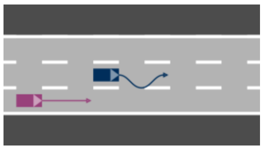
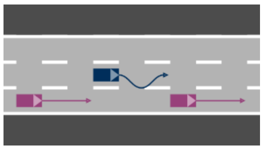
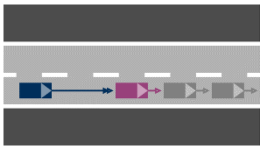
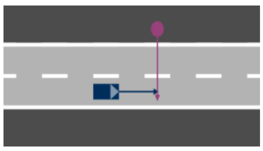
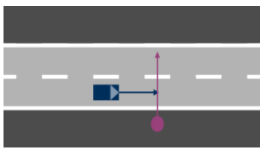
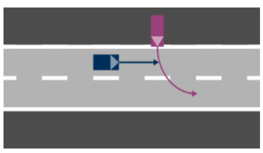
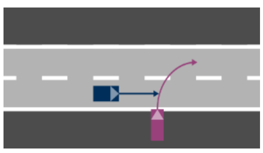
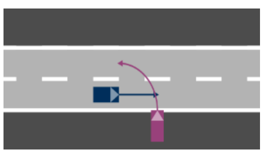
The entry point for users of the ontology is the hierarchy of base-scenarios, while the concept and abstract properties support two applications. One is the checks for consistency of the derived base-scenarios, the other is to provide internal insight into the derivation of base-scenarios, which can serve as input for the implementation of algorithms for scenarios detection or generation (see. CROSS). One primary consistency check used is that at each branching of the tree at least one relation to a property is added to from a new, more refined superclass or multiple superclasses realizing different concepts are combined in one child-class. Base-scenarios are always derived from superclasses, typically by the combination of multiple superclasses.

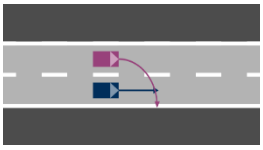
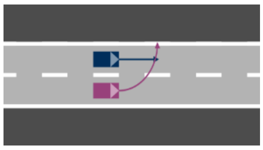
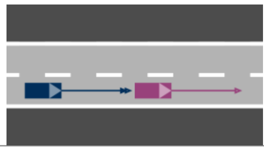
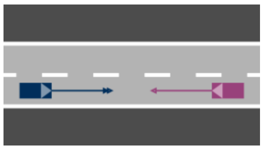
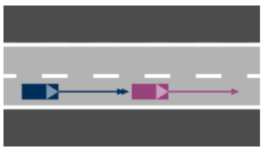
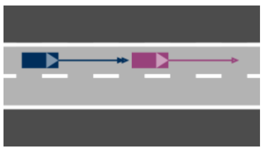
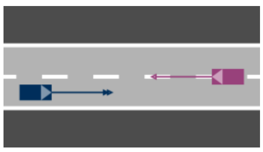
While the ontology represents a model that is typically explored top-down, it can also be used, to generate a bottom-up representation of all base-scenarios, i.e. the catalog of base scenarios. Within the tabular format, all base-scenarios could be listed, as well as their superclasses and the realized abstract properties. Additionally, an overview of superclasses and all concepts are provided. All base-scenarios derived are listed in Table 4. The overall structure of superclasses is given in Table 5.

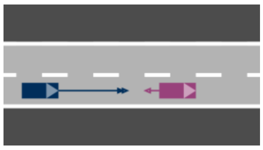
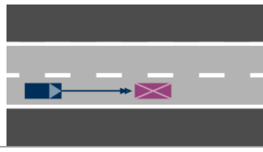
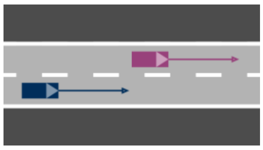
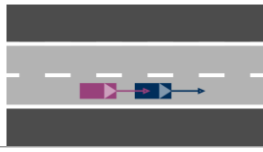
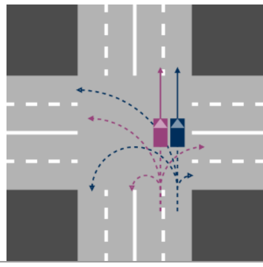
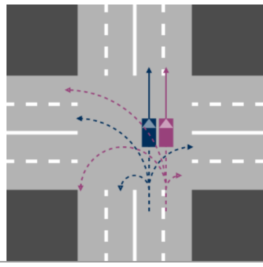
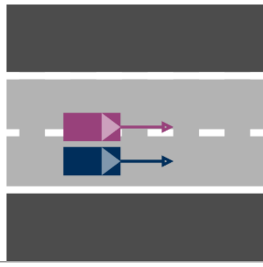
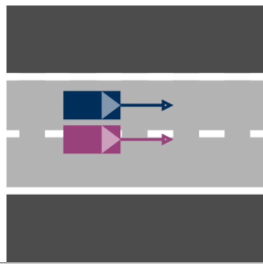
Apart from names, which aim to express the nature of the base-scenarios, pictograms for each base scenario were generated programmatically using the Cairo graphics library in Python. To derive the pictograms, each base-scenario was evaluated for its used concepts, which were the used to position vehicle on an intersection or stretch of road and to define the movements using arrows. Furthermore, the implementation of the programmatical generation of pictograms served as an additional check for constancy check of the base-scenario and concept ontology, as inconsistencies either became apparent in the processing of the ontology or could be easily determined by visually inspecting the resulting catalog of visualized scenarios.

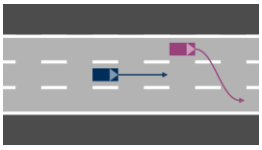
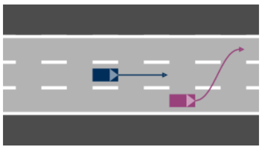
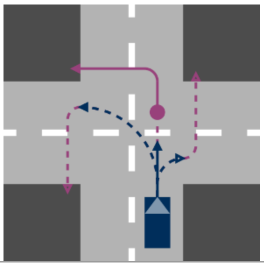
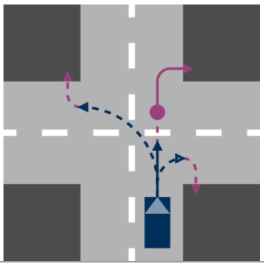
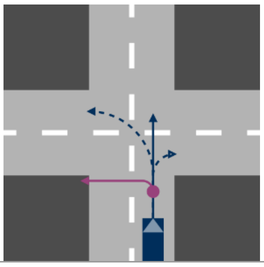
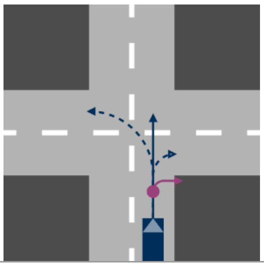
Table 4: Overview of base scenarios

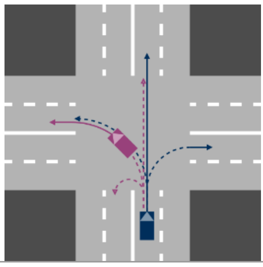
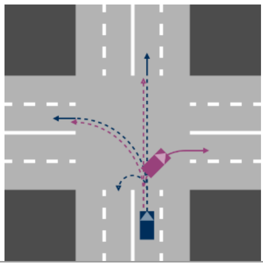
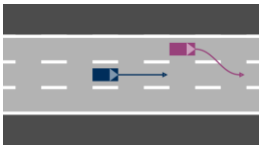
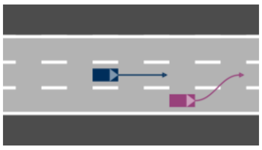
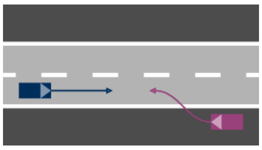
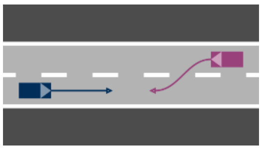
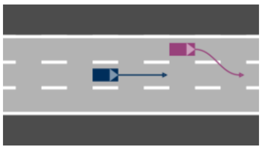
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
aborted_enter_lead_l	Abgebrochener Eintritt eines Vorausfahrenden von links	Aborted lead entering from left	aborted_enter_lead	For obj: same_direction transition For obj: aborted For obj: at_left For obj: entering longitunidal_traffic For obj: stay_in_traffic_area	
aborted_enter_lead_r	Abgebrochener Eintritt eines Vorausfahrenden von rechts	Aborted lead entering from right	aborted_enter_lead	For obj: same_direction transition For obj: aborted For obj: at_right For obj: entering longitudinal_traffic For obj: stay_in_traffic_area	
aborted_lc_l_0	Abgebrochener freier Fahrstreifenwechsel nach links	Aborted uninfluenced lane change left	aborted_lc_0 aborted_lc_l	transition For ego: to_left For ego: free For ego: lane_change longitudinal_traffic For ego: aborted	
aborted_lc_l_1	Abgebrochener Fahrstreifenwechsel nach links mit Vorausfahrendem	Aborted lane change left with lead object	aborted_lc_1 aborted_lc_l	For obj: in_front_of transition For ego: to_left For ego: lane_change longitudinal_traffic For ego: aborted	
aborted_lc_l_2	Abgebrochener Fahrstreifenwechsel nach links mit Hinterherfahrendem	Aborted lane change left with following object	aborted_lc_2 aborted_lc_l	transition For ego: to_left For obj: behind For ego: lane_change longitudinal_traffic For ego: aborted	
aborted_lc_l_3	Abgebrochener Fahrstreifenwechsel nach links mit Vorausfahrendem und Hinterherfahrendem	Aborted lane change left with lead object and following object	aborted_lc_3 aborted_lc_l	transition For ego: to_left For ego: lane_change longitudinal_traffic For obj: in_front_of For ego: aborted	
aborted_lc_r_0	Abgebrochener freier Fahrstreifenwechsel nach rechts	Aborted uninfluenced lane change right	aborted_lc_0 aborted_lc_r	transition For ego: to_right For ego: free For ego: lane_change longitudinal_traffic For ego: aborted	
aborted_lc_r_1	Abgebrochener Fahrstreifenwechsel nach rechts mit Vorausfahrendem	Aborted lane change right with lead object	aborted_lc_1 aborted_lc_r	For obj: in_front_of transition For ego: to_right For ego: lane_change longitudinal_traffic For ego: aborted	

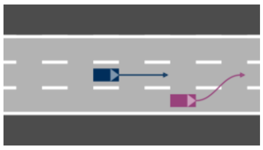
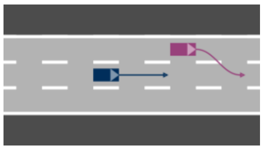
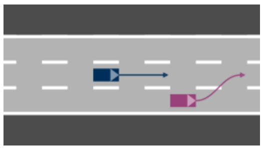
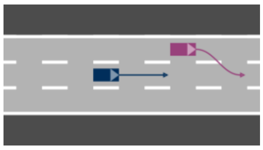
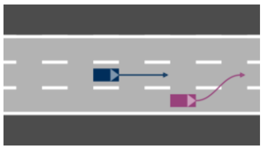
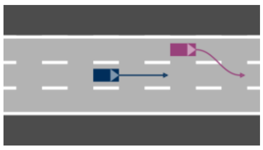
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
aborted_lc_r_2	Abgebrochener Fahrstreifenwechsel nach rechts mit Hinterherfahrendem	Aborted lane change right with following object	aborted_lc_2 aborted_lc_r	transition For ego: to_right For obj: behind For ego: lane_change longitunidal_traffic For ego: aborted	
aborted_lc_r_3	Abgebrochener Fahrstreifenwechsel nach rechts mit Vorausfahrendem und Hinterherfahrendem	Aborted lane change right with lead object and following object	aborted_lc_3 aborted_lc_r	transition For ego: to_right For ego: lane_change longitunidal_traffic For obj: in_front_of For ego: aborted	
approach_TJ	Annähern an ein Stauende	Approach a traffic jam	approach	For ego: approaching For obj: traffic_jam longitunidal_traffic state	
approach_lat_crossing_traffic_area_from_left	Annähern an ein den Ego-Verkehrsraum von links kreuzendes Objekt	Approach a laterally moving object crossing the ego-traffic area from left	approach_lat_crossing_traffic_area approach_lat_from_left	For obj: lateral_obj For ego: approaching For obj: from_left state longitunidal_traffic For obj: cross_ego_traffic_area	
approach_lat_crossing_traffic_area_from_right	Annähern an ein den Ego-Verkehrsraum von rechts kreuzendes Objekt	Approach a laterally moving object crossing the ego-traffic area from right	approach_lat_crossing_traffic_area approach_lat_from_right	For obj: lateral_obj For ego: approaching For obj: from_right state longitunidal_traffic For obj: cross_ego_traffic_area	
approach_lat_entering_traffic_area_from_left	Annähern an ein den Ego-Verkehrsraum von links betretendes Objekt	Approach a laterally moving object entering the ego-traffic area from left	approach_lat_entering_traffic_area approach_lat_from_left	For obj: lateral_obj For ego: approaching For obj: enter_ego_traffic_area For obj: from_left longitunidal_traffic state	
approach_lat_entering_traffic_area_from_right_forward	Annähern an ein den Ego-Verkehrsraum von rechts vorwärts betretendes Objekt	Approach a laterally moving object entering the ego-traffic area from right forward	approach_lat_entering_traffic_area_from_right	For obj: lateral_obj For ego: approaching For obj: enter_ego_traffic_area For obj: from_right For obj: same_direction longitunidal_traffic state	
approach_lat_entering_traffic_area_from_right_oncoming	Annähern an ein den Ego-Verkehrsraum von rechts betretendes entgegenkommendes Objekt	Approach a laterally moving object entering the ego-traffic area from right oncoming	approach_lat_entering_traffic_area_from_right	For obj: lateral_obj For ego: approaching For obj: enter_ego_traffic_area For obj: from_right	

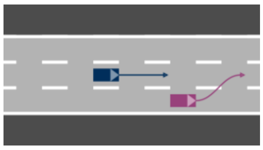
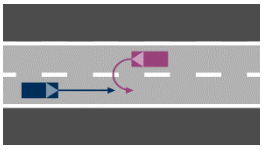
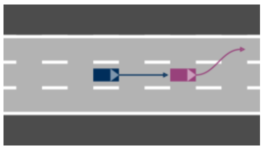
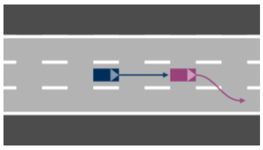
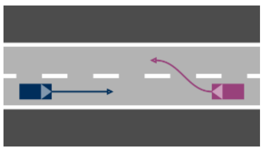
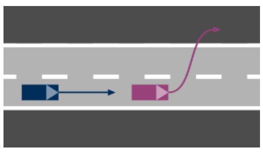
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
				longitudinal_traffic For obj: opposite_direction state	
`approach_lat_leaving_traffic_area_from_left`	Annähern an ein den Ego-Verkehrsraum von links verlassendes Objekt	Approach a laterally moving object leaving the ego-traffic area from left	`approach_lat_from_left` `approach_lat_leaving_traffic_area`	For obj: lateral_obj For ego: approaching For obj: from_left For obj: leaving_ego_traffic_area longitudinal_traffic state	
`approach_lat_leaving_traffic_area_from_right`	Annähern an ein den Ego-Verkehrsraum von rechts verlassendes Objekt	Approach a laterally moving object leaving the ego-traffic area from right	`approach_lat_from_right` `approach_lat_leaving_traffic_area`	For obj: lateral_obj For ego: approaching longitudinal_traffic For obj: leaving_ego_traffic_area For obj: from_right state	
`approach_lead`	Annähern an einen Vorausfahrenden	Approach a leading object	`approach`	For ego: approaching longitudinal_traffic For obj: leading state	
`approach_oncoming`	Annähern an ein entgegenkommendes Objekt	Approach Oncoming object	`approach`	For ego: approaching For obj: oncoming longitudinal_traffic state	
`approach_overlapping_lead_l`	Annähern an einen seinen Fahrstreifen überlappenden Vorausfahrenden links	Approach a leading object overlapping its lane left	`approach_overlapping_lead` `approach_overlapping_left`	For ego: approaching For obj: leading For obj: overlapping_object For obj: at_left longitudinal_traffic state	
`approach_overlapping_lead_r`	Annähern an einen seinen Fahrstreifen überlappenden Vorausfahrenden rechts	Approach a leading object overlapping its lane right	`approach_overlapping_lead` `approach_overlapping_right`	For ego: approaching For obj: leading For obj: overlapping_object For obj: at_right longitudinal_traffic state	
`approach_overlapping_oncoming`	Annähern an einen seinen Fahrstreifen überlappenden Entgegenkommenden	Approahc an oncoming object overlapping its lane	`approach_overlapping_left` `approach_overlapping`	For obj: oncoming For ego: approaching longitudinal_traffic For obj: overlapping_object For obj: at_left state	

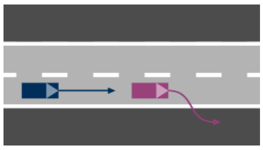
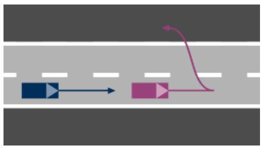
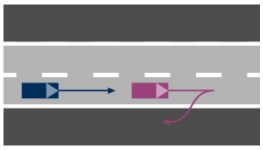
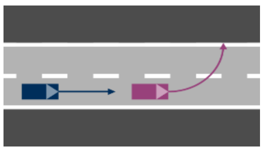
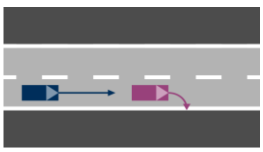
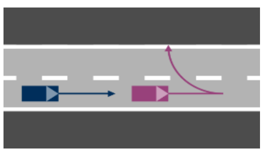
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
approach_reversing	Annähern an ein rückwärtsfahrendes Objekt	Approach a reversing object	approach	For ego: approaching For obj: reversing_lead longitudinal_traffic state	
approach_static	Annähern an ein statisches Objekt	Approach a static object	approach	For ego: approaching For obj: static longitudinal_traffic state	
avoid_undertaking	Rechtsüberholen vermeiden	Avoid undertaking	follow	For ego: following For obj: undertaken longitudinal_traffic state	
close_obj_behind	Dichtes Auffahren eines Hinterherfahrenden	Tailgating	long_traffic_overlay	overlay For obj: following longitudinal_traffic	
close_obj_side_in_intersection_left	Laterales Nahdistanz-Ereignis links in Kreuzung	Lateral close distance event left in intersection	close_obj_side_in_intersection	overlay For ego: close_obj For obj: at_left intersecting_traffic same_arm	
close_obj_side_in_intersection_right	Laterales Nahdistanz-Ereignis rechts in Kreuzung	Lateral close distance event right in intersection	close_obj_side_in_intersection	overlay For obj: at_right For ego: close_obj intersecting_traffic same_arm	
close_obj_side_l	Laterales Nahdistanz-Ereignis links	Lateral close distance event left	close_obj_side	overlay For obj: at_left longitudinal_traffic For ego: close_obj	
close_obj_side_r	Laterales Nahdistanz-Ereignis rechts	Lateral close distance event right	close_obj_side	overlay For obj: at_right longitudinal_traffic For ego: close_obj	

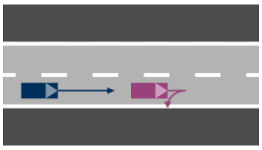
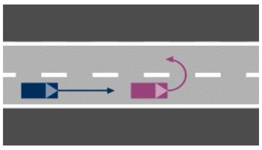
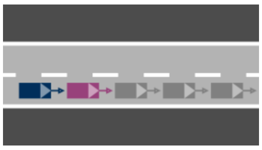
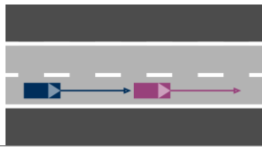
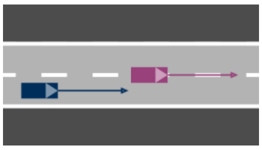
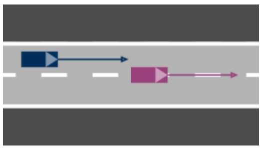
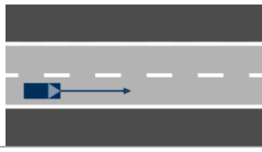
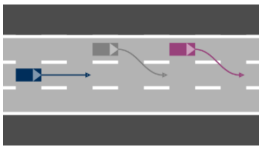
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
cut_through_l	Durchscherer von links	Cut through from left	cut_through	For obj: same_direction For obj: stay_in_traffic_area transition For obj: exiting For obj: at_left For obj: entering longitunidal_traffic For obj: stay_in_traffic_area	
cut_through_r	Durchscherer von rechts	Cut through from right	cut_through	For obj: same_direction For obj: stay_in_traffic_area transition For obj: exiting For obj: entering longitunidal_traffic For obj: at_right For obj: stay_in_traffic_area	
diverging_lead_leaving_TA_after_node_l	Divergierender Vorausfahrender, der den Ego-Verkehrsraum nach dem Knoten nach links verlässt	Diverging leading object leaving ego-traffic area after the node left	diverging_lead_leaving_TA_after_node diverging_lead_left	behind_node transition For obj: to_left For obj: diverging intersecting_traffic For obj: leaving_ego_traffic_area	
diverging_lead_leaving_TA_after_node_r	Divergierender Vorausfahrender, der den Ego-Verkehrsraum nach dem Knoten nach rechts verlässt	Diverging leading object leaving ego-traffic area after the node right	diverging_lead_leaving_TA_after_node diverging_lead_right	behind_node transition For obj: diverging intersecting_traffic For obj: to_right For obj: leaving_ego_traffic_area	
diverging_lead_leaving_TA_before_node_l	Divergierender Vorausfahrender, der den Ego-Verkehrsraum vor dem Knoten nach links verlässt	Diverging leading object leaving ego-traffic area before the node left	diverging_lead_leaving_TA_before_node diverging_lead_left	transition For obj: to_left For obj: diverging intersecting_traffic before_node For obj: leaving_ego_traffic_area	
diverging_lead_leaving_TA_before_node_r	Divergierender Vorausfahrender, der den Ego-Verkehrsraum vor dem Knoten nach rechts verlässt	Diverging leading object leaving ego-traffic area before the node right	diverging_lead_leaving_TA_before_node diverging_lead_right	transition For obj: to_right For obj: diverging intersecting_traffic before_node For obj: leaving_ego_traffic_area	

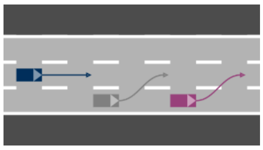
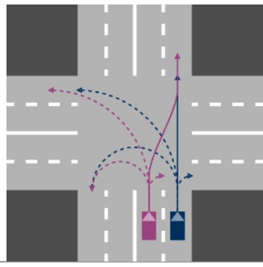
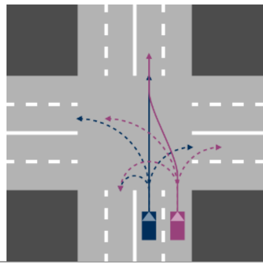
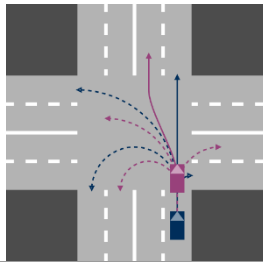
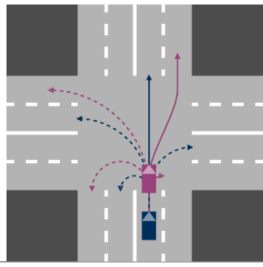
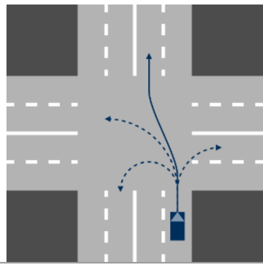
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
diverging_lead_within_ego_TA_l	Divergierender Vorausfahrender nach links innerhalb des Ego-Verkehrsraums	Diverging leading object withing ego traffic area left	diverging_lead_left diverging_lead_within_ego_TA	transition For obj: stay_in_traffic_area For obj: to_left For obj: diverging intersecting_traffic	
diverging_lead_within_ego_TA_r	Divergierender Vorausfahrender nach rechts innerhalb des Ego-Verkehrsraums	Diverging leading object withing ego traffic area right	diverging_lead_right diverging_lead_within_ego_TA	transition For obj: stay_in_traffic_area For obj: diverging intersecting_traffic For obj: to_right	
enter_lead_l	Eintritt eines Vorausfahrenden von links	A lead object entering from left	enter_l enter_within_ego_TA	For obj: same_direction transition For obj: from_left For obj: entering longitudinal_traffic For obj: stay_in_traffic_area	
enter_lead_r	Eintritt eines Vorausfahrenden von rechts	A lead object entering from right	enter_r enter_within_ego_TA	For obj: same_direction transition For obj: entering longitudinal_traffic For obj: from_right For obj: stay_in_traffic_area	
enter_oncoming_entering_ego_traffic_area	Eintritt eines entgegenkommenden Objekts vorwärts in den Ego-Verkehrsraum	Oncoming object entering ego-traffic area	enter_oncoming	transition For obj: entering For obj: opposite_direction longitudinal_traffic For obj: enter_ego_traffic_area	
enter_oncoming_within_ego_traffic_area	Eintritt eines entgegenkommenden Objekts (innerhalb des Ego-Verkehrsraums)	Oncoming object entering (within ego-traffic area)	enter_oncoming	transition For obj: stay_in_traffic_area For obj: entering For obj: opposite_direction longitudinal_traffic	
enter_parallel_forward_left	Paralleler Eintritt eines Objekts vorwärts von links	Object entering parallel forward from left	enter_forward_left enter_parallel_forward enter_parallel_left	For obj: same_direction transition longitudinal_traffic For obj: from_left For obj: entering For obj: forward For obj: enter_ego_traffic_area For obj: parallel	

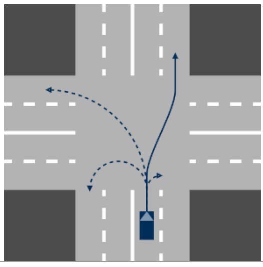
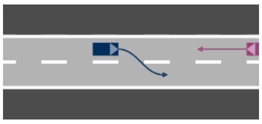
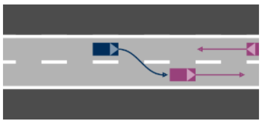
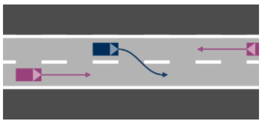
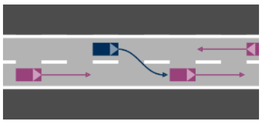
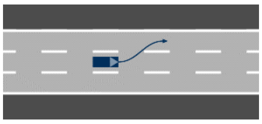
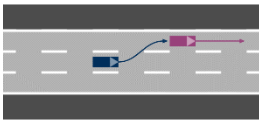
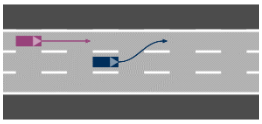
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
enter_parallel_forward_right	Paralleler Eintritt eines Objekts vorwärts von rechts	Object entering parallel forward from right	enter_forward_right enter_parallel_forward enter_parallel_right	For obj: same_direction transition longitudinal_traffic For obj: entering For obj: forward For obj: enter_ego_traffic_area For obj: from_right For obj: parallel	
enter_parallel_reversing_left	Paralleler Eintritt eines Objekts rückwärts von links	Object entering parallel reversing from left	enter_parallel_left enter_parallel_reversing enter_reversing_left	For obj: parallel For obj: same_direction For obj: reversing transition For obj: from_left For obj: enter_ego_traffic_area For obj: entering longitudinal_traffic	
enter_parallel_reversing_right	Paralleler Eintritt eines Objekts rückwärts von rechts	Object entering parallel reversing from right	enter_parallel_reversing enter_parallel_right enter_reversing_right	For obj: same_direction For obj: reversing transition longitudinal_traffic For obj: entering For obj: enter_ego_traffic_area For obj: from_right For obj: parallel	
enter_turning_forward_left	Eintritt eines Objekts vorwärts einbiegend von links	Object entering forward turning from left	enter_forward_left enter_turning_forward enter_turning_left	For obj: same_direction transition longitudinal_traffic For obj: from_left For obj: turning For obj: entering For obj: forward For obj: enter_ego_traffic_area	
enter_turning_forward_right	Eintritt eines Objekts vorwärts einbiegend von rechts	Object entering forward turning from right	enter_forward_right enter_turning_forward enter_turning_right	For obj: same_direction transition longitudinal_traffic For obj: turning For obj: entering For obj: forward For obj: enter_ego_traffic_area For obj: from_right	
enter_turning_reversing_left	Eintritt eines Objekts rückwärts einbiegend von links	Object entering turning reversing from left	enter_reversing_left enter_turning_left enter_turning_reversing	For obj: same_direction transition For obj: reversing For obj: from_left For obj: turning For obj: enter_ego_traffic_area	

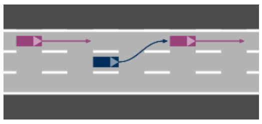
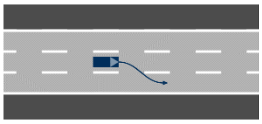
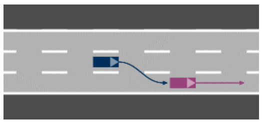
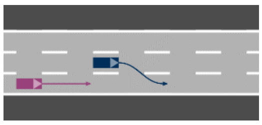
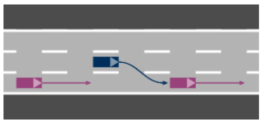
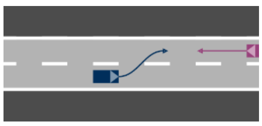
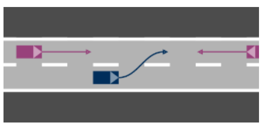
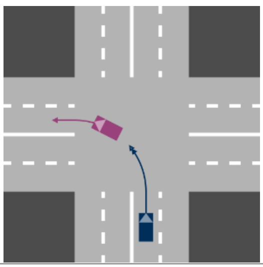
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
				For obj: entering longitunidal_traffic	
enter_turning_reversing_right	Eintritt eines Objekts rückwärts einbiegend von rechts	Object entering turning reversing from right	enter_reversing_right enter_turning_reversing enter_turning_right	For obj: same_direction For obj: reversing transition For obj: turning For obj: enter_ego_traffic_area For obj: entering longitunidal_traffic For obj: from_right	
enter_u-turning	Eintritt eines Objekts wendend	Object entering doing a u-turn	enter_within_ego_TA	For obj: same_direction transition longitunidal_traffic For obj: entering For obj: u-turn_within_TA For obj: stay_in_traffic_area	
exit_lead_l	Austritt eines Vorausfahrenden nach links	A leading object exiting from left	exit_l exit_within_ego_TA	For obj: same_direction transition For obj: to_left longitunidal_traffic For obj: stay_in_traffic_area For obj: exiting	
exit_lead_r	Austritt eines Vorausfahrenden nach rechts	A leading object exiting from right	exit_r exit_within_ego_TA	For obj: same_direction transition longitunidal_traffic For obj: to_right For obj: stay_in_traffic_area For obj: exiting	
exit_oncoming_leaving_ego_traffic_area	Austritt eines Entgegenkommenden aus dem Ego-Verkehrsraum	Oncoming object exiting leaving ego-traffic area	exit_oncoming	For obj: leaving_ego_traffic_area transition For obj: opposite_direction longitunidal_traffic For obj: exiting	
exit_oncoming_within_ego_traffic_area	Austritt eines Entgegenkommenden innerhalb des Ego-Verkehrsraum	Oncoming object exiting (within ego-traffic area)	exit_oncoming	transition For obj: stay_in_traffic_area For obj: opposite_direction longitunidal_traffic For obj: exiting	
exit_parallel_forward_left	Paralleler Austritt aus dem Ego-Verkehrsraum vorwärts nach links	Object exiting parallel forward leaving ego-traffic area to the left	exit_forward_left exit_parallel_forward exit_parallel_left	For obj: same_direction For obj: leaving_ego_traffic_area transition For obj: to_left For obj: forward longitunidal_traffic For obj: parallel For obj: exiting	

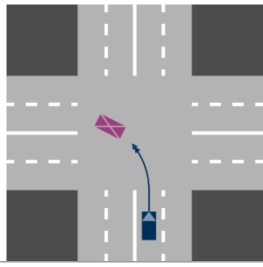
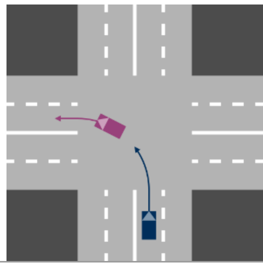
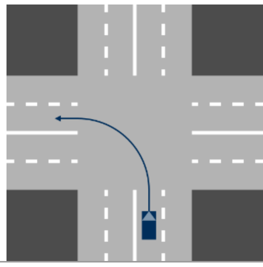
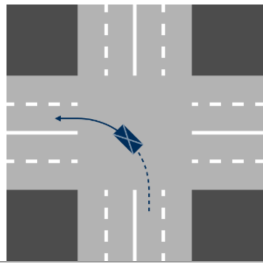
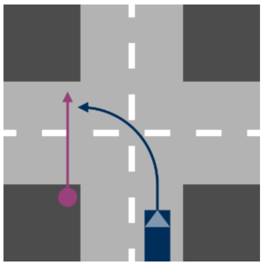
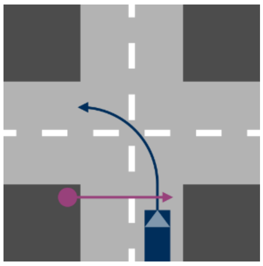
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
exit_parallel_forward_right	Paralleler Austritt aus dem Ego-Verkehrsraum vorwärts nach rechts	Object exiting parallel forward leaving ego-traffic area to the right	exit_forward_right exit_parallel_forward exit_parallel_right	For obj: same_direction For obj: leaving_ego_traffic_area transition For obj: exiting For obj: forward longitudinal_traffic For obj: to_right For obj: parallel	
exit_parallel_reversing_left	Paralleler Austritt aus dem Ego-Verkehrsraum rückwärts nach links	Object exiting parallel reversing to the left	exit_parallel_left exit_parallel_reversing exit_reversing_left	For obj: same_direction For obj: leaving_ego_traffic_area transition For obj: to_left For obj: reversing longitudinal_traffic For obj: parallel For obj: exiting	
exit_parallel_reversing_right	Paralleler Austritt aus dem Ego-Verkehrsraum rückwärts nach rechts	Object exiting parallel reversing to the right	exit_parallel_reversing exit_parallel_right exit_reversing_right	For obj: same_direction For obj: leaving_ego_traffic_area transition For obj: reversing longitudinal_traffic For obj: to_right For obj: parallel For obj: exiting	
exit_turning_forward_left	Austritt abbiegend vorwärts nach links	Object exiting forward turning to the left	exit_forward_left exit_turning_forward exit_turning_left	For obj: same_direction For obj: leaving_ego_traffic_area transition For obj: to_left For obj: forward longitudinal_traffic For obj: turning For obj: exiting	
exit_turning_forward_right	Austritt abbiegend vorwärts nach rechts	Object exiting forward turning to the right	exit_forward_right exit_turning_forward exit_turning_right	For obj: same_direction For obj: leaving_ego_traffic_area transition For obj: turning For obj: forward longitudinal_traffic For obj: to_right For obj: exiting	
exit_turning_reversing_left	Austritt abbiegend rückwärts nach links	Object exiting turning reversing to the right	exit_reversing_left exit_turning_left exit_turning_reversing	For obj: same_direction For obj: leaving_ego_traffic_area transition For obj: to_left For obj: reversing longitudinal_traffic	

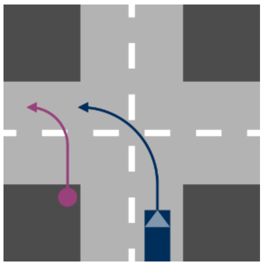
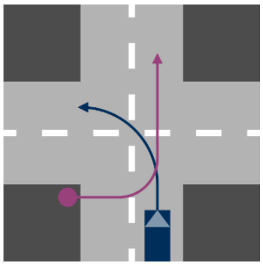
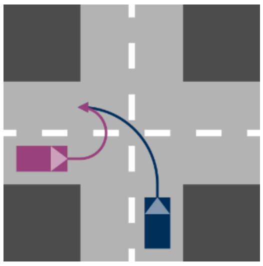
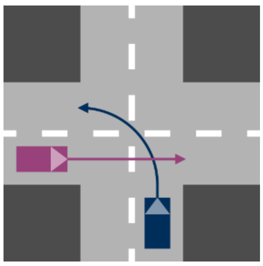
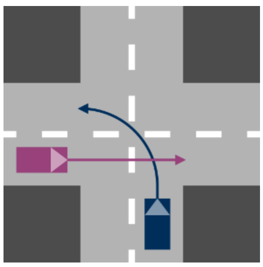
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
				For obj: turning For obj: exiting	
exit_turning_ reversing_ right	Austritt abbiegend rückwärts nach rechts	Object exiting turning reversing to the left	exit_reversing_ right exit_turning_ reversing exit_turning_ right	For obj: same_ direction For obj: leaving_ego_ traffic_area transition For obj: turning For obj: reversing longitunidal_ traffic For obj: to_ right For obj: exiting	
exit_u-turning	Austritt wendend	Object exiting doing a u-turn	exit_within_ego_ TA	For obj: same_ direction transition For obj: u-turn_ within_TA longitunidal_ traffic For obj: stay_ in_traffic_area For obj: exiting	
follow_TJ	Fahren im Stau	Driving in Traffic Jam	follow	For ego: following longitunidal_ traffic For obj: traffic_jam state	
follow_lead	Einem Vorausfahrenden folgen	Follow a leading object	follow	For ego: following For obj: leading longitunidal_ traffic state	
follow_ overlapping_l	Einem seinen Fahrstreifen überlappendem Objekt links folgen	Follow object overlapping its lane left	follow_ overlapping	For ego: following For obj: overlapping_ object longitunidal_ traffic state For obj: at_left	
follow_ overlapping_r	Einem seinen Fahrstreifen überlappendem Objekt rechts folgen	Follow object overlapping its lane right	follow_ overlapping	For obj: at_ right For ego: following For obj: overlapping_ object longitunidal_ traffic state	
free	Freies Fahren	free driving	long_traffic_ state	For ego: free longitunidal_ traffic state	
incomplete_ enter_lead_l	Abgebrochener Eintritt eines Vorausfahrenden von links	Incomplete lead entering from left	incomplete_ enter_lead	For obj: same_ direction incomplete transition For obj: at_left For obj: entering longitunidal_ traffic For obj: stay_ in_traffic_area	

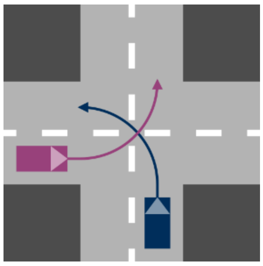
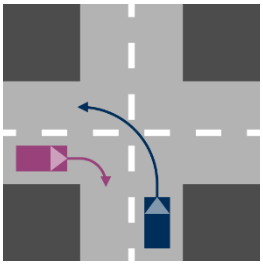
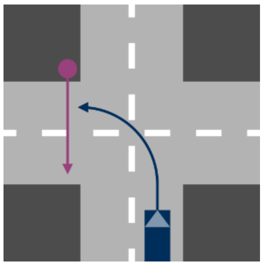
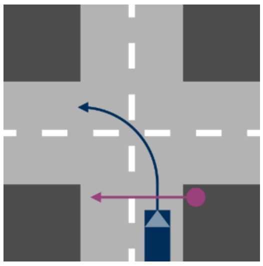
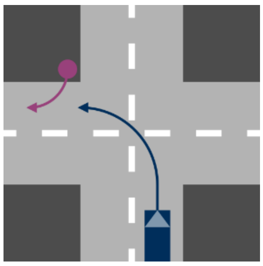
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
incomplete_ enter_lead_r	Abgebrochener Eintritt eines Vorausfahrenden von rechts	Incomplete lead entering from right	incomplete_ enter_lead	For obj: same_ direction incomplete transition For obj: entering longitunidal_ traffic For obj: at_ right For obj: stay_ in_traffic_area	
intersection_ enter_lead_l	Eintritt eines Vorausfahrenden von links innerhalb einer Kreuzung	Lead object entering in intersection from the left	intersection_ enter_lead	transition For obj: at_left For obj: entering intersecting_ traffic	
intersection_ enter_lead_r	Eintritt eines Vorausfahrenden von rechts innerhalb einer Kreuzung	Lead object entering in intersection from the right	intersection_ enter_lead	For obj: at_ right transition For obj: entering intersecting_ traffic	
intersection_ exit_lead_l	Austritt eines Vorausfahrenden nach links innerhalb einer Kreuzung	Lead object exiting in intersection to the left	intersection_ exit_lead	transition For obj: exiting For obj: to_left intersecting_ traffic	
intersection_ exit_lead_r	Austritt eines Vorausfahrenden nach rechts innerhalb einer Kreuzung	Lead object exiting in intersection to the right	intersection_ exit_lead	transition For obj: exiting intersecting_ traffic For obj: to_ right	
intersection_ lc_l	Fahrstreifenwechsel nach links innerhalb einer Kreuzung	Lane change left in intersection	intersection_ lane_change	transition For obj: to_left For obj: lane_ change intersecting_ traffic	

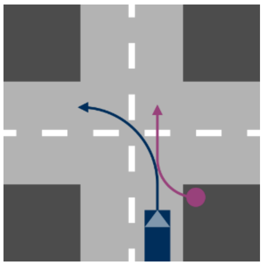
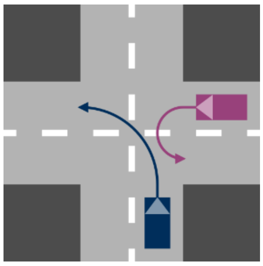
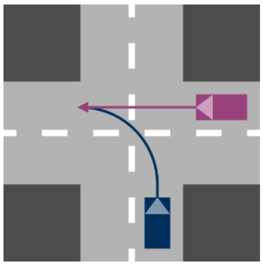
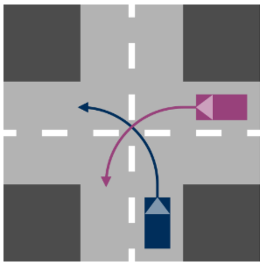
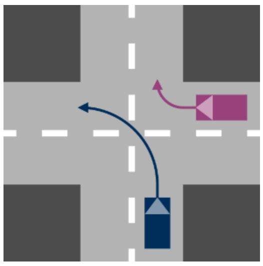
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
intersection_lc_r	Fahrstreifenwechsel nach rechts innerhalb einer Kreuzung	Lane change right in intersection	intersection_lane_change	transition For obj: lane_change For obj: to_right intersecting_traffic	
lc_from_oc_0	Fahrstreifenwechsel aus Gegenverkehr frei	Free lane change from oncoming traffic	lc_0 lc_from_oncoming	transition free from_oncoming For ego: lane_change For obj: opposite_direction longitunidal_traffic	
lc_from_oc_1	Fahrstreifenwechsel aus Gegenverkehr mit Vorausfahrendem	Lane change from oncoming traffic with leading vehicle	lc_1 lc_from_oncoming	transition from_oncoming For obj: in_front_of For ego: lane_change For obj: opposite_direction longitunidal_traffic	
lc_from_oc_2	Fahrstreifenwechsel aus Gegenverkehr mit Hinterherfahrendem	Lane change from oncoming traffic with following vehicle	lc_2 lc_from_oncoming	transition from_oncoming For ego: lane_change For obj: opposite_direction longitunidal_traffic For obj: behind	
lc_from_oc_3	Fahrstreifenwechsel aus Gegenverkehr mit Vorausfahrendem und Hinterherfahrenden	Lane change from oncoming traffic with leading and following vehicles	lc_3 lc_from_oncoming	transition For obj: in_front_of from_oncoming For ego: lane_change For obj: opposite_direction longitunidal_traffic	
lc_l_0	Freier Fahrstreifenwechsel nach links	Uninfluenced lane change left	lc_0 lc_left	For ego: to_left transition free For ego: lane_change longitunidal_traffic	
lc_l_1	Fahrstreifenwechsel mit Vorausfahrendem nach links	Lane change left with lead object	lc_1 lc_left	For ego: to_left transition For ego: lane_change For obj: in_front_of longitunidal_traffic	
lc_l_2	Fahrstreifenwechsel mit Hinterherfahrendem nach links	Lane change right with following object	lc_2 lc_left	For ego: to_left transition For ego: lane_change longitunidal_traffic For obj: behind	

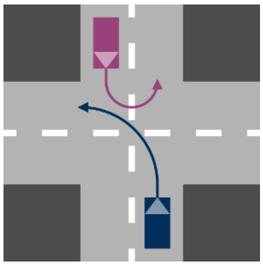
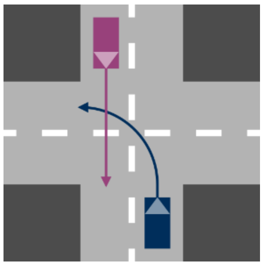
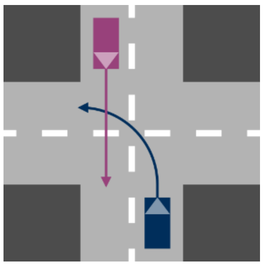
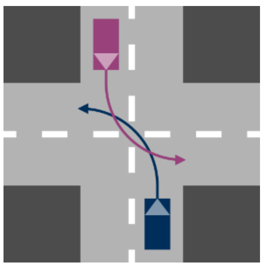
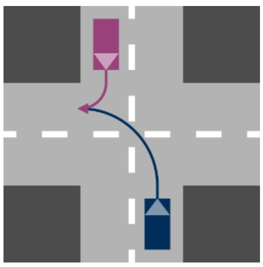
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
lc_l_3	Fahrstreifenwechsel mit Vorausfahrendem und Hinterherfahrendem nach links	Lane change left with lead object and following object	lc_3 lc_left	For ego: to_left transition For obj: in_front_of For ego: lane_change longitunidal_traffic	
lc_r_0	Freier Fahrstreifenwechsel nach rechts	Uninfluenced lane change right	lc_0 lc_right	transition free For ego: to_right For ego: lane_change longitunidal_traffic	
lc_r_1	Fahrstreifenwechsel mit Vorausfahrendem nach rechts	Lane change right with lead object	lc_1 lc_right	transition For ego: to_right For obj: in_front_of For ego: lane_change longitunidal_traffic	
lc_r_2	Fahrstreifenwechsel mit Hinterherfahrendem nach rechts	Lane change right with following object	lc_2 lc_right	transition For ego: to_right For ego: lane_change longitunidal_traffic For obj: behind	
lc_r_3	Fahrstreifenwechsel mit Vorausfahrendem und Hinterherfahrendem nach rechts	Lane change right with lead object and following object	lc_3 lc_right	transition For obj: in_front_of For ego: to_right For ego: lane_change longitunidal_traffic	
lc_to_oncoming_0	Fahrstreifenwechsel in Gegenverkehr ohne Hinterherfahrenden	Lane change to oncoming traffic without following object	lc_0 lc_to_oncoming	transition free For ego: lane_change For obj: opposite_direction longitunidal_traffic to_oncoming	
lc_to_oncoming_2	Fahrstreifenwechsel in Gegenverkehr mit Hinterherfahrendem	Lane change to oncoming traffic with following object	lc_2 lc_to_oncoming	transition For ego: lane_change For obj: opposite_direction longitunidal_traffic For obj: behind to_oncoming	
left_turn_approaching_lead	Einem Vorausfahrenden beim Linksabbiegen annähern	Left turn approaching a leading object	approaching_lead_in_intersection left_turn	maneuver state For ego: left_turn For obj: leading For ego: approaching intersecting_traffic	

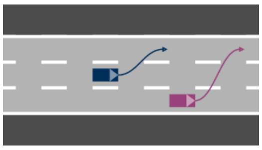
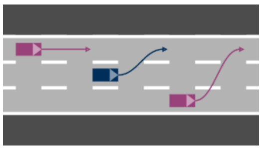
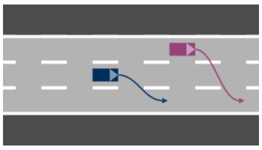
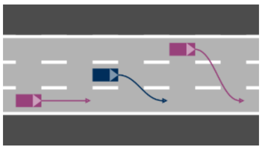
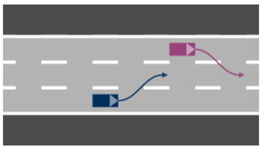
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
left_turn_approaching_static	Einem statischen Objekt beim Linksabbiegen annähern	Left turn approaching a static object	approaching_static_in_intersection left_turn	maneuver state For ego: left_turn For obj: static For ego: approaching intersecting_traffic	
left_turn_following_lead	Einem Vorausfahrenden beim Linksabbiegen folgen	Left turn following a leading object	following_in_intersection left_turn	maneuver intersecting_traffic state For ego: left_turn For ego: following	
left_turn_free	Freies Linksabbiegen	Free left turn	free_intersection_maneuver left_turn	maneuver state For ego: free For ego: left_turn intersecting_traffic	
left_turn_standstill	Stillstand beim Linksabbiegen	Standstill while turning left	left_turn standstill_in_intersection	maneuver intersecting_traffic state For ego: standstill For ego: left_turn	
left_turn_with_obj_from_left_crossing_after_node	Linksabbiegen mit nach dem Knotenpunkt kreuzendem Objekt	Left turn with object from left crossing after the node	conflict_after_node conflict_with_obj_crossing_ego_TA conflict_with_obj_from_left_with_TA_change intersecting_conflict left_turn	maneuver For ego: left_turn For obj: cross_ego_traffic_area traffic_area_change conflict For obj: intersecting For obj: at_left intersecting_traffic behind_node	
left_turn_with_obj_from_left_crossing_before_node	Linksabbiegen mit vor dem Knotenpunkt kreuzendem Objekt	Left turn with object from left crossing before the node	conflict_before_node conflict_with_obj_crossing_ego_TA conflict_with_obj_from_left_with_TA_change intersecting_conflict left_turn	maneuver For ego: left_turn For obj: cross_ego_traffic_area traffic_area_change before_node conflict For obj: intersecting For obj: at_left intersecting_traffic	

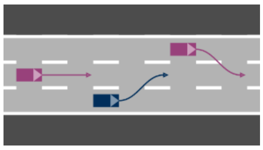
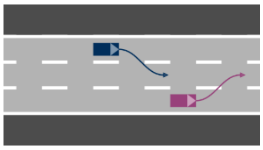
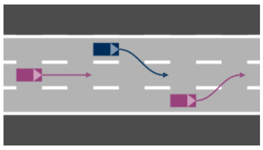
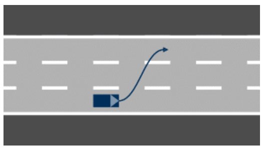
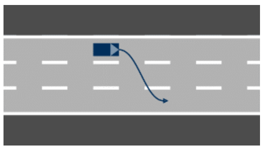
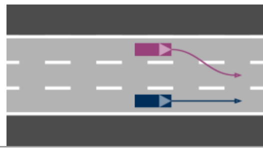
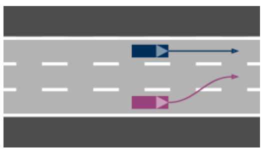
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
`left_turn_with_obj_from_left_entering_after_node`	Linksabbiegen mit nach dem Knotenpunkt eintretendem Objekt	Left turn with object from left entering after the node	conflict_after_node conflict_with_obj_entering_ego_TA conflict_with_obj_from_left_with_TA_change left_turn merging_following_conflict	For obj: enter_ego_traffic_area maneuver merging For ego: left_turn traffic_area_change conflict same_direction For obj: at_left intersecting_traffic behind_node	
`left_turn_with_obj_from_left_entering_before_node`	Linksabbiegen mit vor dem Knotenpunkt eintretendem Objekt	Left turn with object from right entering before the node	conflict_before_node conflict_with_obj_entering_ego_TA conflict_with_obj_from_left_with_TA_change left_turn merging_following_conflict	For obj: enter_ego_traffic_area maneuver merging For ego: left_turn traffic_area_change before_node conflict same_direction For obj: at_left intersecting_traffic	
`left_turn_with_obj_from_left_making_a_u-turn`	Linksabbiegen mit wendendem Objekt von links	Left turn with object from left making a u-turn	conflict_with_obj_from_left_arm conflict_with_obj_making_a_u-turn left_turn merging_following_conflict	For obj: stay_in_traffic_area maneuver merging For ego: left_turn For obj: u-turn conflict same_direction For obj: at_left intersecting_traffic For obj: different_arms	
`left_turn_with_obj_from_left_passing_straight_intersecting`	Linksabbiegen mit kreuzendem Objekt von links bei kreuzendem Konflikt	Left turn with object from left passing straight with intersecting conflict	intersecting_conflict left_turn_with_obj_from_left_passing_straight	For obj: stay_in_traffic_area maneuver For obj: passing_straight For ego: left_turn conflict For obj: intersecting For obj: at_left For obj: different_arms intersecting_traffic	
`left_turn_with_obj_from_left_passing_straight_merging`	Linksabbiegen mit kreuzendem Objekt von links bei zusammenführen dem Konflikt	Left turn with object from left passing straight with merging conflict	left_turn_with_obj_from_left_passing_straight merging_oncoming_conflict	For obj: stay_in_traffic_area maneuver For obj: passing_straight opposite_direction For ego: left_turn merging conflict For obj: at_left For obj: different_arms intersecting_traffic	

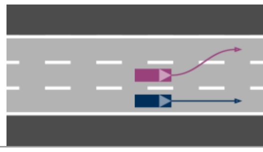
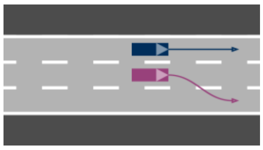
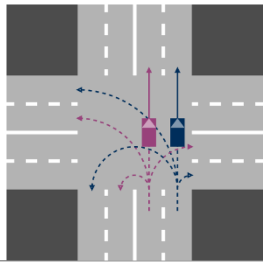
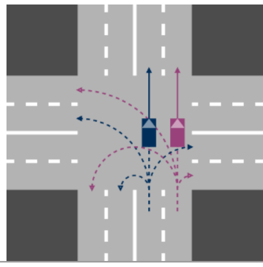
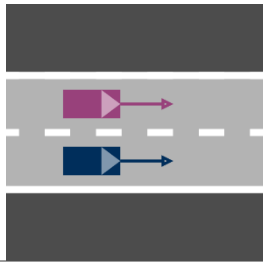
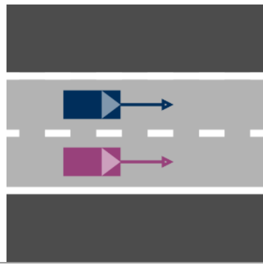
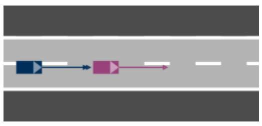
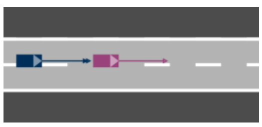
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
left_turn_with_obj_from_left_turning_left	Linksabbiegen mit linksabbiegendem Objekt von links	Left turn with object from left turning left	conflict_with_obj_from_left_arm conflict_with_obj_turning_left intersecting_conflict left_turn	For obj: stay_in_traffic_area maneuver For ego: left_turn For obj: left_turn conflict For obj: intersecting For obj: at_left intersecting_traffic For obj: different_arms	
left_turn_with_obj_from_left_turning_right	Linksabbiegen mit linksabbiegendem Objekt von rechts	Left turn with object from left turning right	conflict_with_obj_from_left_arm conflict_with_obj_turning_right left_turn oncoming_conflict	For obj: stay_in_traffic_area maneuver For obj: right_turn For ego: left_turn conflict For obj: at_left For obj: different_arms intersecting_traffic For obj: oncoming	
left_turn_with_obj_from_right_crossing_after_node	Linksabbiegen mit kreuzendem Objekt von rechts nach dem Knoten	Left turn with object from right crossing after the node	conflict_after_node conflict_with_obj_crossing_ego_TA conflict_with_obj_from_right_with_TA_change intersecting_conflict left_turn	maneuver For ego: left_turn For obj: cross_ego_traffic_area traffic_area_change conflict For obj: intersecting For obj: at_right intersecting_traffic behind_node	
left_turn_with_obj_from_right_crossing_before_node	Linksabbiegen mit kreuzendem Objekt von rechts vor dem Knoten	Left turn with object from right crossing before the node	conflict_before_node conflict_with_obj_crossing_ego_TA conflict_with_obj_from_right_with_TA_change intersecting_conflict left_turn	maneuver For ego: left_turn For obj: cross_ego_traffic_area traffic_area_change before_node conflict For obj: intersecting For obj: at_right intersecting_traffic	
left_turn_with_obj_from_right_entering_after_node	Linksabbiegen mit eintretendem Objekt von rechts nach dem Knoten	Left turn with object from right entering after the node	conflict_after_node conflict_with_obj_entering_ego_TA conflict_with_obj_from_right_with_TA_change left_turn merging_following_conflict	For obj: enter_ego_traffic_area maneuver merging For ego: left_turn traffic_area_change conflict same_direction For obj: at_right intersecting_traffic behind_node	

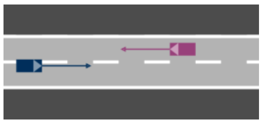
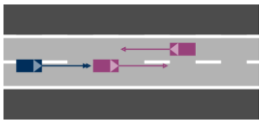
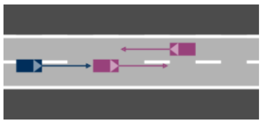
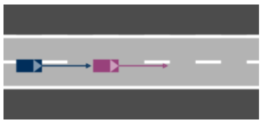
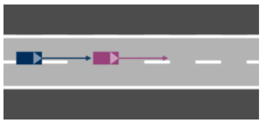
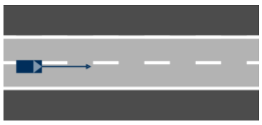
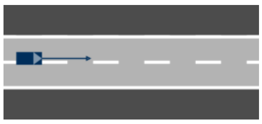
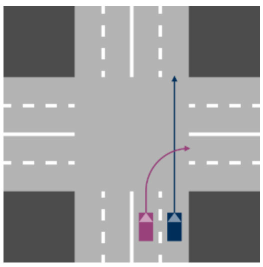
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
left_turn_with_obj_from_right_entering_before_node	Linksabbiegen mit eintretendem Objekt von rechts vor dem Knoten	Left turn with object from right entering before the node	conflict_before_node conflict_with_obj_entering_ego_TA conflict_with_obj_from_right_with_TA_change left_turn merging_following_conflict	For obj: enter_ego_traffic_area maneuver merging For ego: left_turn before_node traffic_area_change conflict same_direction For obj: at_right intersecting_traffic	
left_turn_with_obj_from_right_making_a_u-turn	Linksabbiegen mit wendendem Objekt von rechts	Left turn with object from right making a u-turn	conflict_with_obj_from_right_arm conflict_with_obj_making_a_u-turn left_turn touching_oncoming_conflict	For obj: stay_in_traffic_area maneuver For ego: left_turn For obj: u-turn touching conflict For obj: at_right For obj: different_arms intersecting_traffic opposite_direction	
left_turn_with_obj_from_right_passing_straight	Linksabbiegen mit kreuzendem Objekt von rechts	Left turn with object from right passing straight	conflict_with_obj_from_right_arm conflict_with_obj_passing_straight left_turn merging_following_conflict	For obj: stay_in_traffic_area maneuver For obj: passing_straight merging For ego: left_turn conflict same_direction For obj: at_right For obj: different_arms intersecting_traffic	
left_turn_with_obj_from_right_turning_left	Linksabbiegen mit linksabbiegendem Objekt von rechts	Left turn with object from right turning left	conflict_with_obj_from_right_arm conflict_with_obj_turning_left left_turn touching_following_conflict	For obj: stay_in_traffic_area maneuver same_direction For ego: left_turn For obj: left_turn touching conflict For obj: at_right For obj: different_arms intersecting_traffic	
left_turn_with_obj_from_right_turning_right	Linksabbiegen mit rechtsabbiegendem Objekt von rechts	Left turn with object from right turning right	conflict_with_obj_from_right_arm conflict_with_obj_turning_right left_turn touching_following_conflict	For obj: stay_in_traffic_area maneuver same_direction For obj: right_turn For ego: left_turn touching conflict For obj: at_right For obj: different_arms intersecting_traffic	

Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
left_turn_with_oncoming_obj_making_a_u-turn	Linksabbiegen mit wendenden entgegenkommenden Objekt	Left turn with oncoming object making a u-turn	conflict_with_obj_making_a_u-turn conflict_with_oncoming_obj left_turn touching_oncoming_conflict	For obj: stay_in_traffic_area maneuver For ego: left_turn For obj: u-turn touching conflict For obj: in_front_of For obj: different_arms intersecting_traffic opposite_direction	
left_turn_with_oncoming_obj_passing_straight_intersecting	Linksabbiegen mit kreuzendem entgegenkommenden Objekt bei kreuzendem Konflikt	Left turn with oncoming object passing straight with intersecting conflict	intersecting_conflict left_turn_with_oncoming_obj_passing_straight	For obj: stay_in_traffic_area maneuver For obj: passing_straight For ego: left_turn conflict For obj: intersecting For obj: in_front_of intersecting_traffic For obj: different_arms	
left_turn_with_oncoming_obj_passing_straight_oncoming	Linksabbiegen mit kreuzendem entgegenkommenden Objekt bei entgegenkommenden Konflikt	Left turn with oncoming object passing straight with oncoming conflict	left_turn_with_oncoming_obj_passing_straight oncoming_conflict	For obj: stay_in_traffic_area maneuver For obj: passing_straight For ego: left_turn conflict For obj: in_front_of For obj: different_arms intersecting_traffic For obj: oncoming	
left_turn_with_oncoming_obj_turning_left	Linksabbiegen mit linksabbiegendem entgegenkommenden Objekt von links	Left turn with oncoming object turning left	conflict_with_obj_turning_left conflict_with_oncoming_obj left_turn touching_oncoming_conflict	For obj: stay_in_traffic_area maneuver For ego: left_turn For obj: left_turn touching conflict For obj: in_front_of For obj: different_arms intersecting_traffic opposite_direction	
left_turn_with_oncoming_obj_turning_right	Linksabbiegen mit rechtsabbiegendem entgegenkommenden Objekt	Left turn with oncoming object turning right	conflict_with_obj_turning_right conflict_with_oncoming_obj left_turn merging_oncoming_conflict	For obj: stay_in_traffic_area maneuver For obj: right_turn opposite_direction For ego: left_turn merging conflict For obj: in_front_of For obj: different_arms	

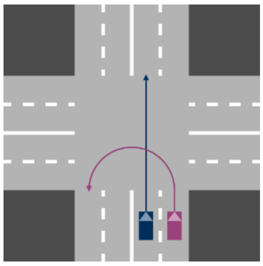
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
				intersecting_ traffic	
merging_cut_ through_l_0	Zusammenführen der Durchscherer nach links ohne Hinterherfahrenden	Merging cut through left without following vehicle	merging_cut_ through_0 merging_cut_ through_l	For obj: exiting For ego: to_left merging For ego: lane_ change For obj: entering For obj: stay_ in_traffic_area free transition longitudinal_ traffic For obj: same_ direction For obj: stay_ in_traffic_area	
merging_cut_ through_l_2	Zusammenführen der Durchscherer nach links mit Hinterherfahrendem	Merging cut through left with following vehicle	merging_cut_ through_2 merging_cut_ through_l	For obj: exiting For ego: to_left merging For ego: lane_ change For obj: entering For obj: stay_ in_traffic_area transition longitudinal_ traffic For obj: behind For obj: same_ direction For obj: stay_ in_traffic_area	
merging_cut_ through_r_0	Zusammenführen der Durchscherer nach rechts ohne Hinterherfahrenden	Merging cut through right without following vehicle	merging_cut_ through_0 merging_cut_ through_r	For ego: to_ right For obj: exiting merging For ego: lane_ change For obj: entering For obj: stay_ in_traffic_area free transition longitudinal_ traffic For obj: same_ direction For obj: stay_ in_traffic_area	
merging_cut_ through_r_2	Zusammenführen der Durchscherer nach rechts mit Hinterherfahrendem	Merging cut through right with following vehicle	merging_cut_ through_2 merging_cut_ through_r	For ego: to_ right For obj: exiting merging For ego: lane_ change For obj: entering For obj: stay_ in_traffic_area transition longitudinal_ traffic For obj: behind For obj: same_ direction For obj: stay_ in_traffic_area	
merging_lcs_l_ 0	Zusammenführen de Fahrstreifenwechsel nach links ohne Hinterherfahrenden	Merging lane changes left without following object	merging_lcs_0 merging_lcs_l	merging For obj: same_ direction transition For ego: to_left free For obj: entering For ego: lane_	

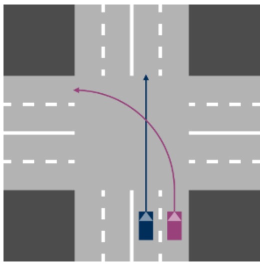
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
				change longitundial_traffic For obj: stay_in_traffic_area	
merging_lcs_l_2	Zusammenführende Fahrstreifenwechsel nach links mit Hinterherfahrenden	Merging lane changes left with following object	merging_lcs_2 merging_lcs_l	merging For obj: same_direction transition For ego: to_left For obj: entering For ego: lane_change longitundial_traffic For obj: behind For obj: stay_in_traffic_area	
merging_lcs_r_0	Zusammenführende Fahrstreifenwechsel nach rechts ohne Hinterherfahrenden	Merging lane changes right without following object	merging_lcs_0 merging_lcs_r	merging For obj: same_direction For ego: to_right transition free For obj: entering For ego: lane_change longitundial_traffic For obj: stay_in_traffic_area	
merging_lcs_r_2	Zusammenführende Fahrstreifenwechsel nach rechts mit Hinterherfahrenden	Merging lane changes right with following object	merging_lcs_2 merging_lcs_r	merging For obj: same_direction For ego: to_right transition For obj: entering For ego: lane_change longitundial_traffic For obj: behind For obj: stay_in_traffic_area	
multi_lcs_l	Mehrfacher Fahrstreifenwechsel nach links	Multiple lane changes left	lc_left multi_lcs	For ego: to_left transition For ego: lane_change multi_lc longitundial_traffic	
multi_lcs_r	Mehrfacher Fahrstreifenwechsel nach rechts	Multiple lane changes right	lc_right multi_lcs	transition For ego: to_right For ego: lane_change multi_lc longitundial_traffic	
neighbor_entering_l	Eintretendes Fahrzeug auf den Nachbarfahrstreifen links	Object entering to the left side of ego	neighbor_entering	overlay longitundial_traffic For obj: having_neighbor For obj: at_left	
neighbor_entering_r	Eintretendes Fahrzeug auf den Nachbarfahrstreifen rechts	Object entering to the right side of ego	neighbor_entering	overlay For obj: at_right longitundial_traffic For obj: having_neighbor	

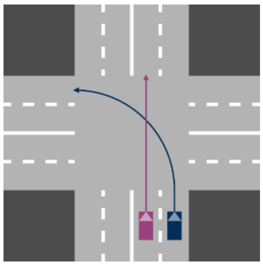
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
neighbor_exiting_l	Vom Nachbarfahrstreifen nach links austretendes Fahrzeug	Object exiting from the left side of ego	neighbor_exiting	overlay For obj: having_neighbor longitunidal_traffic For obj: to_left	
neighbor_exiting_r	Vom Nachbarfahrstreifen nach rechts austretendes Fahrzeug	Object exiting from the right side of ego	neighbor_exiting	overlay For obj: to_right For obj: having_neighbor longitunidal_traffic	
neighbor_in_intersection_left	Laterales Verweilen eines anderen Fahrzeugs in der Kreuzung links des Ego-Fahrzeugs	Object staying left in intersection	neighbor_in_intersection	overlay For obj: at_left For ego: having_neighbor intersecting_traffic same_arm	
neighbor_in_intersection_right	Laterales Verweilen eines anderen Fahrzeugs in der Kreuzung rechts des Ego-Fahrzeugs	Object staying right in intersection	neighbor_in_intersection	overlay For ego: having_neighbor For obj: at_right intersecting_traffic same_arm	
neighbour_l	Laterales Verweilen eines anderen Fahrzeugs links des Ego-Fahrzeugs	Object staying left	neighbor	For obj: at_left overlay For ego: having_neighbor longitudinal_traffic	
neighbour_r	Laterales Verweilen eines anderen Fahrzeugs rechts des Ego-Fahrzeugs	Object staying right	neighbor	overlay For ego: having_neighbor For obj: at_right longitudinal_traffic	
overlap_lane_approaching_lead_l	Den Fahrstreifen links überlappend sich einem Vorausfahrenden annähern	Overlap lane approaching a leading object left	overlap_lane_left overlap_lane_with_leading	transition For obj: overlap_lane For ego: at_left longitudinal_traffic For ego: approaching For obj: leading	
overlap_lane_approaching_lead_r	Den Fahrstreifen rechts überlappend sich einem Vorausfahrenden annähern	Overlap lane approaching a leading object right	overlap_lane_right overlap_lane_with_leading	transition For ego: at_right For ego: approaching For obj: overlap_lane longitudinal_traffic For obj: leading	

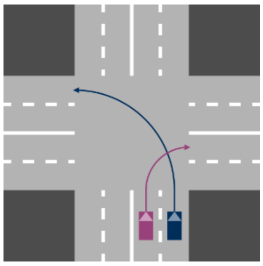
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
overlap_lane_approaching_oncoming	Den Fahrstreifen überlappend sich einem Entgegenkommenden annähern	Overlap lane approaching an oncoming object	overlap_lane_left overlap_lane_with_oncoming	transition For ego: approaching For obj: overlap_lane For ego: at_left longitunidal_traffic For obj: oncoming	
overlap_lane_approaching_oncoming_approaching_lead	Den Fahrstreifen überlappend sich einem Entgegenkommenden und einem Vorausfahrenden annähern	Overlap lane approaching an oncoming and a leading object	overlap_lane_left overlap_lane_with_leading overlap_lane_with_oncoming	transition For obj: overlap_lane For ego: at_left For ego: approaching longitunidal_traffic For obj: oncoming For obj: leading	
overlap_lane_approaching_oncoming_following_lead	Den Fahrstreifen überlappend einem Vorausfahrenden folgen	Overlap lane approaching an oncoming and following a leading object	overlap_lane_left overlap_lane_with_leading overlap_lane_with_oncoming	For ego: following transition For obj: overlap_lane For ego: at_left longitunidal_traffic For obj: oncoming For obj: leading	
overlap_lane_following_lead_l	Fahrstreifen links überlappen und einem Vorausfahrenden folgen	Overlap lane following a leading object left	overlap_lane_left overlap_lane_with_leading	transition For obj: overlap_lane For ego: at_left For obj: same_direction longitunidal_traffic For obj: leading	
overlap_lane_following_lead_r	Fahrstreifen rechts überlappen und einem Vorausfahrenden folgen	Overlap lane following a leading object right	overlap_lane_right overlap_lane_with_leading	transition For ego: at_right For obj: overlap_lane longitunidal_traffic For ego: following For obj: leading	
overlap_lane_free_l	Den Fahrstreifen frei nach links überlappen	Overlap lane free to the left	overlap_lane_free overlap_lane_left	transition For ego: free For obj: overlap_lane For ego: at_left longitunidal_traffic	
overlap_lane_free_r	Den Fahrstreifen frei nach rechts überlappen	Overlap lane free to the right	overlap_lane_free overlap_lane_right	transition For ego: free For ego: at_right For obj: overlap_lane longitunidal_traffic	
parallel_entry_passing_straight_with_obj_left_turning_right	Kreuzen mit Konflikt mit zeitgleich eintretendem rechtsabbiegendem Objekt von links	Parallel entry passing straight with object from left turning right	conflict_with_obj_turning_right conflict_with_parallel_entry_from_left intersecting_conflict pass_straight	For obj: stay_in_traffic_area maneuver For obj: right_turn intersecting_traffic same_arm conflict For obj: intersecting For ego: passing_straight For obj: at_left	

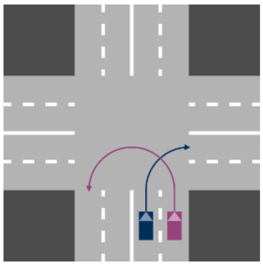
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image

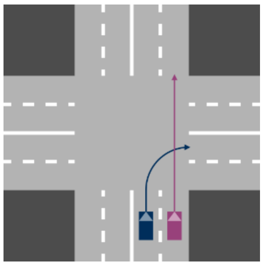
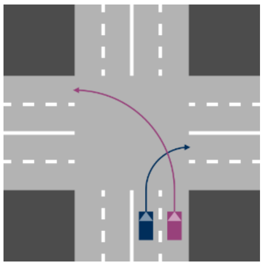
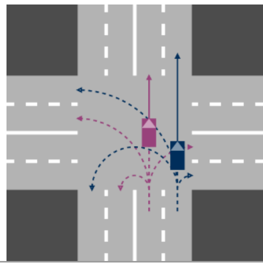
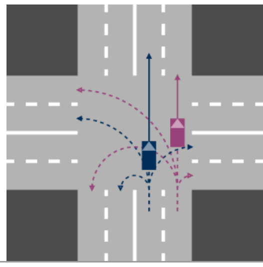
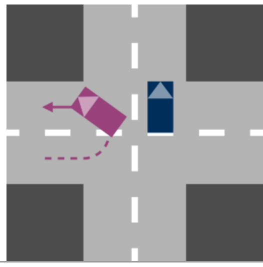
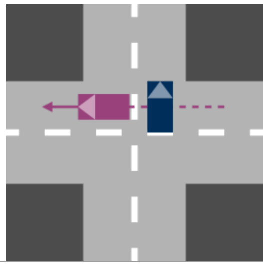
parallel_entry_passing_straight_with_obj_right_making_u-turn	Kreuzen mit Konflikt mit zeitgleich eintretendem wendendem Objekt von rechts	Parallel entry passing straight with object from right making u-turn	`conflict_with_obj_making_a_u-turn`
			`conflict_with_parallel_entry_from_right`
			`intersecting_conflict`
			`pass_straight`
				`For obj: stay_in_traffic_area maneuver`
				`For obj: u-turn same_arm conflict`
				`For obj: intersecting`
				`For ego: passing_straight`
				`For obj: at_right intersecting_traffic`
					

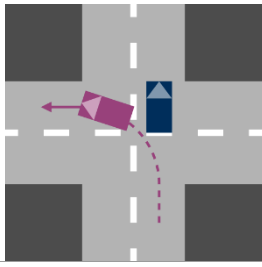
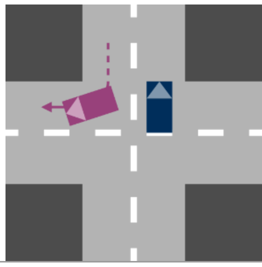
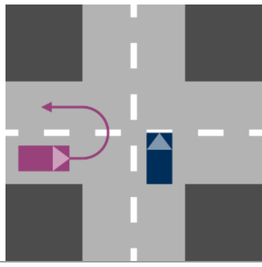
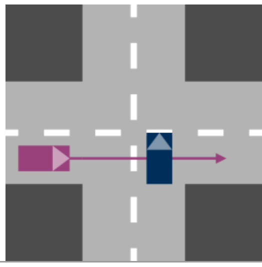
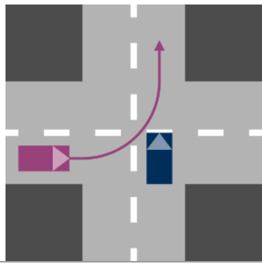
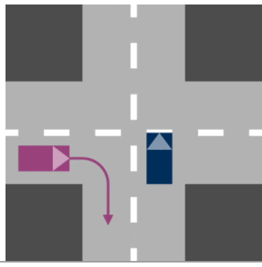
parallel_entry_passing_straight_with_obj_right_turning_left	Kreuzen mit Konflikt mit zeitgleich eintretendem linksabbiegendem Objekt von rechts	Parallel entry passing straight with object from right turning left	`conflict_with_obj_turning_left`
			`conflict_with_parallel_entry_from_right`
			`intersecting_conflict`
			`pass_straight`
				`For obj: stay_in_traffic_area maneuver`
				`For obj: left_turn same_arm conflict`
				`For obj: intersecting`
				`For ego: passing_straight`
				`For obj: at_right intersecting_traffic`
					

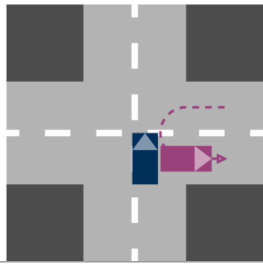
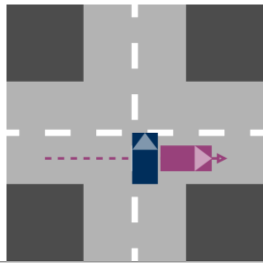
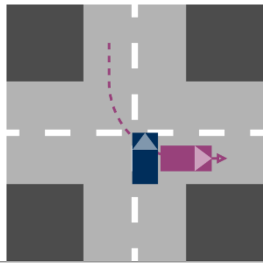
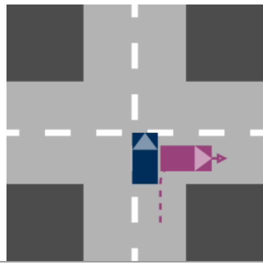
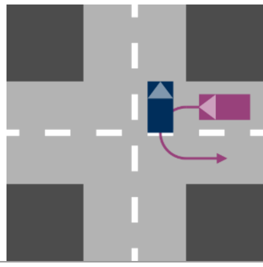
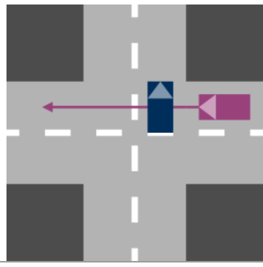
parallel_entry_turning_left_with_obj_left_passing_straight	Linksabbiegen mit Konflikt mit zeitgleich eintretendem kreuzendem Objekt von links	Parallel entry turning left with object from left passing straight	`conflict_with_obj_passing_straight`
			`conflict_with_parallel_entry_from_left`
			`intersecting_conflict`
			`left_turn`
				`For obj: stay_in_traffic_area maneuver`
				`For obj: passing_straight`
				`For ego: left_turn same_arm conflict`
				`For obj: intersecting`
				`For obj: at_left intersecting_traffic`
					

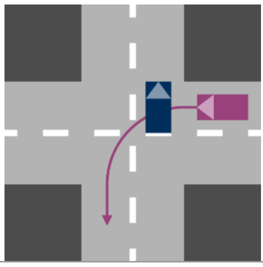
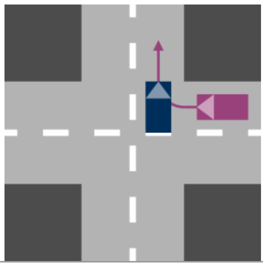
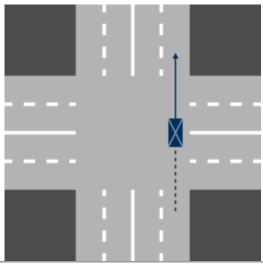
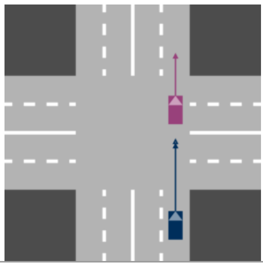
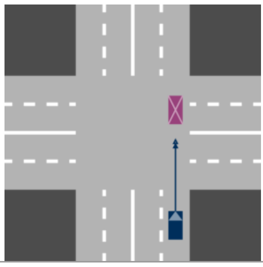
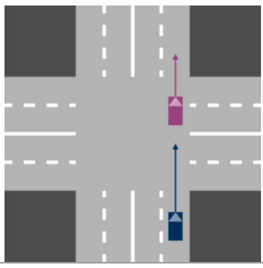
parallel_entry_turning_left_with_obj_left_turning_right	Linksabbiegen mit Konflikt mit zeitgleich eintretendem rechtsabbiegendem Objekt von links	Parallel entry turning left with object from left turning right	`conflict_with_obj_turning_right`
			`conflict_with_parallel_entry_from_left`
			`intersecting_conflict`
			`left_turn`
				`For obj: stay_in_traffic_area maneuver`
				`For obj: right_turn`
				`For ego: left_turn same_arm conflict`
				`For obj: intersecting`
				`For obj: at_left intersecting_traffic`
					

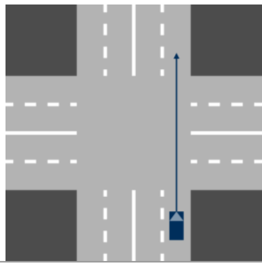
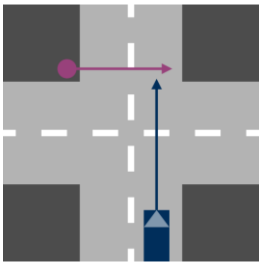
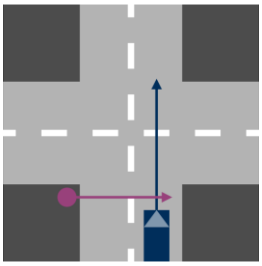
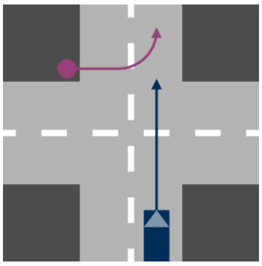
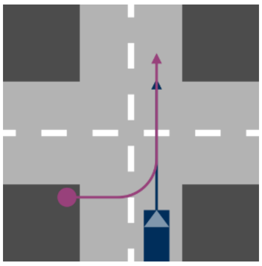
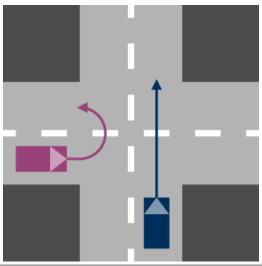
parallel_entry_turning_right_with_obj_right_making_u-turn	Rechtsabbiegen mit Konflikt mit zeitgleich eintretendem wendenden Objekt von rechts	Parallel entry turning right with object from right making u-turn	`conflict_with_obj_making_a_u-turn`
			`conflict_with_parallel_entry_from_right`
			`intersecting_conflict`
			`right_turn`
				`For obj: stay_in_traffic_area maneuver`
				`For obj: u-turn same_arm`
				`For ego: right_turn conflict`
				`For obj: intersecting`
				`For obj: at_right intersecting_traffic`
					

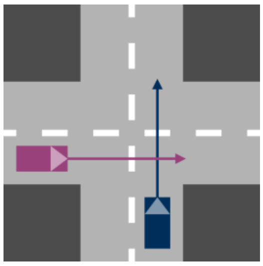
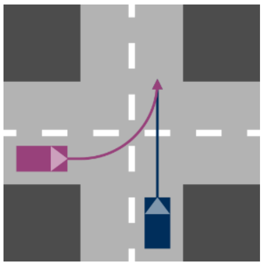
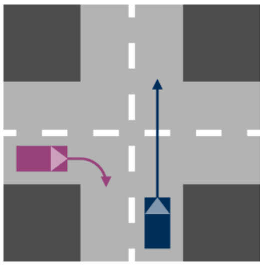
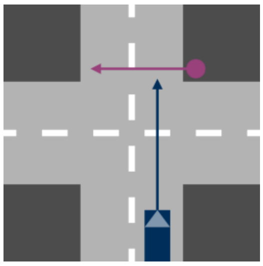
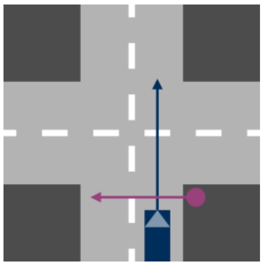
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
parallel_entry_turning_right_with_obj_right_passing_straight	Rechtsabbiegen mit Konflikt mit zeitgleich eintretendem kreuzendem Objekt von rechts	Parallel entry turning right with object from right passing straight	conflict_with_obj_passing_straight conflict_with_parallel_entry_from_right intersecting_conflict right_turn	For obj: stay_in_traffic_area maneuver For obj: passing_straight same_arm For ego: right_turn conflict For obj: intersecting For obj: at_right intersecting_traffic	
parallel_entry_turning_right_with_obj_right_turning_left	Rechtsabbiegen mit Konflikt mit zeitgleich eintretendem linksabbiegendem Objekt von rechts	Parallel entry turning right with object from right turning left	conflict_with_obj_turning_left conflict_with_parallel_entry_from_right intersecting_conflict right_turn	For obj: stay_in_traffic_area maneuver For obj: left_turn same_arm For ego: right_turn conflict For obj: intersecting For obj: at_right intersecting_traffic	
pass_obj_in_intersection_parallel_left	Parallele Vorbeifahrt an einem Objekt in der Kreuzung links	Pass object on the left parallel in intersection	pass_obj_in_intersection_parallel	overlay same_arm intersecting_traffic For obj: at_left For ego: pass	
pass_obj_in_intersection_parallel_right	Parallele Vorbeifahrt an einem Objekt in der Kreuzung rechts	Pass object on the right parallel in intersection	pass_obj_in_intersection_parallel	overlay same_arm For obj: at_right intersecting_traffic For ego: pass	
pass_obj_left_moving_away_making_u-turn	Vorbeifahrt an einem sich entfernenden, wendenden Objekt links	Pass object on the left moving away doing a u-turn	pass_obj_in_intersection_left pass_obj_in_intersection_moving_away pass_obj_making_u-turn	overlay different_arms For obj: u-turn For obj: away intersecting_traffic For obj: at_left	
pass_obj_left_moving_away_passing_straight	Vorbeifahrt an einem sich entfernenden, kreuzenden Objekt links	Pass object on the left moving away passing straight	pass_obj_in_intersection_left pass_obj_in_intersection_moving_away pass_obj_passing_straight_in_intersection	overlay For obj: passing_straight different_arms For obj: away intersecting_traffic For obj: at_left	

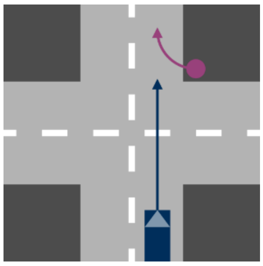
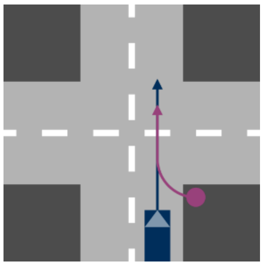
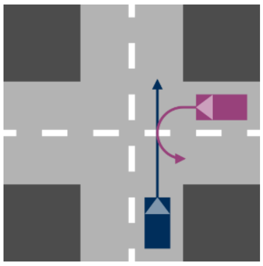
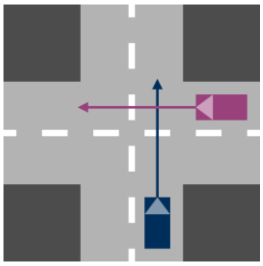
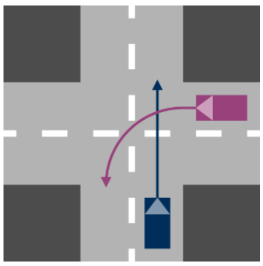
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
pass_obj_left_moving_away_turning_left	Vorbeifahrt an einem sich entfernenden, linksabbiegenden Objekt links	Pass object on the left moving away turning left	pass_obj_in_intersection_left pass_obj_in_intersection_moving_away pass_obj_turning_left_in_intersection	overlay different_arms For obj: left_turn For obj: away intersecting_traffic For obj: at_left	
pass_obj_left_moving_away_turning_right	Vorbeifahrt an einem sich entfernenden, rechtsabbiegenden Objekt links	Pass object on the left moving away turning right	pass_obj_in_intersection_left pass_obj_in_intersection_moving_away pass_obj_turning_right_in_intersection	overlay different_arms For obj: right_turn For obj: away intersecting_traffic For obj: at_left	
pass_obj_left_moving_toward_making_u-turn	Vorbeifahrt an einem sich nähernden, wendenden Objekt links	Pass object on the left moving towards ego, making u-turn	pass_obj_in_intersection_left pass_obj_in_intersection_moving_toward pass_obj_making_u-turn	overlay For obj: towards different_arms For obj: u-turn intersecting_traffic For obj: at_left	
pass_obj_left_moving_toward_passing_straight	Vorbeifahrt an einem sich nähernden, kreuzenden Objekt links	Pass object on the left moving towards ego, passing straight	pass_obj_in_intersection_left pass_obj_in_intersection_moving_toward pass_obj_passing_straight_in_intersection	overlay For obj: towards intersecting_traffic For obj: passing_straight different_arms For obj: at_left	
pass_obj_left_moving_toward_turning_left	Vorbeifahrt an einem sich nähernden, linksabbiegenden Objekt links	Pass object on the left moving towards ego, turning left	pass_obj_in_intersection_left pass_obj_in_intersection_moving_toward pass_obj_turning_left_in_intersection	overlay For obj: towards different_arms For obj: left_turn intersecting_traffic For obj: at_left	
pass_obj_left_moving_toward_turning_right	Vorbeifahrt an einem sich nähernden, rechtsabbiegenden Objekt links	Pass object on the left moving towards ego, turning right	pass_obj_in_intersection_left pass_obj_in_intersection_moving_toward pass_obj_turning_right_in_intersection	overlay For obj: towards different_arms For obj: right_turn intersecting_traffic For obj: at_left	

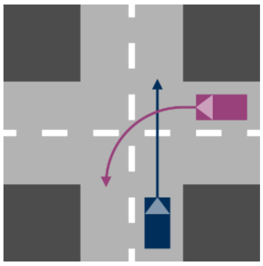
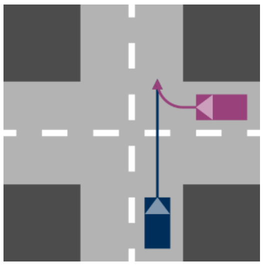
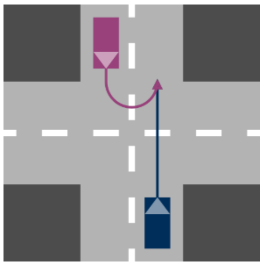
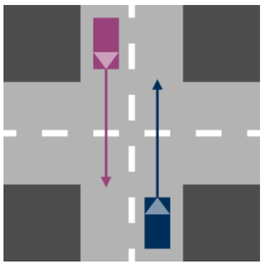
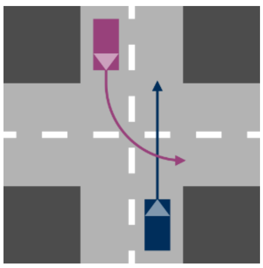
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
pass_obj_right_moving_away_making_u-turn	Vorbeifahrt an einem sich entfernenden, wendenden Objekt rechts	Pass object on the right moving away from ego, making u-turn	pass_obj_in_intersection_moving_away pass_obj_in_intersection_right pass_obj_making_u-turn	overlay different_arms For obj: u-turn For obj: away For obj: at_right intersecting_traffic	
pass_obj_right_moving_away_passing_straight	Vorbeifahrt an einem sich entfernenden, kreuzenden Objekt rechts	Pass object on the right moving away from ego, passing straight	pass_obj_in_intersection_moving_away pass_obj_in_intersection_right pass_obj_passing_straight_in_intersection	overlay different_arms For obj: away For obj: at_right intersecting_traffic For obj: passing_straight	
pass_obj_right_moving_away_turning_left	Vorbeifahrt an einem sich entfernenden, linksabbiegenden Objekt rechts	Pass object on the right moving away from ego, turning left	pass_obj_in_intersection_moving_away pass_obj_in_intersection_right pass_obj_turning_left_in_intersection	overlay different_arms For obj: left_turn For obj: away For obj: at_right intersecting_traffic	
pass_obj_right_moving_away_turning_right	Vorbeifahrt an einem sich entfernenden, rechtsabbiegenden Objekt rechts	Pass object on the right moving away from ego, turning right	pass_obj_in_intersection_moving_away pass_obj_in_intersection_right pass_obj_turning_right_in_intersection	overlay different_arms For obj: right_turn For obj: away For obj: at_right intersecting_traffic	
pass_obj_right_moving_toward_making_u-turn	Vorbeifahrt an einem sich nähernden, wendenden Objekt rechts	Pass object on the right moving towards ego, making u-turn	pass_obj_in_intersection_moving_toward pass_obj_in_intersection_right pass_obj_making_u-turn	overlay For obj: towards different_arms For obj: u-turn For obj: at_right intersecting_traffic	
pass_obj_right_moving_toward_passing_straight	Vorbeifahrt an einem sich nähernden, kreuzenden Objekt rechts	Pass object on the right moving towards ego, passing straight	pass_obj_in_intersection_moving_toward pass_obj_in_intersection_right pass_obj_passing_straight_in_intersection	overlay For obj: towards For obj: passing_straight different_arms For obj: at_right intersecting_traffic	

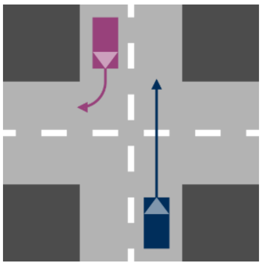
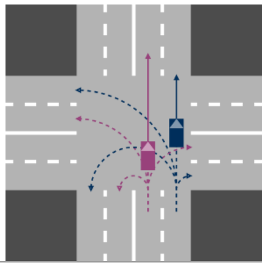
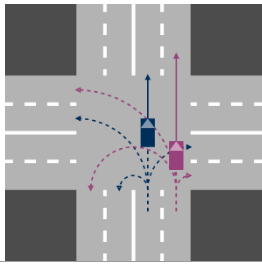
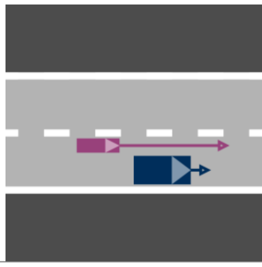
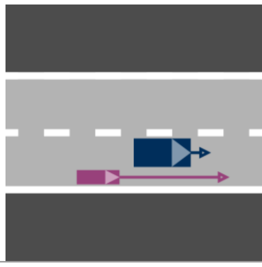
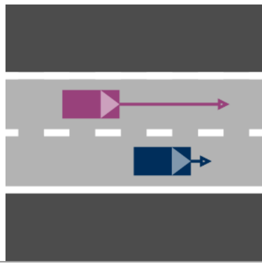
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
pass_obj_right_moving_toward_turning_left	Vorbeifahrt an einem sich nähernden, linksabbiegenden Objekt rechts	Pass object on the right moving towards ego, turning left	pass_obj_in_intersection_moving_toward pass_obj_in_intersection_right pass_obj_turning_left_in_intersection	overlay For obj: towards different_arms For obj: left_turn For obj: at_right intersecting_traffic	
pass_obj_right_moving_toward_turning_right	Vorbeifahrt an einem sich nähernden, rechtsabbiegenden Objekt rechts	Pass object on the right moving towards ego, turning right	pass_obj_in_intersection_moving_toward pass_obj_in_intersection_right pass_obj_turning_right_in_intersection	overlay For obj: towards different_arms For obj: right_turn For obj: at_right intersecting_traffic	
pass_standstill	Vorbeifahrt an einem stehenden Objekt	Pass object standing still	pass_straight standstill_in_intersection	maneuver state For ego: passing_straight For ego: standstill intersecting_traffic	
pass_straight_approaching_lead	Sich einem Vorausfahrenden nähernd kreuzen	Pass straight approaching a leading object	approaching_lead_in_intersection pass_straight	maneuver state For obj: leading For ego: approaching For ego: passing_straight intersecting_traffic	
pass_straight_approaching_static	Sich einem statischen Objekt nähern kreuzen	Pass straight approaching a static object	approaching_static_in_intersection pass_straight	maneuver state For obj: static For ego: approaching For ego: passing_straight intersecting_traffic	
pass_straight_following_lead	Einem Vorausfahrenden folgend kreuzen	Pass straight following a leading object	following_in_intersection pass_straight	maneuver For ego: following state For ego: passing_straight intersecting_traffic	

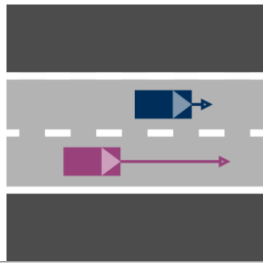
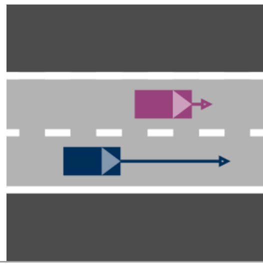
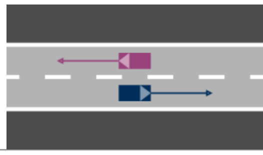
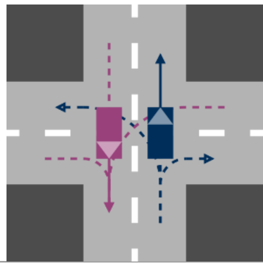
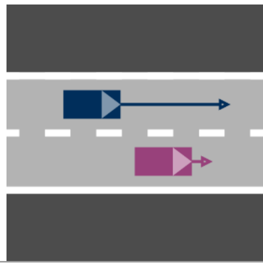
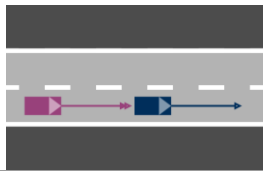
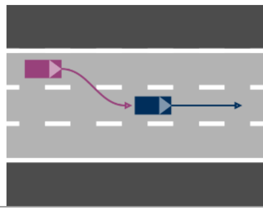
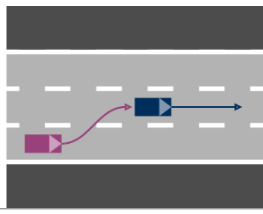
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
pass_straight_free	Freies Kreuzen	Pass straight freely	free_intersection_maneuver pass_straight	maneuver state For ego: free For ego: passing_straight intersecting_traffic	
pass_straight_with_obj_from_left_crossing_after_node	Kreuzen mit nach dem Knotenpunkt kreuzendem Objekt	Pass straight with object from left crossing after the node	conflict_after_node conflict_with_obj_crossing_ego_TA conflict_with_obj_from_left_with_TA_change intersecting_conflict pass_straight	maneuver For obj: cross_ego_traffic_area traffic_area_change conflict For obj: intersecting For ego: passing_straight For obj: at_left intersecting_traffic behind_node	
pass_straight_with_obj_from_left_crossing_before_node	Kreuzen mit vor dem Knotenpunkt kreuzendem Objekt	Pass straight with object from left crossing before the node	conflict_before_node conflict_with_obj_crossing_ego_TA conflict_with_obj_from_left_with_TA_change intersecting_conflict pass_straight	maneuver For obj: cross_ego_traffic_area traffic_area_change before_node conflict For obj: intersecting For ego: passing_straight For obj: at_left intersecting_traffic	
pass_straight_with_obj_from_left_entering_after_node	Kreuzen mit nach dem Knotenpunkt eintretendem Objekt	Pass straight with object from left entering after the node	conflict_after_node conflict_with_obj_entering_ego_TA conflict_with_obj_from_left_with_TA_change merging_following_conflict pass_straight	For obj: enter_ego_traffic_area maneuver merging traffic_area_change conflict same_direction For ego: passing_straight For obj: at_left intersecting_traffic behind_node	
pass_straight_with_obj_from_left_entering_before_node	Kreuzen mit vor dem Knotenpunkt eintretendem Objekt	Pass straight with object from left entering before the node	conflict_before_node conflict_with_obj_entering_ego_TA conflict_with_obj_from_left_with_TA_change merging_following_conflict pass_straight	For obj: enter_ego_traffic_area maneuver merging traffic_area_change before_node conflict same_direction For ego: passing_straight For obj: at_left intersecting_traffic	
pass_straight_with_obj_from_left_making_a_u-turn	Kreuzen mit wendendem Objekt von links	Pass straight with object from left making a u-turn	conflict_with_obj_from_left_arm conflict_with_obj_making_a_u-turn pass_straight touching_following_conflict	For obj: stay_in_traffic_area maneuver same_direction For obj: u-turn touching conflict For ego: passing_straight For obj: at_left For obj: different_arms	

Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
				intersecting_traffic	
pass_straight_with_obj_from_left_passing_straight	Kreuzen mit kreuzendem Objekt von links	Pass straight with object from left passing straight	conflict_with_obj_from_left_arm conflict_with_obj_passing_straight intersecting_conflict pass_straight	For obj: stay_in_traffic_area maneuver For obj: passing_straight conflict For obj: intersecting For ego: passing_straight For obj: at_left For obj: different_arms intersecting_traffic	
pass_straight_with_obj_from_left_turning_left	Kreuzen mit linksabbiegendem Objekt von links	Pass straight with object from left turning left	conflict_with_obj_from_left_arm conflict_with_obj_turning_left merging_following_conflict pass_straight	For obj: stay_in_traffic_area maneuver merging For obj: left_turn conflict same_direction For ego: passing_straight For obj: at_left For obj: different_arms intersecting_traffic	
pass_straight_with_obj_from_left_turning_right	Kreuzen mit linksabbiegendem Objekt von rechts	Pass straight with object from left turning right	conflict_with_obj_from_left_arm conflict_with_obj_turning_right merging_oncoming_conflict pass_straight	For obj: stay_in_traffic_area maneuver For obj: right_turn opposite_direction merging conflict For ego: passing_straight For obj: at_left For obj: different_arms intersecting_traffic	
pass_straight_with_obj_from_right_crossing_after_node	Kreuzen mit kreuzendem Objekt von rechts nach dem Knoten	Pass straight with object from right crossing after the node	conflict_after_node conflict_with_obj_crossing_ego_TA conflict_with_obj_from_right_with_TA_change intersecting_conflict pass_straight	maneuver For obj: cross_ego_traffic_area traffic_area_change conflict For obj: intersecting For ego: passing_straight For obj: at_right intersecting_traffic behind_node	
pass_straight_with_obj_from_right_crossing_before_node	Kreuzen mit kreuzendem Objekt von rechts vor dem Knoten	Pass straight with object from right crossing before the node	conflict_before_node conflict_with_obj_crossing_ego_TA conflict_with_obj_from_right_with_TA_change intersecting_conflict pass_straight	maneuver For obj: cross_ego_traffic_area traffic_area_change before_node conflict For obj: intersecting For ego: passing_straight For obj: at_right intersecting_traffic	

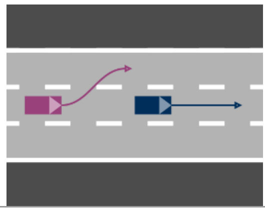
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
pass_straight_with_obj_from_right_entering_after_node	Kreuzen mit eintretendem Objekt von rechts nach dem Knoten	Pass straight with object from right entering after the node	conflict_after_node conflict_with_obj_entering_ego_TA conflict_with_obj_from_right_with_TA_change merging_following_conflict pass_straight	For obj: enter_ego_traffic_area maneuver merging traffic_area_change conflict same_direction For obj: at_right For ego: passing_straight intersecting_traffic behind_node	
pass_straight_with_obj_from_right_entering_before_node	Kreuzen mit eintretendem Objekt von rechts vor dem Knoten	Pass straight with object from right entering before the node	conflict_before_node conflict_with_obj_entering_ego_TA conflict_with_obj_from_right_with_TA_change merging_following_conflict pass_straight	For obj: enter_ego_traffic_area maneuver merging before_node traffic_area_change conflict same_direction For obj: at_right For ego: passing_straight intersecting_traffic	
pass_straight_with_obj_from_right_making_a_u-turn	Kreuzen mit wendendem Objekt von rechts	Pass straight with object from right making a u-turn	conflict_with_obj_from_right_arm conflict_with_obj_making_a_u-turn pass_straight touching_oncoming_conflict	For obj: stay_in_traffic_area maneuver For obj: u-turn touching conflict For ego: passing_straight For obj: at_right For obj: different_arms intersecting_traffic opposite_direction	
pass_straight_with_obj_from_right_passing_straight	Kreuzen mit kreuzendem Objekt von rechts	Pass straight with object from right passing straight	conflict_with_obj_from_right_arm conflict_with_obj_passing_straight intersecting_conflict pass_straight	For obj: stay_in_traffic_area maneuver For obj: passing_straight intersecting_traffic conflict For obj: intersecting For ego: passing_straight For obj: at_right For obj: different_arms	
pass_straight_with_obj_from_right_turning_left_intersecting	Kreuzen mit rechtsabbiegendem Objekt von rechts	Pass straight with object from right turning left with intersecting conflict	intersecting_conflict pass_straight_with_obj_from_right_turning_left	For obj: stay_in_traffic_area maneuver For obj: left_turn conflict For obj: intersecting For ego: passing_straight For obj: at_right For obj: different_arms intersecting_traffic	

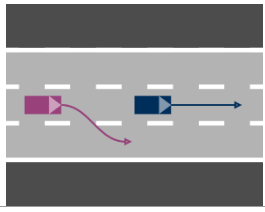
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
pass_straight_with_obj_from_right_turning_left_merging	Kreuzen mit wendenden entgegenkommenden Objekt	Pass straight with object from right turning left with merging conflict	merging_oncoming_conflict pass_straight_with_obj_from_right_turning_left	For obj: stay_in_traffic_area maneuver merging opposite_direction For obj: left_turn conflict For ego: passing_straight For obj: at_right For obj: different_arms intersecting_traffic	
pass_straight_with_obj_from_right_turning_right	Kreuzen mit kreuzendem entgegenkommenden Objekt	Pass straight with object from right turning right	conflict_with_obj_from_right_arm conflict_with_obj_turning_right merging_following_conflict pass_straight	For obj: stay_in_traffic_area maneuver For obj: right_turn merging conflict same_direction For obj: at_right For obj: different_arms For ego: passing_straight intersecting_traffic	
pass_straight_with_oncoming_obj_making_a_u-turn	Kreuzen mit kreuzendem entgegenkommenden Objekt bei kreuzendem Konflikt	Pass straight with oncoming object making a u-turn	conflict_with_obj_making_a_u-turn conflict_with_oncoming_obj merging_following_conflict pass_straight	For obj: stay_in_traffic_area maneuver intersecting_traffic merging For obj: u-turn conflict same_direction For ego: passing_straight For obj: in_front_of For obj: different_arms	
pass_straight_with_oncoming_obj_passing_straight	Kreuzen mit kreuzendem entgegenkommenden Objekt bei entgegenkommenden Konflikt	Pass straight with oncoming object passing straight	conflict_with_obj_passing_straight conflict_with_oncoming_obj oncoming_conflict pass_straight	For obj: stay_in_traffic_area maneuver For obj: passing_straight conflict For ego: passing_straight For obj: in_front_of For obj: different_arms intersecting_traffic For obj: oncoming	
pass_straight_with_oncoming_obj_turning_left	Kreuzen mit linksabbiegendem entgegenkommenden Objekt von links	Pass straight with oncoming object turning left	conflict_with_obj_turning_left conflict_with_oncoming_obj intersecting_conflict pass_straight	For obj: stay_in_traffic_area maneuver For obj: left_turn conflict For obj: intersecting For ego: passing_straight For obj: in_front_of For obj: different_arms intersecting_traffic	

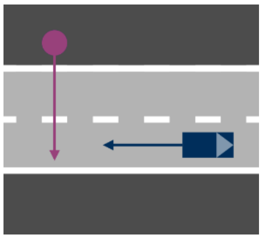
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
pass_straight_with_oncoming_obj_turning_right	Kreuzen mit rechtsabbiegendem entgegenkommenden Objekt	Pass straight with oncoming object turning right	conflict_with_obj_turning_right conflict_with_oncoming_obj oncoming_conflict pass_straight	For obj: stay_in_traffic_area maneuver For obj: right_turn conflict For ego: passing_straight For obj: in_front_of For obj: different_arms intersecting_traffic For obj: oncoming	
passed_in_intersection_left	Vorbeifahrt eines Objekts in der Kreuzung links	An object passing left in intersection	passed_in_itersection	overlay For ego: being_passed For obj: at_left intersecting_traffic same_arm	
passed_in_intersection_right	Vorbeifahrt eines Objekts in der Kreuzung rechts	An object passing right in intersection	passed_in_itersection	overlay For obj: at_right For ego: being_passed intersecting_traffic same_arm	
passed_in_lane_l	Im Fahrstreifen links passiert werden	Object passing in lane left	passed_in_lane	overlay longitunidal_traffic For ego: being_passed_in_lane For obj: at_left	
passed_in_lane_r	Im Fahrstreifen rechts passiert werden	Object passing in lane right	passed_in_lane	For obj: at_right overlay longitunidal_traffic For ego: being_passed_in_lane	
passed_l	Vorbeifahrt eines anderen Fahrzeugs links	An object passing left	passed	overlay For ego: being_passed longitunidal_traffic For obj: at_left	

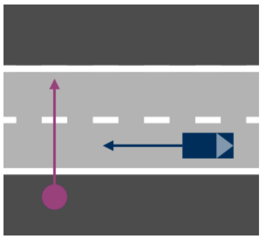
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
passed_r	Vorbeifahrt eines anderen Fahrzeugs rechts	An object passing right	passed	overlay longitunidal_ traffic For ego: being_ passed For obj: at_ right	
passing_l	Vorbeifahrt an einem anderen Fahrzeug links	Passing an object left	passing	overlay For ego: pass For obj: at_left longitunidal_ traffic	
passing_ oncoming	Vorbeifahrt an einem Entgegenkommen den	Passing an oncoming object	passing	overlay For obj: oncoming For ego: pass longitunidal_ traffic	
passing_ oncoming_in_ intersection	Vorbeifahrt an einem Entgegenkommen den in der Kreuzung	Passing an oncoming object in an intersection	pass_obj_in_ intersection_ parallel	overlay same_arm For obj: oncoming intersecting_ traffic For ego: pass	
passing_r	Vorbeifahrt an einem anderen Fahrzeug rechts	Passing an object right	passing	overlay For ego: pass For obj: at_ right longitunidal_ traffic	
rear_obj_ approaching	Annäherung eines Hinterherfahrende n	Rear object approaching	long_traffic_ overlay	overlay For obj: behind longitunidal_ traffic	
rear_obj_ entering_l	Eintritt eines Hinterherfahrende n von links	Rear object entering from left	rear_obj_ entering	For obj: from_ left overlay longitunidal_ traffic For obj: entering	
rear_obj_ entering_r	Eintritt eines Hinterherfahrende n von rechts	Rear object entering from right	rear_obj_ entering	overlay For obj: from_ right longitunidal_ traffic For obj: entering	

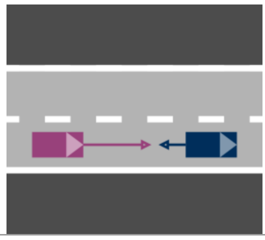
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image

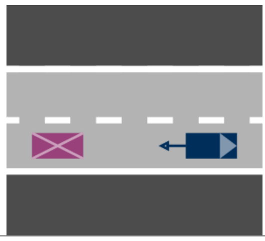
`rear_obj_exiting_l`	Austritt eines Hinterherfahrenden nach links	Rear object exiting to left	`rear_obj_exiting`	`overlay`
				`For obj: exiting`
				`longitudinal_traffic`
				`For obj: to_left`
					

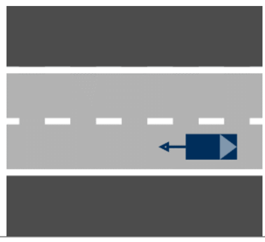
`rear_obj_exiting_r`	Austritt eines Hinterherfahrenden nach rechts	Rear object exiting to right	`rear_obj_exiting`	`overlay`
				`For obj: exiting`
				`For obj: to_right`
				`longitudinal_traffic`
					

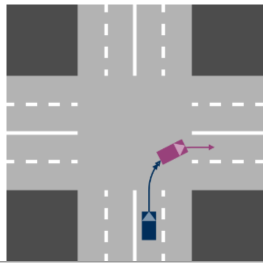
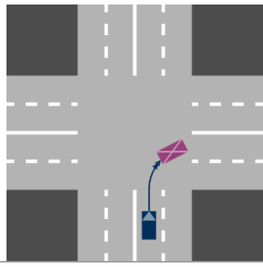
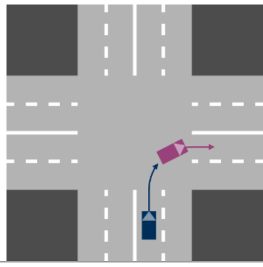
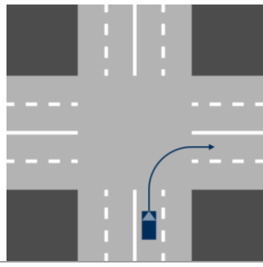
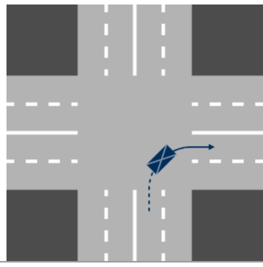
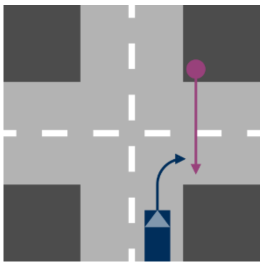
`reverse_approach_lat_l`	Rückwärtsfahrt, sich einem lateral bewegenden Objekt von links annähernd	Reverse, approaching a laterally moving object from left	`reverse_approach_lat`	`For ego: reversing`
				`For obj: lateral_obj`
				`longitudinal_traffic`
				`For ego: approaching state`
				`For obj: from_left`
					

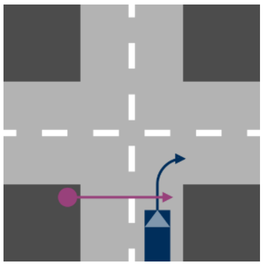
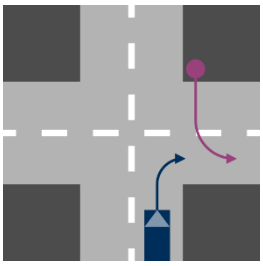
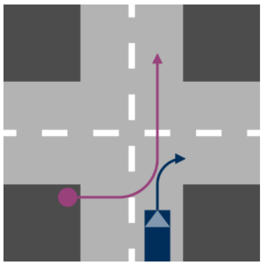
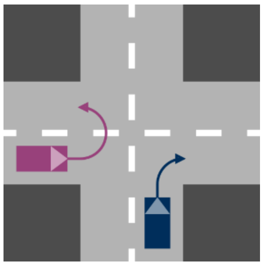
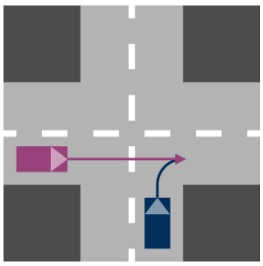
`reverse_approach_lat_r`	Rückwärtsfahrt, sich einem lateral bewegenden Objekt von rechts annähernd	Reverse, approaching a laterally moving object from right	`reverse_approach_lat`	`For ego: reversing`
				`For obj: from_right`
				`For obj: lateral_obj`
				`longitudinal_traffic`
				`For ego: approaching state`
					

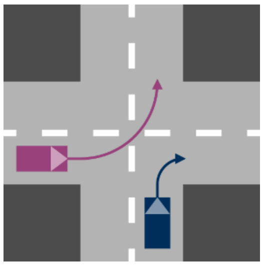
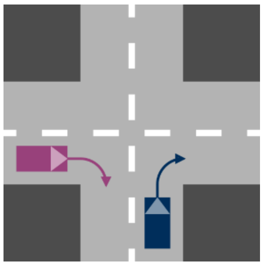
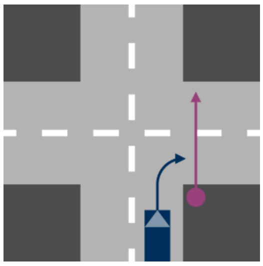
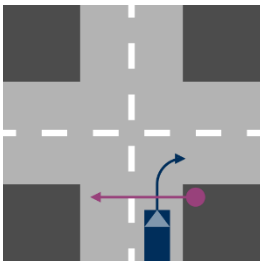
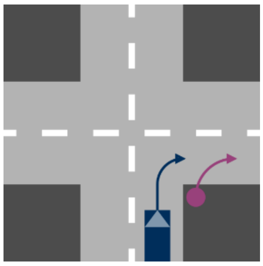
`reverse_approach_oncoming`	Rückwärtsfahrt, sich einem entgegenkommenden Objekt annähernd	Reverse, approaching an oncoming object	`reverse_approach`	`For ego: reversing`
				`For obj: oncoming`
				`longitudinal_traffic`
				`For ego: approaching state`
					

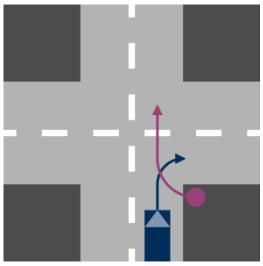
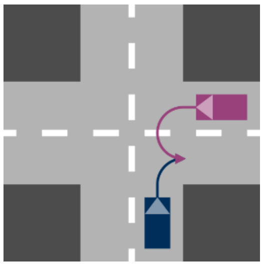
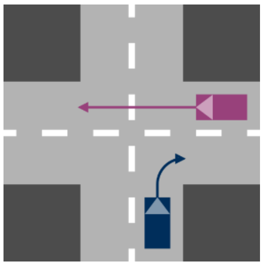
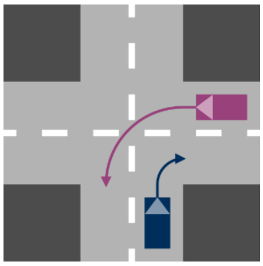
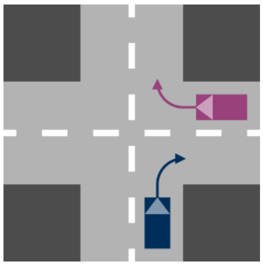
`reverse_approach_static`	Rückwärtsfahrt, sich einem statischen Objekt annähernd	Reverse, approaching a static object	`reverse_approach`	`For ego: reversing`
				`For obj: static`
				`longitudinal_traffic`
				`For ego: approaching state`
					

`reverse_free`	Freie Rückwärtsfahrt	Reverse, approaching a free object	`reverse`	`For ego: reversing`
				`For ego: free`
				`longitudinal_traffic state`
					

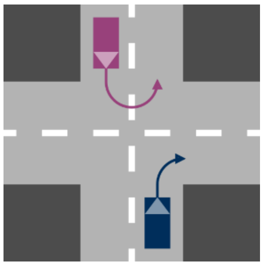
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
right_turn_approaching_lead	Einem Vorausfahrenden beim Rechtsabbiegen annähern	Right turn approaching a leadign object	approaching_lead_in_intersection right_turn	maneuver state For obj: leading For ego: right_turn For ego: approaching intersecting_traffic	
right_turn_approaching_static	Einem statischen Objekt beim Rechtsabbiegen annähern	Right turn approaching a static object	approaching_static_in_intersection right_turn	maneuver state For obj: static For ego: right_turn For ego: approaching intersecting_traffic	
right_turn_following_lead	Einem Vorausfahrenden beim Rechtsabbiegen folgen	Right turn following a leading object	following_in_intersection right_turn	maneuver intersecting_traffic state For ego: right_turn For ego: following	
right_turn_free	Freies Rechtsabbiegen	Free right turn	free_intersection_maneuver right_turn	maneuver state For ego: free For ego: right_turn intersecting_traffic	
right_turn_standstill	Stillstand beim Rechtsabbiegen	Standstill while turning right	right_turn standstill_in_intersection	maneuver state For ego: right_turn For ego: standstill intersecting_traffic	
right_turn_with_obj_from_left_crossing_after_node	Rechtsabbiegen mit nach dem Knotenpunkt kreuzendem Objekt	Right turn with object from left crossing after the node	conflict_after_node conflict_with_obj_crossing_ego_TA conflict_with_obj_from_left_with_TA_change intersecting_conflict right_turn	maneuver For obj: cross_ego_traffic_area traffic_area_change For ego: right_turn conflict For obj: intersecting For obj: at_left intersecting_traffic behind_node	

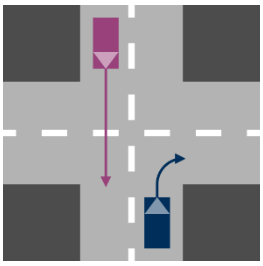
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
right_turn_with_obj_from_left_crossing_before_node	Rechtsabbiegen mit vor dem Knotenpunkt kreuzendem Objekt	Right turn with object from left crossing before the node	conflict_before_node conflict_with_obj_crossing_ego_TA conflict_with_obj_from_left_with_TA_change intersecting_conflict right_turn	maneuver For obj: cross_ego_traffic_area traffic_area_change before_node For ego: right_turn conflict For obj: intersecting For obj: at_left intersecting_traffic	
right_turn_with_obj_from_left_entering_after_node	Rechtsabbiegen mit nach dem Knotenpunkt eintretendem Objekt	Right turn with object from left entering after the node	conflict_after_node conflict_with_obj_entering_ego_TA conflict_with_obj_from_left_with_TA_change merging_following_conflict right_turn	For obj: enter_ego_traffic_area maneuver merging traffic_area_change For ego: right_turn conflict same_direction For obj: at_left intersecting_traffic behind_node	
right_turn_with_obj_from_left_entering_before_node	Rechtsabbiegen mit vor dem Knotenpunkt eintretendem Objekt	Right turn with object from left entering before the node	conflict_before_node conflict_with_obj_entering_ego_TA conflict_with_obj_from_left_with_TA_change merging_following_conflict right_turn	For obj: enter_ego_traffic_area maneuver merging traffic_area_change before_node For ego: right_turn conflict same_direction For obj: at_left intersecting_traffic	
right_turn_with_obj_from_left_making_a_u-turn	Rechtsabbiegen mit wendendem Objekt von links	Right turn with object from left making a u-turn	conflict_with_obj_from_left_arm conflict_with_obj_making_a_u-turn right_turn touching_following_conflict	For obj: stay_in_traffic_area maneuver same_direction For obj: u-turn touching For ego: right_turn conflict For obj: at_left For obj: different_arms intersecting_traffic	
right_turn_with_obj_from_left_passing_straight	Rechtsabbiegen mit kreuzendem Objekt von links	Right turn with object from left passing straight	conflict_with_obj_from_left_arm conflict_with_obj_passing_straight merging_following_conflict right_turn	For obj: stay_in_traffic_area maneuver For obj: passing_straight merging For ego: right_turn conflict same_direction For obj: at_left For obj: different_arms intersecting_traffic	

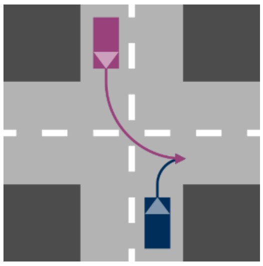
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
right_turn_with_obj_from_left_turning_left	Rechtsabbiegen mit linksabbiegendem Objekt von links	Right turn with object from left turning left	conflict_with_obj_from_left_arm conflict_with_obj_turning_left right_turn touching_following_conflict	For obj: stay_in_traffic_area maneuver same_direction For obj: left_turn touching For ego: right_turn conflict For obj: at_left For obj: different_arms intersecting_traffic	
right_turn_with_obj_from_left_turning_right	Rechtsabbiegen mit linksabbiegendem Objekt von rechts	Right turn with object from left turning right	conflict_with_obj_from_left_arm conflict_with_obj_turning_right merging_oncoming_conflict right_turn	For obj: stay_in_traffic_area maneuver For obj: right_turn opposite_direction merging For ego: right_turn conflict For obj: at_left For obj: different_arms intersecting_traffic	
right_turn_with_obj_from_right_crossing_after_node	Rechtsabbiegen mit kreuzendem Objekt von rechts nach dem Knoten	Right turn with object from right crossing after the node	conflict_after_node conflict_with_obj_crossing_ego_TA conflict_with_obj_from_right_with_TA_change intersecting_conflict right_turn	maneuver intersecting_traffic For obj: cross_ego_traffic_area traffic_area_change For ego: right_turn conflict For obj: intersecting For obj: at_right behind_node	
right_turn_with_obj_from_right_crossing_before_node	Rechtsabbiegen mit kreuzendem Objekt von rechts vor dem Knoten	Right turn with object from right crossing before the node	conflict_before_node conflict_with_obj_crossing_ego_TA conflict_with_obj_from_right_with_TA_change intersecting_conflict right_turn	maneuver For obj: cross_ego_traffic_area traffic_area_change before_node For ego: right_turn conflict For obj: intersecting For obj: at_right intersecting_traffic	
right_turn_with_obj_from_right_entering_after_node	Rechtsabbiegen mit eintretendem Objekt von rechts nach dem Knoten	Right turn with object from right entering after the node	conflict_after_node conflict_with_obj_entering_ego_TA conflict_with_obj_from_right_with_TA_change merging_following_conflict right_turn	For obj: enter_ego_traffic_area maneuver merging traffic_area_change For ego: right_turn conflict same_direction For obj: at_right intersecting_traffic behind_node	

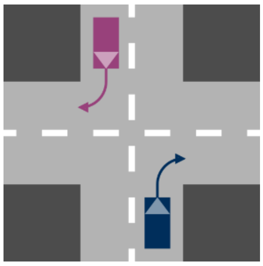
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
right_turn_with_obj_from_right_entering_before_node	Rechtsabbiegen mit eintretendem Objekt von rechts vor dem Knoten	Right turn with object from right entering before the node	conflict_before_node conflict_with_obj_entering_ego_TA conflict_with_obj_from_right_with_TA_change merging_following_conflict right_turn	For obj: enter_ego_traffic_area maneuver merging before_node traffic_area_change For ego: right_turn conflict same_direction For obj: at_right intersecting_traffic	
right_turn_with_obj_from_right_making_a_u-turn	Rechtsabbiegen mit wendendem Objekt von rechts	Right turn with object from right making a u-turn	conflict_with_obj_from_right_arm conflict_with_obj_making_a_u-turn merging_following_conflict right_turn	For obj: stay_in_traffic_area maneuver merging For obj: u-turn For ego: right_turn conflict same_direction For obj: at_right For obj: different_arms intersecting_traffic	
right_turn_with_obj_from_right_passing_straight	Rechtsabbiegen mit kreuzendem Objekt von rechts	Right turn with object from right passing straight	conflict_with_obj_from_right_arm conflict_with_obj_passing_straight merging_oncoming_conflict right_turn	For obj: stay_in_traffic_area maneuver For obj: passing_straight opposite_direction merging For ego: right_turn conflict For obj: at_right For obj: different_arms intersecting_traffic	
right_turn_with_obj_from_right_turning_left	Rechtsabbiegen mit linksabbiegendem Objekt von rechts	Right turn with object from right turning left	conflict_with_obj_from_right_arm conflict_with_obj_turning_left oncoming_conflict right_turn	For obj: stay_in_traffic_area maneuver For obj: left_turn For ego: right_turn conflict For obj: at_right For obj: different_arms intersecting_traffic For obj: oncoming	
right_turn_with_obj_from_right_turning_right	Rechtsabbiegen mit rechtsabbiegendem Objekt von rechts	Right turn with object from right turning right	conflict_with_obj_from_right_arm conflict_with_obj_turning_right merging_oncoming_conflict right_turn	For obj: stay_in_traffic_area maneuver For obj: right_turn opposite_direction merging For ego: right_turn conflict For obj: at_right For obj: different_arms intersecting_traffic	

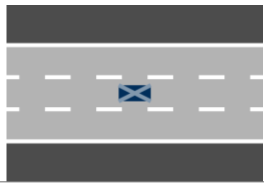
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image

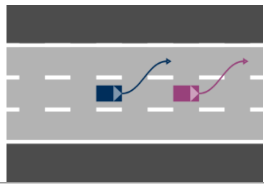
`right_turn_with_oncoming_obj_making_a_u-turn`	Rechtsabbiegen mit wendenden entgegenkommenden Objekt	Right turn with oncoming object making a u-turn	`conflict_with_obj_making_a_u-turn`
			`conflict_with_oncoming_obj`
			`right_turn`
			`touching_following_conflict`
				`For obj: stay_in_traffic_area`
				`maneuver`
				`same_direction`
				`For obj: u-turn`
				`touching`
				`For ego: right_turn`
				`conflict`
				`For obj: in_front_of`
				`For obj: different_arms`
				`intersecting_traffic`
					

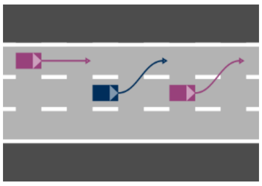
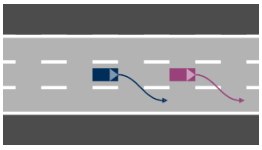
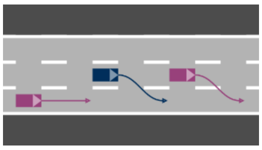
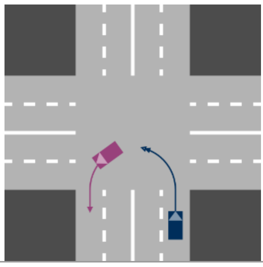
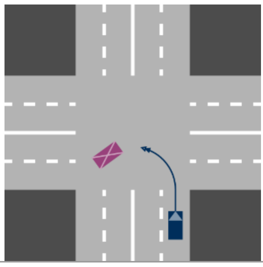
`right_turn_with_oncoming_obj_passing_straight`	Rechtsabbiegen mit kreuzendem entgegenkommenden Objekt	Right turn with oncoming object passing straight	`conflict_with_obj_passing_straight`
			`conflict_with_oncoming_obj`
			`oncoming_conflict`
			`right_turn`
				`For obj: stay_in_traffic_area`
				`maneuver`
				`For obj: passing_straight`
				`For ego: right_turn`
				`conflict`
				`For obj: in_front_of`
				`For obj: different_arms`
				`intersecting_traffic`
				`For obj: oncoming`
					

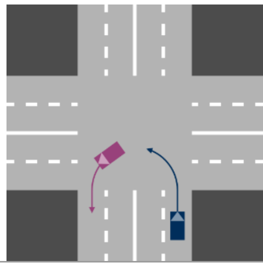
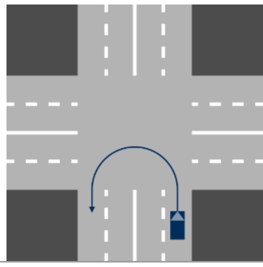
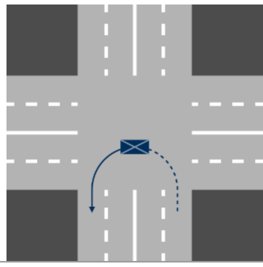
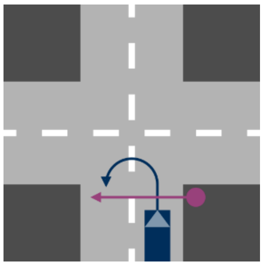
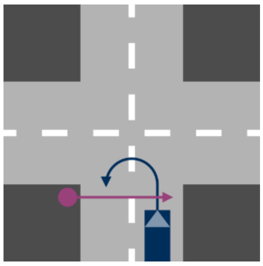
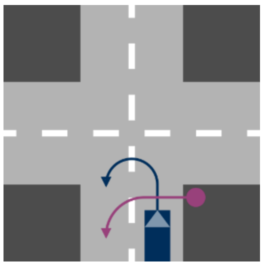
`right_turn_with_oncoming_obj_turning_left`	Rechtsabbiegen mit linksabbiegendem entgegenkommenden Objekt von links	Right turn with oncoming object turning left	`conflict_with_obj_turning_left`
			`conflict_with_oncoming_obj`
			`merging_following_conflict`
			`right_turn`
				`For obj: stay_in_traffic_area`
				`maneuver`
				`merging`
				`For obj: left_turn`
				`For ego: right_turn`
				`conflict`
				`same_direction`
				`For obj: in_front_of`
				`For obj: different_arms`
				`intersecting_traffic`
					

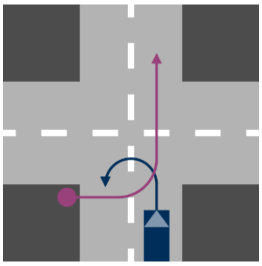
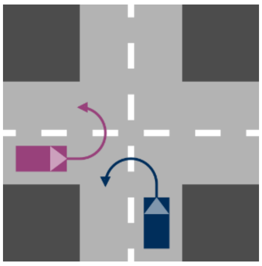
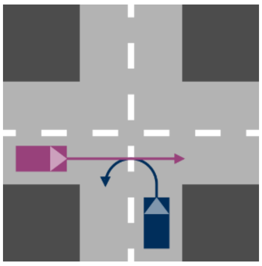
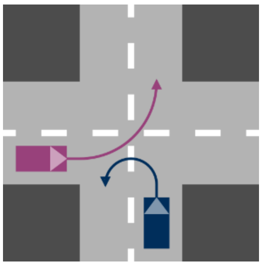
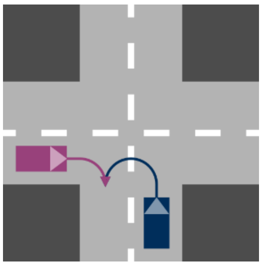
`right_turn_with_oncoming_obj_turning_right`	Rechtsabbiegen mit rechtsabbiegendem entgegenkommenden Objekt	Right turn with oncoming object turning right	`conflict_with_obj_turning_right`
			`conflict_with_oncoming_obj`
			`right_turn`
			`touching_oncoming_conflict`
				`For obj: stay_in_traffic_area`
				`maneuver`
				`For obj: right_turn`
				`touching`
				`For ego: right_turn`
				`conflict`
				`For obj: in_front_of`
				`For obj: different_arms`
				`intersecting_traffic`
				`opposite_direction`
					

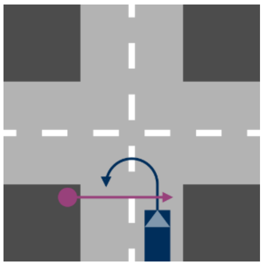
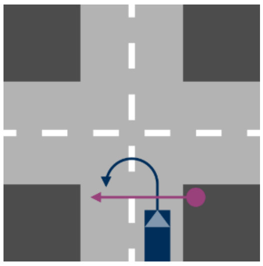
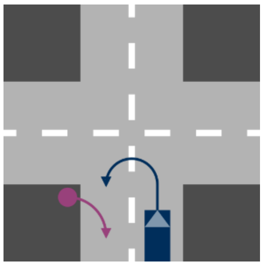
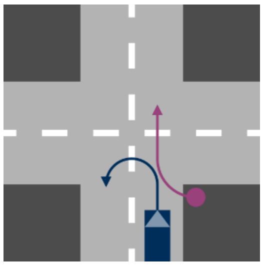
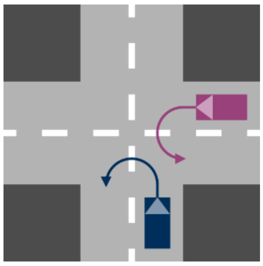
`standstill`	Stillstand	Standstill	`long_traffic_state`	`For ego: standstill`
				`longitudinal_traffic`
				`state`
					

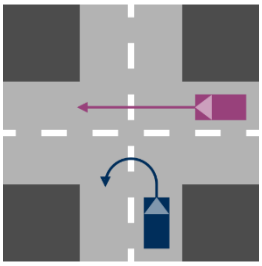
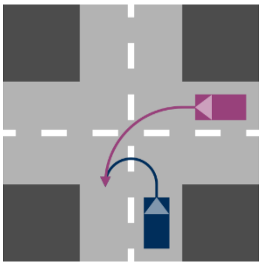
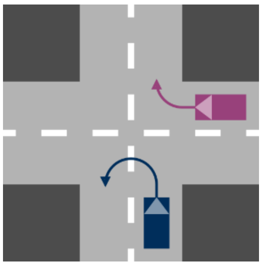
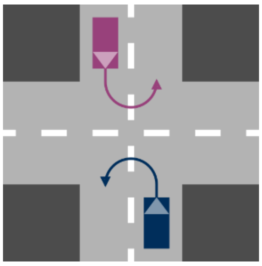
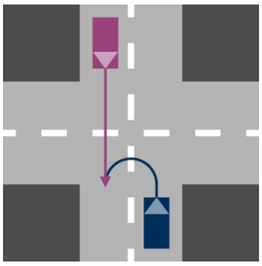
`sync_lcs_l_0`	Synchrone Fahrstreifenwechsel nach links ohne Hinterherfahrenden	Synchronous lane changes left without following object	`sync_lcs_0`
			`sync_lcs_l`
				`For obj: same_direction`
				`transition`
				`synchronous`
				`For ego: to_left`
				`For obj: entering`
				`For ego: lane_`
					

Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
				change longitudinal_traffic free For obj: stay_in_traffic_area	
sync_lcs_l_2	Synchrone Fahrstreifenwechsel nach links mit Hinterherfahrendem	Synchronous lane changes left with following object	sync_lcs_2 sync_lcs_l	For obj: same_direction transition synchronous For ego: to_left For obj: entering For ego: lane_change For obj: behind longitudinal_traffic For obj: stay_in_traffic_area	
sync_lcs_r_0	Synchrone Fahrstreifenwechsel nach rechts ohne Hinterherfahrenden	Synchronous lane changes right without following object	sync_lcs_0 sync_lcs_r	For obj: same_direction transition For ego: to_right synchronous For obj: entering For ego: lane_change longitudinal_traffic free For obj: stay_in_traffic_area	
sync_lcs_r_2	Synchrone Fahrstreifenwechsel nach rechts mit Hinterherfahrendem	Synchronous lane changes right with following object	sync_lcs_2 sync_lcs_r	For obj: same_direction transition For ego: to_right synchronous For obj: entering For ego: lane_change For obj: behind longitudinal_traffic For obj: stay_in_traffic_area	
u-turn_approaching_lead	Einem Vorausfahrenden beim Wenden annähern	U-turn approaching a leading object	approaching_lead_in_intersection u-turn	maneuver state For obj: leading For ego: approaching intersecting_traffic For ego: u-turn	
u-turn_approaching_static	Einem statischen Objekt beim Wenden annähern	U-turn approaching a static object	approaching_static_in_intersection u-turn	maneuver state For obj: static For ego: approaching intersecting_traffic For ego: u-turn	

Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
u-turn_following_lead	Einem Vorausfahrenden beim Wenden folgen	U-turn following leading object	following_in_intersection u-turn	maneuver intersecting_traffic state For ego: following For ego: u-turn	
u-turn_free	Freies Wenden	Free u-turn	free_intersection_maneuver u-turn	maneuver state For ego: free For ego: u-turn intersecting_traffic	
u-turn_standstill	Stillstand beim Wenden	Standstill while turning making a u-turn	standstill_in_intersection u-turn	maneuver state For ego: standstill intersecting_traffic For ego: u-turn	
u-turn_with_obj_from_left_crossing_after_node	Wenden mit nach dem Knotenpunkt kreuzendem Objekt	U-turn with object from left crossing after the node	conflict_after_node conflict_with_obj_crossing_ego_TA conflict_with_obj_from_left_with_TA_change intersecting_conflict u-turn	maneuver For obj: cross_ego_traffic_area traffic_area_change conflict For obj: intersecting For obj: at_left intersecting_traffic behind_node For ego: u-turn	
u-turn_with_obj_from_left_crossing_before_node	Wenden mit vor dem Knotenpunkt kreuzendem Objekt	U-turn with object from left crossing before the node	conflict_before_node conflict_with_obj_crossing_ego_TA conflict_with_obj_from_left_with_TA_change intersecting_conflict u-turn	maneuver For obj: cross_ego_traffic_area traffic_area_change before_node conflict For obj: intersecting For obj: at_left intersecting_traffic For ego: u-turn	
u-turn_with_obj_from_left_entering_after_node	Wenden mit nach dem Knotenpunkt eintretendem Objekt	U-turn with object from left entering after the node	conflict_after_node conflict_with_obj_entering_ego_TA conflict_with_obj_from_left_with_TA_change merging_following_conflict u-turn	For obj: enter_ego_traffic_area maneuver merging traffic_area_change conflict same_direction For obj: at_left intersecting_traffic behind_node For ego: u-turn	

Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
u-turn_with_ obj_from_left_ entering_ before_node	Wenden mit vor dem Knotenpunkt eintretendem Objekt	U-turn with object from left entering after the node	conflict_before_ node conflict_with_ obj_entering_ ego_TA conflict_with_ obj_from_left_ with_TA_change merging_ following_ conflict u-turn	For obj: enter_ ego_traffic_area maneuver merging traffic_area_ change before_node conflict same_direction For obj: at_left intersecting_ traffic For ego: u-turn	
u-turn_with_ obj_from_left_ making_a_u- turn	Wenden mit wendendem Objekt von links	U-turn with object from left making a u-turn	conflict_with_ obj_from_left_ arm conflict_with_ obj_making_a_u- turn touching_ oncoming_ conflict u-turn	For obj: stay_ in_traffic_area maneuver For obj: u-turn touching conflict For obj: at_left For obj: different_arms intersecting_ traffic For ego: u-turn opposite_ direction	
u-turn_with_ obj_from_left_ passing_ straight	Wenden mit kreuzendem Objekt von links	U-turn with object from left passing straight	conflict_with_ obj_from_left_ arm conflict_with_ obj_passing_ straight touching_ oncoming_ conflict u-turn	For obj: stay_ in_traffic_area maneuver For obj: passing_straight touching conflict For obj: at_left For obj: different_arms intersecting_ traffic For ego: u-turn opposite_ direction	
u-turn_with_ obj_from_left_ turning_left	Wenden mit linksabbiegendem Objekt von links	U-turn with object from left turning left	conflict_with_ obj_from_left_ arm conflict_with_ obj_turning_left touching_ oncoming_ conflict u-turn	For obj: stay_ in_traffic_area maneuver For obj: left_ turn touching conflict For obj: at_left For obj: different_arms intersecting_ traffic For ego: u-turn opposite_ direction	
u-turn_with_ obj_from_left_ turning_right	Wenden mit linksabbiegendem Objekt von rechts	U-turn with object from left turning right	conflict_with_ obj_from_left_ arm conflict_with_ obj_turning_ right merging_ following_ conflict u-turn	For obj: stay_ in_traffic_area maneuver For obj: right_ turn merging conflict same_direction For obj: at_left For obj: different_arms intersecting_ traffic For ego: u-turn	

Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
u-turn_with_obj_from_right_crossing_after_node	Wenden mit kreuzendem Objekt von rechts nach dem Knoten	U-turn with object from right crossing after the node	conflict_after_node conflict_with_obj_crossing_ego_TA conflict_with_obj_from_right_with_TA_change intersecting_conflict u-turn	maneuver For obj: cross_ego_traffic_area traffic_area_change conflict For obj: intersecting For obj: at_right intersecting_traffic behind_node For ego: u-turn	
u-turn_with_obj_from_right_crossing_before_node	Wenden mit kreuzendem Objekt von rechts vor dem Knoten	U-turn with object from right crossing before the node	conflict_before_node conflict_with_obj_crossing_ego_TA conflict_with_obj_from_right_with_TA_change intersecting_conflict u-turn	maneuver For obj: cross_ego_traffic_area traffic_area_change before_node conflict For obj: intersecting For obj: at_right intersecting_traffic For ego: u-turn	
u-turn_with_obj_from_right_entering_after_node	Wenden mit eintretendem Objekt von rechts nach dem Knoten	U-turn with object from right entering after the node	conflict_after_node conflict_with_obj_entering_ego_TA conflict_with_obj_from_right_with_TA_change merging_following_conflict u-turn	For obj: enter_ego_traffic_area maneuver merging traffic_area_change conflict same_direction For obj: at_right For ego: u-turn intersecting_traffic behind_node	
u-turn_with_obj_from_right_entering_before_node	Wenden mit eintretendem Objekt von rechts vor dem Knoten	U-turn with object from right entering before the node	conflict_before_node conflict_with_obj_entering_ego_TA conflict_with_obj_from_right_with_TA_change merging_following_conflict u-turn	For obj: enter_ego_traffic_area maneuver merging before_node traffic_area_change conflict same_direction For obj: at_right intersecting_traffic For ego: u-turn	
u-turn_with_obj_from_right_making_a_u-turn	Wenden mit wendendem Objekt von rechts	U-turn with object from right making a u-turn	conflict_with_obj_from_right_arm conflict_with_obj_making_a_u-turn touching_oncoming_conflict u-turn	For obj: stay_in_traffic_area maneuver For obj: u-turn touching conflict For obj: at_right For obj: different_arms For ego: u-turn intersecting_traffic opposite_direction	

Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
u-turn_with_obj_from_right_passing_straight	Wenden mit kreuzendem Objekt von rechts	U-turn with object from right passing straight	conflict_with_obj_from_right_arm conflict_with_obj_passing_straight touching_following_conflict u-turn	For obj: stay_in_traffic_area same_direction maneuver For obj: passing_straight touching conflict For obj: at_right For obj: different_arms intersecting_traffic For ego: u-turn	
u-turn_with_obj_from_right_turning_left	Wenden mit linksabbiegendem Objekt von rechts	U-turn with object from right turning left	conflict_with_obj_from_right_arm conflict_with_obj_turning_left merging_following_conflict u-turn	For obj: stay_in_traffic_area maneuver merging For obj: left_turn conflict same_direction For obj: at_right For obj: different_arms intersecting_traffic For ego: u-turn	
u-turn_with_obj_from_right_turning_right	Wenden mit rechtsabbiegendem Objekt von rechts	U-turn with object from right turning right	conflict_with_obj_from_right_arm conflict_with_obj_turning_right touching_following_conflict u-turn	For obj: stay_in_traffic_area same_direction maneuver For obj: right_turn touching conflict For obj: at_right For obj: different_arms intersecting_traffic For ego: u-turn	
u-turn_with_oncoming_obj_making_a_u-turn	Wenden mit wendenden entgegenkommenden Objekt	U-turn with oncoming object making a u-turn	conflict_with_obj_making_a_u-turn conflict_with_oncoming_obj touching_oncoming_conflict u-turn	For obj: stay_in_traffic_area maneuver For obj: u-turn touching conflict For obj: in_front_of For obj: different_arms intersecting_traffic For ego: u-turn opposite_direction	
u-turn_with_oncoming_obj_passing_straight	Wenden mit kreuzendem entgegenkommenden Objekt	U-turn with oncoming object passing straight	conflict_with_obj_passing_straight conflict_with_oncoming_obj merging_following_conflict u-turn	For obj: stay_in_traffic_area maneuver For obj: passing_straight merging conflict same_direction For obj: in_front_of For obj: different_arms intersecting_traffic For ego: u-turn	

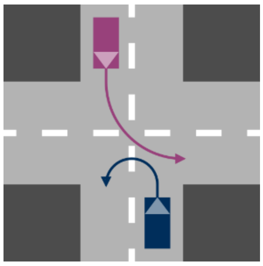
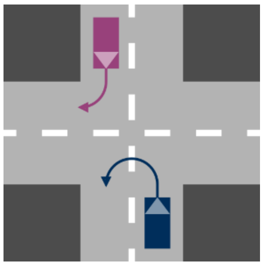
Identifier	Name (de)	Name (en)	Superclass	Concepts	Image
u-turn_with_ oncoming_obj_ turning_left	Wenden mit linksabbiegendem entgegenkommen den Objekt	U-turn with oncoming object turning left	conflict_with_ obj_turning_left conflict_with_ oncoming_obj touching_ oncoming_ conflict u-turn	For obj: stay_ in_traffic_area maneuver For obj: left_ turn touching conflict For obj: in_ front_of For obj: different_arms intersecting_ traffic For ego: u-turn opposite_ direction	
u-turn_with_ oncoming_obj_ turning_right	Wenden mit rechtsabbiegende m entgegenkommen den Objekt	U-turn with oncoming object turning right	conflict_with_ obj_turning_ right conflict_with_ oncoming_obj touching_ following_ conflict u-turn	For obj: stay_ in_traffic_area same_direction maneuver For obj: right_ turn touching conflict For obj: in_ front_of For obj: different_arms intersecting_ traffic For ego: u-turn	

Table 5: Overview of superclasses of base-scenarios

Identifier	Name (de)	Name (en)	Superclass	Concepts
aborted_enter_lead	Abgebrochener Eintritt eines Vorausfahrenden	Aborted lead entering	enter_within_ego_TA	For obj: aborted
enter_within_ego_TA	Eintritt innerhalb des Ego-Verkehrsraums	Entering within ego traffic area	enter_lead	For obj: stay_in_ traffic_area
aborted_lc	Abgebrochener Fahrstreifenwechsel	Aborted lane change	lane_change	For ego: aborted
lane_change	Fahrstreifenwechsel	Lane change	long_traffic_ transition	For ego: lane_ change
aborted_lc_0	Abgebrochener freier Fahrstreifenwechel	Aborted uninfluenced lane change	aborted_lc	For ego: free
aborted_lc_1	Abgebrochener Fahrstreifenwechsel mit Vorausfahrendem	Aborted lane change with lead object	aborted_lc	For obj: in_ front_of
aborted_lc_2	Abgebrochener Fahrstreifenwechsel mit Hinterherfahrendem	Aborted lane change with following object	aborted_lc	For obj: behind
aborted_lc_3	Abgebrochener Fahrstreifenwechsel mit Vorausfahrendem und Hinterherfahrendem	Aborted lane change with lead object and following object	aborted_lc	For obj: behind For obj: in_ front_of
aborted_lc_l	Abgebrochener Fahrstreifenwechel nach links	Aborted lane change left	aborted_lc	For ego: to_left
aborted_lc_r	Abgebrochener Fahrstreifenwechel nach rechts	Aborted lane change right	aborted_lc	For ego: to_right
approach	Annähern	Approach	long_traffic_state	For ego: approaching
long_traffic_state	Fahrzustand	State	longitudinal_traffic	state
approach_lat	Annähern an ein sich lateral bewegendes Objekt	Approach a laterally moving object	approach	For obj: lateral_ obj
approach_lat_ crossing_traffic_ area	Annähern an ein den Ego-Verkehrsraum kreuzendes Objekt	Approach a laterally moving object crossing the ego-traffic area	approach_lat	For obj: cross_ ego_traffic_area
approach_lat_from_ left	Annähern an ein Objekt von links	Approach a laterally moving object from left	approach_lat	For obj: from_ left
approach_lat_from_ right	Annähern an ein Objekt von rechts	Approach a laterally moving object from right	approach_lat	For obj: from_ right
approach_lat_ entering_traffic_ area	Annähern an ein den Ego-Verkehrsraum betretendes Objekt	Approach a laterally moving object entering the ego-traffic area	approach_lat	For obj: enter_ ego_traffic_area
approach_lat_ entering_traffic_ area_from_right	Annähern an ein den Ego-Verkehrsraum von rechts betretendes Objekt	Approach a laterally moving object entering the ego-traffic area from right	approach_lat_ entering_traffic_ area approach_lat_from_ right	
approach_lat_ leaving_traffic_area	Annähern an ein den Ego-Verkehrsraum verlassendes Objekt	Approahc a laterally moving object leaving the ego-traffic area	approach_lat	For obj: leaving_ ego_traffic_area
approach_overlapping	Annähern an ein seinen Fahrstreifen überlappendes Objekt	Approach an object overlapping its lane	approach	For obj: overlapping_ object
approach_ overlapping_lead	Annähern an einen seinen Fahrstreifen überlappenden Vorausfahrenden	Approach a leading object overlapping its lane	approach_overlapping	For obj: leading
approach_ overlapping_left	Annähern an ein seinen Fahrstreifen überlappendes Objekt links	Approach an object overlapping its lane right	approach_overlapping	For obj: at_left

Identifier	Name (de)	Name (en)	Superclass	Concepts
approach_overlapping_right	Annähern an ein seinen Fahrstreifen überlappendes rechts	Approach an object overlapping its lane left	approach_overlapping	For obj: at_right
approaching_in_intersection	Annähern in einer Kreuzung	Approach Object in an intersection	intersection_state	For ego: approaching
intersection_state	Fahrzustand innerhalb einer Kreuzung	Driving state within intersection	intersection_scenario	state
approaching_lead_in_intersection	Annähern an einen Vorausfahrenden in der Kreuzung	Approach a leading object in an intersection	approaching_in_intersection	For obj: leading
approaching_static_in_intersection	Annähern an ein statisches Objekt in der Kreuzung	Approach a static object in an intersection	approaching_in_intersection	For obj: static
follow	Folgen	Follow object	long_traffic_state	For ego: following
base_scenario	Grundszenario	Base scenario	Thing	
long_traffic_overlay	überlagerte Interaktion	Superimposed interaction	longitudinal_traffic	overlay
close_obj_side	Laterales Nahdistanz-Ereignis	Lateral close distance event	long_traffic_overlay	For ego: close_obj
close_obj_side_in_intersection	Laterales Nahdistanz-Ereignis in Kreuzung	Lateral close distance event in intersection	parallel_intersection_overlay	For ego: close_obj
parallel_intersection_overlay	Überlagerte Interaktion parallel in Kreuzung	Parallel superimposed interaction in intersection	intersection_overlay	same_arm
conflict_after_node	Konflikt vor dem Knotenpunkt	Conflict before node	conflict_with_TA_change	behind_node
conflict_with_TA_change	Konflikt mit Verkehrsraumwechsel	Conflict with traffic area change	intersection_conflict	traffic_area_change
conflict_before_node	Konflikt nach dem Knotenpunkt	Conflict after node	conflict_with_TA_change	before_node
intersection_conflict	Kreuzungskonflikt	Intersection conflict	intersection_scenario	conflict
conflict_with_obj_crossing_ego_TA	Konflikt mit Objekt, das den Ego-Verkehrsraum kreuzt	Conflict with object crossing ego-traffic area	conflict_with_TA_change	For obj: cross_ego_traffic_area
conflict_with_obj_entering_ego_TA	Konflikt mit Objekt, das in den Ego-Verkehrsraum eintritt	Conflict with object entering ego-traffic area	conflict_with_TA_change	For obj: enter_ego_traffic_area
conflict_with_obj_from_left	Konflikt mit Objekt von links	Conflict with object from left	intersection_conflict	For obj: at_left
conflict_with_obj_from_left_arm	Konflikt mit Objekt von linken Arm	Conflict with object from left arm	conflict_with_obj_from_left conflict_with_obj_on_different_arm	
conflict_with_obj_on_different_arm	Konflikt mit Objekt auf anderem Arm	Conflict with object on different arm	conflict_within_TA	For obj: different_arms
conflict_with_obj_from_left_with_TA_change	Konflikt mit Objekt von links mit Verkehrsraumwechsel	Conflict with object from left arm with traffic area change	conflict_with_TA_change conflict_with_obj_from_left	
conflict_with_obj_from_right	Konflikt mit Objekt von rechts	Conflict with object from right	intersection_conflict	For obj: at_right
conflict_with_obj_from_right_arm	Konflikt mit Objekt vom rechten Arm	Conflict with object from right arm	conflict_with_obj_from_right conflict_with_obj_on_different_arm	
conflict_with_obj_from_right_with_TA_change	Konflikt mit Objekt von rechts mit Verkehrsraumwechsel	Conflict with object from right with traffic area change	conflict_with_TA_change conflict_with_obj_from_right	
conflict_with_obj_making_a_u-turn	Konflikt mit Wendendem	Conflict with object making u-turn	conflict_within_TA	For obj: u-turn

Identifier	Name (de)	Name (en)	Superclass	Concepts
conflict_within_TA	Konflikt mit Verkehrsraumwechsel	Conflict with traffic area change	intersection_ conflict	For obj: stay_in_ traffic_area
conflict_with_obj_ passing_straight	Konflikt mit gerade Passierendem	Conflict with object passing straight	conflict_within_TA	For obj: passing_ straight
conflict_with_obj_ turning_left	Konflikt mit Linksabbiegendem	Conflict with object turning left	conflict_within_TA	For obj: left_ turn
conflict_with_obj_ turning_right	Konflikt mit Rechtsabbiegenden	Conflict with object turning right	conflict_within_TA	For obj: right_ turn
conflict_with_ oncoming_obj	Konflikt mit Entgegenkommenden	Conflict with oncoming object	conflict_with_obj_ on_different_arm	For obj: in_ front_of
conflict_with_ parallel_entry	Konflikt mit parallelem Eintritt	Conflict with parallel entry	conflict_within_TA	same_arm
conflict_with_ parallel_entry_from_ left	Konflikt auf gleichem Arm von links	Conflict on same arm from left	conflict_with_obj_ from_left conflict_with_ parallel_entry	
conflict_with_ parallel_entry_from_ right	Konflikt auf gleichem Arm von rechts	Conflict on same arm from right	conflict_with_obj_ from_right conflict_with_ parallel_entry	
cut_through	Durchscherer	Cut through	enter_within_ego_TA exit_within_ego_TA	
exit_within_ego_TA	Austritt innerhalb des Ego-Verkehrsraum	Exiting within ego traffic area	exit_lead	For obj: stay_in_ traffic_area
diverging_lead	Divergierender Vorausfahrender	Diverging leading object	intersection_ transition	For obj: diverging
intersection_ transition	Transition innerhalb einer Kreuzung	Transition in intersection	intersection_ scenario	transition
diverging_lead_ leaving_TA_after_ node	Divergierender Vorausfahrender, der den Ego-Verkehrsraum nach dem Knoten verlässt	Diverging leading object leaving ego-traffic area after the node	diverging_lead_ leaving_ego_TA	behind_node
diverging_lead_ leaving_ego_TA	Divergierender Vorausfahrender, der den Ego-Verkehrsraum verlässt	Diverging leading object leaving ego-traffic area	diverging_lead	For obj: leaving_ ego_traffic_area
diverging_lead_left	Divergierender Vorausfahrender nach links	Diverging leading object to left	diverging_lead	For obj: to_left
diverging_lead_right	Divergierender Vorausfahrender nach rechts	Diverging leading object to right	diverging_lead	For obj: to_right
diverging_lead_ leaving_TA_before_ node	Divergierender Vorausfahrender, der den Ego-Verkehrsraum vor dem Knoten verlässt	Diverging leading object leaving ego-traffic area before the node	diverging_lead_ leaving_ego_TA	before_node
diverging_lead_ within_ego_TA	Divergierender Vorausfahrender innerhalb des Ego-Verkehrsraums	Diverging leading object withing ego traffic area	diverging_lead	For obj: stay_in_ traffic_area
enter_forward	Eintritt eines Vorausfahrenden	A leading object entering	enter_from_non_ego_ TA	For obj: forward
enter_from_non_ego_ TA	Eintritt in den Ego-Verkehrsraum	Object entering ego-traffic area	enter_lead	For obj: enter_ ego_traffic_area
enter_forward_left	Eintritt eines Vorausfahrenden von links	A leading object entering from left	enter_forward enter_l	
enter_l	Eintritt von links	Object entering from left	enter_obj	For obj: from_ left
enter_forward_right	Eintritt eines Vorausfahrenden von rechts	A leading object entering from right	enter_forward enter_r	

Identifier	Name (de)	Name (en)	Superclass	Concepts
enter_r	Eintritt eines Objekts von rechts	Object entering from right	enter_obj	For obj: from_ right
enter_lead	Eintritt eines Vorausfahrenden	A leading object entering	enter_obj transition_with_lead	
enter_obj	Eintritt	Object entering	long_traffic_ transition	For obj: entering
transition_with_lead	Transition mit Vorausfahrendem	Transition with leading object	long_traffic_ transition	For obj: in_ front_of For obj: same_ direction
long_traffic_ transition	Transition	Transition	longitudinal_traffic	transition
enter_oncoming	Eintritt eines Entgegenkommenden	Oncoming object entering	enter_obj transition_with_ oncoming	
transition_with_ oncoming	Transition mit Entgegenkommenden	Transition with oncoming object	long_traffic_ transition	For obj: opposite_ direction
enter_parallel	Paralleler Eintritt eines Objekts	Object entering parallel	enter_from_non_ego_ TA	For obj: parallel
enter_parallel_ forward	Paralleler Eintritt eines Objekts vorwärts	Object entering parallel forward	enter_forward enter_parallel	
enter_parallel_left	Paralleler Eintritt eines Objekts von links	Object entering parallel from left	enter_l enter_parallel	
enter_parallel_right	Paralleler Eintritt eines Objekts von rechts	Object entering parallel from right	enter_parallel enter_r	
enter_parallel_ reversing	Paralleler Eintritt eines Objekts rückwärts	Object entering parallel reversing	enter_parallel enter_reversing	
enter_reversing	Eintritt eines Objekts rückwärts	Object entering reversing	enter_from_non_ego_ TA	For obj: reversing
enter_reversing_left	Eintritt eines Objekts rückwärts von links	Object entering reversing from left	enter_l enter_reversing	
enter_reversing_ right	Eintritt eines Objekts rückwärts von rechts	Object entering reversing from right	enter_r enter_reversing	
enter_turning	Eintritt eines Objekts einbiegend	Object entering turning	enter_from_non_ego_ TA	For obj: turning
enter_turning_ forward	Eintritt eines Objekts vorwärts einbiegend	Object entering forward turning	enter_forward enter_turning	
enter_turning_left	Eintritt eines Objekts einbiegend von links	Object entering turning from left	enter_l enter_turning	
enter_turning_right	Eintritt eines Objekts einbiegend von rechts	Object entering turning from right	enter_r enter_turning	
enter_turning_ reversing	Eintritt eines Objekts rückwärts einbiegend	Object entering turning reversing	enter_reversing enter_turning	
exit_forward	Austritt eines Vorausfahrenden	Object exiting forward	exit_leaving_ego_TA	For obj: forward
exit_leaving_ego_TA	Austritt aus dem Ego-Verkehrsraum	Object exiting the ego-traffic area	exit_lead	For obj: leaving_ ego_traffic_area
exit_forward_left	Austritt eines Vorausfahrenden nach links	Object exiting forward to the left	exit_forward exit_l	
exit_l	Austritt nach links	Object exiting to the left	exit_obj	For obj: to_left
exit_forward_right	Austritt eines Vorausfahrenden nach rechts	Object exiting forward to the right	exit_forward exit_r	
exit_r	Austritt nach rechts	Object exiting to the right	exit_obj	For obj: to_right
exit_obj	Austritt	Object exiting	long_traffic_ transition	For obj: exiting
exit_lead	Austritt eines Vorausfahrenden	A leading object exiting	exit_obj transition_with_lead	

Identifier	Name (de)	Name (en)	Superclass	Concepts
exit_oncoming	Austritt eines Entgegenkommenden	Oncoming object exiting	exit_obj transition_with_ oncoming	
exit_parallel	Paralleler Austritt aus dem Ego-Verkehrsraum	Object exiting parallel leaving ego-traffic area	exit_leaving_ego_TA	For obj: parallel
exit_parallel_ forward	Paralleler Austritt aus dem Ego Verkehrsraum vorwärts	Object exiting parallel forward leaving ego-traffic area	exit_forward exit_parallel	
exit_parallel_left	Paralleler Austritt aus dem Ego-Verkehrsraum nach links	Object exiting parallel to the left	exit_l exit_parallel	
exit_parallel_right	Paralleler Austritt aus dem Ego-Verkehrsraum	Object exiting parallel to the right	exit_parallel exit_r	
exit_parallel_ reversing	Paralleler Austritt aus dem Ego-Verkehrsraum rückwärts	Object exiting parallel reversing	exit_parallel exit_reversing	
exit_reversing	Austritt rückwärts	Object exiting reversing	exit_leaving_ego_TA	For obj: reversing
exit_reversing_left	Austritt rückwärts nach links	Object exiting reversing to the left	exit_l exit_reversing	
exit_reversing_right	Austritt rückwärts nach rechts	Object exiting reversing to the right	exit_r exit_reversing	
exit_turning	Austritt abbiegend	Object exiting turning	exit_leaving_ego_TA	For obj: turning
exit_turning_forward	Austritt abbiegend vorwärts	Object exiting forward turning	exit_forward exit_turning	
exit_turning_left	Austritt abbiegend nach links	Object exiting turning to the left	exit_l exit_turning	
exit_turning_right	Austritt abbiegend nach rechts	Object exiting turning to the right	exit_r exit_turning	
exit_turning_ reversing	Austritt abbiegend rückwärts nach	Object exiting turning reversing	exit_reversing exit_turning	
follow_overlapping	Einem seinen Fahrstreifen überlappendem Objekt folgen	Follow object overlapping its lane	follow	For obj: overlapping_ object
following_in_ intersection	Einem Vorausfahrenden innerhalb einer Kreuzung folgen	Follow object in intersection	intersection_state	For ego: following
free_intersection_ maneuver	Freies Kreuzungsmanöver	Free maneuver a intersection	intersection_state	For ego: free
incomplete_enter_ lead	Unvollständiger Eintritt eines Vorausfahrenden	Incomplete lead entering	enter_within_ego_TA	incomplete
intersecting_ conflict	Kreuzender Konflikt	Intersecting conflict	intersection_ conflict	For obj: intersecting
intersection_ scenario	Szenario auf Kreuzung	Intersection scenario	base_scenario	intersecting_ traffic
intersection_enter_ lead	Eintritt eines Vorausfahrenden innerhalb einer Kreuzung	Lead object entering in intersection	intersection_ transition	For obj: entering
intersection_exit_ lead	Austritt eines Vorausfahrenden innerhalb einer Kreuzung	Lead object exiting in intersection	intersection_ transition	For obj: exiting
intersection_lane_ change	Fahrstreifenwechsel innerhalb einer Kreuzung	Lane change in intersection	intersection_ transition	For obj: lane_ change
intersection_ maneuver	Kreuzungsmanöver	Maneuver at intersection	intersection_ scenario	maneuver
intersection_overlay	Überlagerte Interaktion innerhalb einer Kreuzung	Superimposed interaction in intersection	intersection_ scenario	overlay
lc_0	Freier Fahrstreifenwechsel	Uninfluenced lane change	lane_change	free
lc_1	Fahrstreifenwechsel mit Vorausfahrendem	Lane change with lead object	lane_change	For obj: in_ front_of

Identifier	Name (de)	Name (en)	Superclass	Concepts
lc_2	Fahrstreifenwechsel mit Hinterherfahrendem	Lane change with following object	lane_change	For obj: behind
lc_3	Fahrstreifenwechsel mit Vorausfahrendem und Hinterherfahrendem	Lane change with lead object and following object	lane_change	For obj: behind For obj: in_ front_of
lc_from_oncoming	Fahrstreifenwechsel aus Gegenverkehr	Lane change from oncoming traffic	lane_change transition_with_ oncoming	from_oncoming
lc_left	Fahrstreifenwechsel nach links	Lane change left	lane_change	For ego: to_left
lc_right	Fahrstreifenwechsel nach rechts	Lane change right	lane_change	For ego: to_right
lc_to_oncoming	Fahrstreifenwechsel in Gegenverkehr	Lane change to oncoming traffic	lane_change transition_with_ oncoming	to_oncoming
left_turn	Linksabbiegen	Left turn	intersection_ maneuver	For ego: left_ turn
standstill_in_ intersection	Stillstand auf der Kreuzung	Standstill in intersection	intersection_state	For ego: standstill
merging_following_ conflict	Zusammenführender folgender Konflikt	Merging following conflict	merging_conflict	same_direction
left_turn_with_obj_ from_left_passing_ straight	Linksabbiegen mit kreuzendem Objekt von links	Left turn with object from left passing straight	conflict_with_obj_ from_left_arm conflict_with_obj_ passing_straight left_turn	
merging_oncoming_ conflict	Zusammenführender entgegenkommender Konflikt	Merging oncoming conflict	merging_conflict	opposite_ direction
oncoming_conflict	Entgegenkommender Konflikt	Oncoming conflict	intersection_ conflict	For obj: oncoming
touching_oncoming_ conflict	Berührender entgegenkommender Konflikt	Touching oncoming conflict	touching_conflict	opposite_ direction
touching_following_ conflict	Berührender gleichgerichteter Konflikt	Touching following conflict	touching_conflict	same_direction
left_turn_with_ oncoming_obj_ passing_straight	Linksabbiegen mit kreuzendem entgegenkommenden Objekt	Left turn with oncoming object passing straight	conflict_with_obj_ passing_straight conflict_with_ oncoming_obj left_turn	
longitudinal_traffic	Längsverkehr	Longitudinal traffic	base_scenario	longitunidal_ traffic
merging_conflict	Zusammenführendender Konflikt	Merging conflict	intersection_ conflict	merging
merging_cut_through	Zusammenführender Durchscherer	Merging cut through	cut_through merging_lcs	
merging_lcs	Zusammenfährende Fahrstreifenwechsel	Merging lane changes	enter_within_ego_TA lane_change	merging
merging_cut_through_ 0	Zusammenführender Durchscherer ohne Hinterherfahrenden	Merging cut through without following vehicle	lc_0 merging_cut_through	
merging_cut_through_ 2	Zusammenführender Durchscherer mit Hinterherfahrendem	Merging cut through with following vehicle	lc_2 merging_cut_through	
merging_cut_through_ l	Zusammenführender Durchscherer nach links	Merging cut through left	lc_left merging_cut_through	
merging_cut_through_ r	Zusammenführender Durchscherer nach rechts	Merging cut through right	lc_right merging_cut_through	
merging_lcs_0	Zusammenführende Fahrstreifenwechsel ohne Hinterherfahrenden	Merging lane changes without following object	merging_lcs	free

Identifier	Name (de)	Name (en)	Superclass	Concepts
merging_lcs_2	Zusammenführende Fahrstreifenwechsel mit Hinterherfahrenden	Merging lane changes with following object	merging_lcs	For obj: behind
merging_lcs_l	Zusammenführende Fahrstreifenwechsel nach links	Merging lane changes left	merging_lcs	For ego: to_left
merging_lcs_r	Zusammenführende Fahrstreifenwechsel nach rechts	Merging lane changes right	merging_lcs	For ego: to_right
multi_lcs	Mehrfacher Fahrstreifenwechsel	Multiple lane changes	lane_change	multi_lc
neighbor	Laterales Verweilen eines anderen Fahrzeugs	Object staying to side of ego	long_traffic_overlay	For ego: having_ neighbor
neighbor_entering	Eintretendes Fahrzeug auf den Nachbarfahrstreifen	Object entering to the side of ego	long_traffic_overlay	For obj: entering For obj: having_ neighbor
neighbor_exiting	Vom Nachbarfahrstreifen austretendes Fahrzeug	Object exiting to the side of ego	long_traffic_overlay	For obj: exiting For obj: having_ neighbor
neighbor_in_ intersection	Laterales Verweilen eines anderen Fahrzeugs in der Kreuzung	Object staying to side of ego in intersection	parallel_ intersection_overlay	For ego: having_ neighbor
non-parallel_ intersection_overlay	Vorbeifahrt an einem Objekt in der Kreuzung	Pass object in the intersection	intersection_overlay	different_arms
overlap_lane	Den Fahrstreifen überlappen	Overlap lane	long_traffic_ transition	For obj: overlap_ lane
overlap_lane_left	Den Fahrstreifen nach links überlappen	Overlap lane to the left	overlap_lane	For ego: at_left
overlap_lane_with_ leading	Den Fahrstreifen überlappen mit einem Vorausfahrenden	Overlap lane with leading object	overlap_lane	For obj: leading
overlap_lane_right	Den Fahrstreifen nach rechts überlappen	Overlap lane to the right	overlap_lane	For ego: at_right
overlap_lane_with_ oncoming	Den Fahrstreifen überlappen mit einem Entgegenkommenden	Overlap lane with oncoming object	overlap_lane	For obj: oncoming
overlap_lane_free	Den Fahrstreifen frei überlappen	Overlap lane free	overlap_lane	For ego: free
pass_straight	Kreuzen	Pass straight through intersection	intersection_ maneuver	For ego: passing_ straight
right_turn	Rechtsabbiegen	Right turn	intersection_ maneuver	For ego: right_ turn
pass_obj_in_ intersection_left	Vorbeifahrt an einem Objekt in der Kreuzung links	Pass object on the left in the intersection	non-parallel_ intersection_overlay	For obj: at_left
pass_obj_in_ intersection_right	Vorbeifahrt an einem Objekt in der Kreuzung rechts	Pass object on the left in the intersection	non-parallel_ intersection_overlay	For obj: at_right
pass_obj_in_ intersection_moving_ away	Vorbeifahrt an einem sich entfernenden Objekt in der Kreuzung	Pass object in the inersection moving away from ego	non-parallel_ intersection_overlay	For obj: away
pass_obj_in_ intersection_moving_ toward	Vorbeifahrt an einem sich nähernden Objekt in der Kreuzung	Pass object in the intersection moving towards ego	non-parallel_ intersection_overlay	For obj: towards
pass_obj_in_ intersection_ parallel	Parallele Vorbeifahrt an einem Objekt in der Kreuzung	Pass object in intersection parallel	parallel_ intersection_overlay	For ego: pass
pass_obj_making_u- turn	Vorbeifahrt an einem wendenden Objekt	Pass object making a u- turn	non-parallel_ intersection_overlay	For obj: u-turn
pass_obj_passing_ straight_in_ intersection	Vorbeifahrt an einem kreuzenden Objekt	Pass object passing straight	non-parallel_ intersection_overlay	For obj: passing_ straight

Identifier	Name (de)	Name (en)	Superclass	Concepts
pass_obj_turning_left_in_intersection	Vorbeifahrt an einem linksabbiegenden Objekt	Pass object turning left in intersection	non-parallel_intersection_overlay	For obj: left_turn
pass_obj_turning_right_in_intersection	Vorbeifahrt an einem rechtsabbiegenden Objekt	Pass object turning right in intersection	non-parallel_intersection_overlay	For obj: right_turn
pass_straight_with_obj_from_right_turning_left	Kreuzen mit linksabbiegendem Objekt von rechts	Pass straight with object from right turning left	conflict_with_obj_from_right_arm conflict_with_obj_turning_left pass_straight	
passed	Vorbeifahrt eines Objekts	Another object passing	long_traffic_overlay	For ego: being_passed
passed_in_itersection	Vorbeifahrt eines Objekts in der Kreuzung rechts	An object passing in intersection	parallel_intersection_overlay	For ego: being_passed
passed_in_lane	Vorbeifahrt eines Objekts im Fahrstreifen	Object passing in lane	long_traffic_overlay	For ego: being_passed_in_lane
passing	Vorbeifahrt an einem anderen Fahrzeug	Passing an object	long_traffic_overlay	For ego: pass
rear_obj_entering	Eintritt eines Hinterherfahrenden	Rear object entering	long_traffic_overlay	For obj: behind For obj: entering
rear_obj_exiting	Austritt eines Hinterherfahrenden	Rear object exiting	long_traffic_overlay	For obj: behind For obj: exiting
reverse	Rückwärtsfahrt	Reversing	long_traffic_state	For ego: reversing
reverse_approach	Rückwärtsfahrt, sich einem Objekt annähernd	Reverse, approaching an object	reverse	For ego: approaching
reverse_approach_lat	Rückwärtsfahrt, sich einem lateral bewegenden Objekt annähernd	Reverse, approaching a laterally moving object	reverse_approach	For obj: lateral_obj
sync_lcs	Synchrone Fahrstreifenwechsel	Synchronous lane changes	enter_within_ego_TA lane_change	synchronous
sync_lcs_0	Synchrone Fahrstreifenwechsel ohne Hinterherfahrenden	Synchronous lane changes without following object	sync_lcs	free
sync_lcs_2	Synchrone Fahrstreifenwechsel mit Hinterherfahrendem	Synchronous lane changes with following object	sync_lcs	For obj: behind
sync_lcs_l	Synchrone Fahrstreifenwechsel nach links	Synchronous lane changes left	sync_lcs	For ego: to_left
sync_lcs_r	Synchrone Fahrstreifenwechsel nach rechts	Synchronous lane changes right	sync_lcs	For ego: to_right
touching_conflict	Berührender Konflikt	Touching conflict	intersection_conflict	touching
u-turn	Wenden	U-turn	intersection_maneuver	For ego: u-turn