Proc. of the European Safety and Reliability Conference (ESREL 2026)
Edited by José C. Matos, Paulo B. Lourenço, Michael Beer, Daniel Oliveira,
Edoardo Patelli, Hélder S. Sousa, Rui A. Silva and Monica Santamaria

doi: 10.3850/ESREL2026061419_esrel26-p26439-cd

From ODDs to Simulation: Automatic Railway Scenario Generation

Leticia Fernandez Moguel; Tommy Lee Kälin; Monika Reif
Safety-Critical Systems Research Lab, Zurich University of Applied Sciences ZHAW, Switzerland.
E-mail: {leticia.fernandezmoguel; tommylee.kaelin; monika.reif} @zhaw.ch

Jiwon Shin
Stadler Signalling, Switzerland. E-mail: jiwon.shin@stadlerrail.com

The safe deployment of autonomous railway systems requires systematic testing across a wide range of representative and safety-critical scenarios. Defining these scenarios in a structured and reproducible manner is a key challenge, as they must accurately reflect the system's intended Operational Design Domain. For the railway domain, however, systematic approaches for defining and generating such scenarios remain limited. This paper presents a methodology for automatic scenario generation that bridges the gap between formal Operational Design Domain definitions and simulation-based testing of autonomous railway systems.
We propose a structured framework to derive simulation scenarios from Operational Design Domain specifications by identifying relevant operational parameters, environmental factors, and interaction types. In this context, an Operational Design Domain for railways is defined as the set of environmental, infrastructural, and operational conditions under which an autonomous train system is designed to operate safely. The approach formalizes Operational Design Domain components, such as infrastructure, operational constraints, and external agents, and defines how they can be parameterized and combined to form meaningful scenario classes. This structured representation ensures that the generated scenarios are traceable to system-level requirements and representative of real operational contexts.
Building upon this methodological foundation, we demonstrate its implementation within the CARLA simulation environment. Although originally designed for road-based traffic, CARLA's open and modular architecture enables its extension to railway applications. We introduce adaptations that support track-based motion, signaling systems, station layouts, and multi-modal interactions, as well as the automated generation of OpenSCENARIO templates aligned with the defined Operational Design Domain structure.
The resulting framework enables scalable and reproducible scenario generation directly linked to Operational Design Domain specifications, supporting systematic validation and verification processes for autonomous railway technologies. By connecting conceptual Operational Design Domain definitions with executable simulations, the approach facilitates structured scenario-based testing and provides a foundation for simulation-driven safety assessment and reliability evaluation in the development of future autonomous rail systems.

Keywords: Operational Design Domain (ODD), Scenario generation, OpenSCENARIO, CARLA.

1. Introduction

In recent years, autonomous driving technologies have rapidly emerged as one of the most transformative innovations in modern transportation. These systems rely on advanced perception, decision-making, and control algorithms powered by artificial intelligence and sensor perception. The safe deployment of autonomous systems requires systematic testing across a wide range of representative and safety-critical scenarios. The definition of such scenarios in a structured and reproducible manner is a key challenge, as they must accurately reflect the system's intended Operational Design Domain (ODD).

In the autonomous driving industry, there have been several initiatives to define ODDs. The SAE J3016 standard (SAE (2021)), ANSI/UL 4600 standard (ANSI (2023)), ISO 34503:2023 (ISO (2023)) standard, BSI-PAS1883:2020 (BSI (2020)), and the ASAM OpenODD (ASAM (2025)) have proposed definitions of the ODD. In comparison with the research on autonomous street vehicles, research in the railway industry on autonomous systems is limited. Recently, a new standard for ODD specific for railways was published DIN DKE SPEC 99004: 2025-05

(DIN-DKE (2025)). Furthermore, in (Zeller et al. (2024)) relevant challenges to automate the different stages of the safe MLOps process, particularly within the railway industry are described. One of the identified challenges during the data preparation is how to automatically analyze the ODD in terms of data distribution and diversity.

Despite the availability of formal ODD definitions (DIN-DKE (2025)), a systematic and automated connection between ODD specifications and executable test scenarios is still largely missing in the railway domain. In particular, there is a lack of methodologies that support the derivation of simulation scenarios directly from ODD elements in a way that is scalable, traceable, and suitable for safety-oriented validation activities.

In this paper, we propose a methodology to tackle the challenge of generating scenarios from the ODD definition. In Section 2.1, we propose further adaptations to existing ODD standards aiming to facilitate connection of the ODD elements with scenario generation. In Section 2.2 we present a methodology for automatic scenario generation that bridges the gap between formal ODD definitions and simulation-based testing of autonomous railway systems. In Section 4, we conclude the paper by describing the main encountered limitations and outlining future work.

2. Methods and tools

2.1. *Operational Design Domain (ODD)*

The concept of the Operational Design Domain (ODD) originates from the automotive industry and serves as a framework for defining the conditions under which automated systems can operate safely and effectively.

In the autonomous driving industry, there have been several initiatives on how to define an ODD. According to the SAE J3016 standard (SAE (2021)), an ODD specifies the operating conditions of a system, including environmental, geographical, temporal, and infrastructural restrictions. The definition is echoed in the ANSI/UL 4600 standard (ANSI (2023)), which describes an ODD as a set of environments and situations within which an automated system is intended to function. Similarly, the United Nations Economic Commission for Europe (UNECE) defines the ODD in terms of environmental, geographic, time-of-day, traffic, infrastructure, weather, and other conditions that determine the operational boundaries of an automated system. The ISO 34503:2023 standard (ISO (2023)) further refines this definition by distinguishing between the ODD and the Target Operational Domain (TOD), where the TOD represents the broader set of real-world conditions a system may encounter. At the same time, the ODD delineates the specific operational parameters under which the system can function safely. Additional definitions of ODDs in the automobile sector can be found in BSI-PAS1883:2020 (BSI (2020)), and the ASAM OpenODD ASAM (2025).

Railway operations take place in highly complex environments where interactions with road vehicles, pedestrians, and cyclists are frequent in certain operational contexts. Consequently, the ODD for train systems must include both predictable and unpredictable factors across diverse mobility settings. Railway infrastructure includes fixed and isolated tracks as well as segments that intersect shared roadways, crossings, and pedestrian areas, requiring precise specification of route characteristics, station configurations, and signaling systems. Environmental conditions such as rain, snow, fog, wind, and extreme temperatures must also be integrated into the ODD, as they influence braking performance, track adhesion, and the reliability of sensor-based systems. The variability of urban weather patterns, which can obscure sensors, reduce visibility, or degrade system performance, imposes operational constraints that need to be explicitly captured within the ODD. Moreover, the dynamic nature of railway environments necessitates consideration of passenger-related factors, including fluctuations in crowd density, the presence of individuals with mobility aids or bicycles, and unpredictable boarding and alighting behaviors. These considerations are reflected in the recently published ODD standard specific for German railways, the DKE SPEC 99004: 2025-05 (DIN-DKE (2025)).

DKE SPEC 99004 builds upon existing standards such as (ISO (2023)) and the ASAM Open-

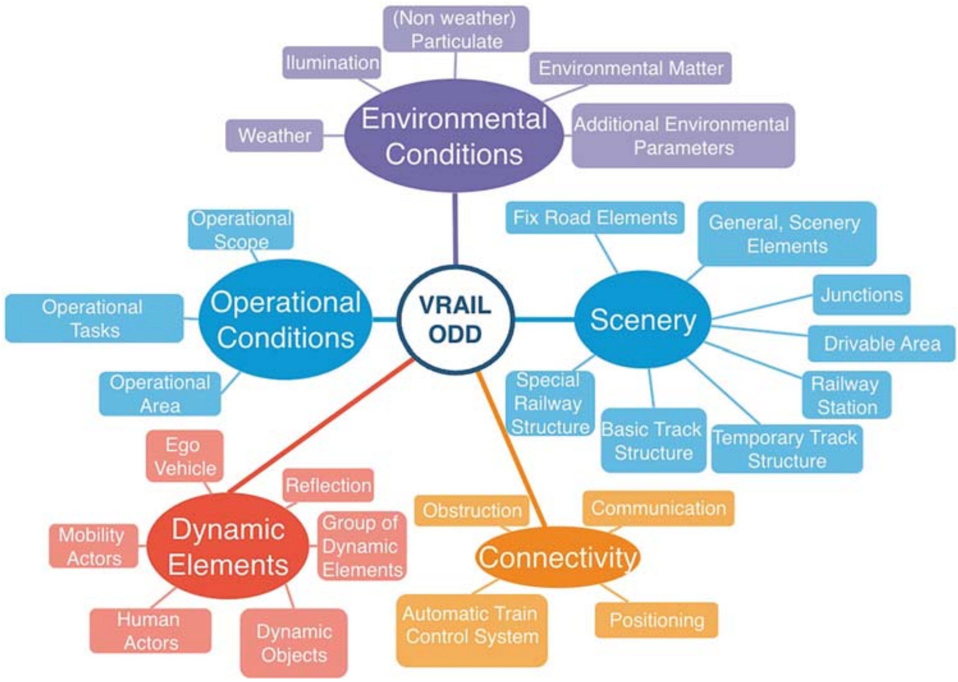

Fig. 1. Proposed two-level structure of the VRAIL ODD, showing the main ODD categories and their subcategories that are used as inputs for scenario generation.

ODD ASAM (2025). In the context of train systems, the ODD must capture the unique characteristics of rail operations and provide a structured boundary for the safe deployment of automated functions. Therefore, the standard was further modified to fit the needs of the railway domain, and important operating conditions for railway-specific AI applications such as Automatic Train Operation (ATO), for example environmental or track conditions, were covered.

However, the document does not provide specific information on explicit route-level ODD elements and the assessment of such an ODD system. Therefore, in the present work we have adapted the ODD by combining concepts from norms derived from the automobile industry and the DKE SPEC 99004. We have designed the ODD elements to be flexible to represent specific track profiles. Figure 1 shows the proposed ODD in the present paper, which we will refer to subsequently as VRAIL ODD. Here, VRAIL refers to the VRAIL project context in which the ODD structure was developed and applied. The figure includes only two levels of ODD elements, but there are up to four levels, for example Dynamic elements, Human Actor, Role, and Rail personnel. In addition, we have borrowed the term “attributes” from the DIN DKE SPEC 99004 to refer to descriptors beyond the last level of an ODD element. For example, a Human actor can have different poses, height, head-wear, and similar characteristics.

The main differences between DKE SPEC 99004: 2025-05 (DIN-DKE (2025)) and the ODD proposed in the present paper are:

(1) DIN DKE SPEC 99004 has three main categories, whereas VRAIL ODD has five. In VRAIL ODD, in addition to Scenery, a main category named Operational Conditions is included. Furthermore, Connectivity is treated as a main category, whereas in DIN DKE SPEC 99004 it is a subcategory of Environmental Conditions.

(2) VRAIL ODD provides a more detailed structure at the subcategory level compared to DIN DKE SPEC 99004.

In the next section, we describe how scenarios are generated in CARLA based on the defined ODD elements.

2.2. *Scenario ODD mapping*

To enable systematic scenario generation, we first define the operational use case for which scenarios are to be generated. The objective of this step is to establish a structured mapping between ODD elements and executable simulation scenarios. Figure 2 illustrates how the ODD elements covered by a given use case are reflected in the generated scenarios.

The mapping procedure follows a structured sequence of steps:

(1) Relevant track profiles are defined for the selected use case, for example tracks located in depot or station areas.
(2) A set of abstract scenario descriptions applicable to the selected track profiles is identified, such as a train operating at a defined speed while encountering an obstacle on the track.
(3) Based on the selected scenarios, one or more maps are created. Each map may cover one or multiple track profiles. For each map, the ODD elements related to Operational Conditions and Scenery (shown in blue in Figure 1) are identified, and the corresponding attributes, where applicable, are specified. This results in an explicit list of ODD elements covered by the map.
(4) In parallel, a collection of assets is defined within the simulation environment. These assets correspond to Dynamic Elements (shown in red in Figure 1). For each asset, the covered ODD elements and their attributes are documented, resulting in a traceable mapping between assets and ODD elements.
(5) For each scenario, an appropriate map and one or more assets are selected. Based on the mappings defined in the previous steps, the set of ODD elements and attributes covered by the scenario is inferred.
(6) Perturbations, corresponding to ODD elements shown in purple in Figure 1, are then applied to the scenario to introduce variability within the defined ODD boundaries.
(7) Finally, the complete set of ODD elements and attributes covered by each generated scenario is stored to enable traceability and coverage analysis.

This mapping process ensures that each generated scenario is explicitly linked to defined ODD elements and associated attributes, thereby supporting traceability and systematic coverage analysis.

Figure 3 summarizes the overall scenario generation pipeline implemented in the present work, from map and asset preparation to scenario execution. The individual pipeline components are described in Sections 2.3 to 2.4.

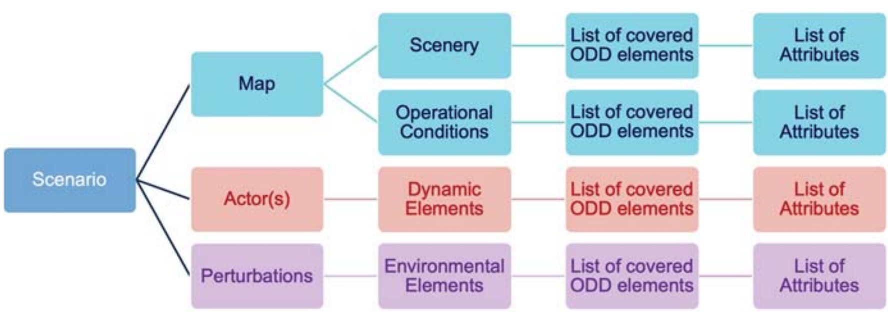

Fig. 2. Conceptual mapping between ODD elements and scenario artifacts, showing how track profiles, maps, assets, and perturbations contribute to ODD coverage at scenario level.

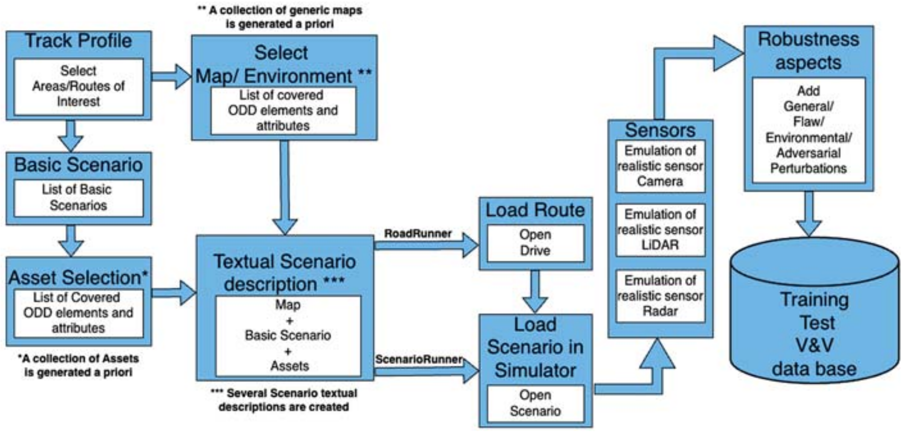

Fig. 3. Overview of the scenario generation pipeline implemented in the present work, illustrating the integration of ODD elements, map generation, scenario configuration, and execution within the CARLA simulation environment.

2.3. *Simulation environment and map generation*

CARLA (Car Learning to Act) is an open-source simulator for autonomous driving research (Dosovitskiy et al. (2017)) and is widely used in academia and industry for testing and validating automated driving functions in a controlled environment. CARLA provides capabilities such as high-fidelity simulation, sensor modeling (camera, LiDAR, radar), multi-agent simulation, physics-based dynamics, and standardized interfaces including OpenDRIVE and OpenSCENARIO. While CARLA is primarily designed for road traffic simulation, its modular architecture enables extensions to support non-road-based use cases.

In the present work, CARLA version 0.9.14 is used as the simulation environment for railway-relevant scenarios by incorporating static railway infrastructure into the environment model. Rail tracks, level crossings, station-level layouts, and signaling-related infrastructure are represented as part of the static map and integrated into the Unreal Engine environment.

2.3.1. *Map generation*

Maps are generated based on the characteristics of the track profiles relevant for the selected use case and are created using OpenDRIVE and RoadRunner. Each generated map is designed to support one or more scenarios by representing a specific spatial configuration of railway and surrounding infrastructure. Across scenarios, parameters such as the initial position of the ego train, the placement of scenario entities, and environmental conditions can be varied in a semi-automatic manner, enabling systematic exploration within the defined ODD boundaries.

2.3.2. *OpenDRIVE*

ASAM OpenDRIVE is an open standard for describing road networks in a structured, machine-readable format and is widely used in simulation and virtual testing to represent network topology and geometric properties. The specification defines:

(1) **Road geometry**: Curves, straight segments, elevations, and cross-sections.
(2) **Lane structure**: Lane types, widths, markings, and connectivity.
(3) **Traffic elements**: Signs, signals, and other roadside objects.
(4) **Junctions and connections**: How network segments are linked.

In the present work, OpenDRIVE is used primarily as a geometric and topological basis for modeling shared spaces, such as level crossings, where railway and road infrastructure interact.

2.3.3. *RoadRunner*

RoadRunner is a 3D road and environment design tool developed by MathWorks and is commonly used to create road networks and scenes for simulation-based testing of autonomous systems. Road geometry, intersections, lane markings, and roadside assets can be created in RoadRunner and exported as an OpenDRIVE (.xodr) file containing the network topology and geometric information. For visual representation, three-dimensional geometries, material definitions, and texture references can be exported using the Filmbox (.fbx) format. Railway-specific static elements that are not directly supported by RoadRunner, such as rail tracks and station platforms, are modeled directly within Unreal Engine and integrated consistently with the imported road infrastructure.

2.4. *Dynamic elements*

Dynamic elements correspond to ODD elements whose state changes during scenario execution. In the present work, dynamic elements include road vehicles, railway vehicles, and human actors interacting with the static environment defined in Section 2.3.

Dynamic elements are instantiated at runtime in the CARLA simulation environment and are parameterized using attributes associated with the corresponding ODD elements. These attributes include initial position, velocity, trajectory constraints, and interaction timing. The simulation setup is based on CARLA version 0.9.14, which provides native vehicle and pedestrian actor models as well as programmatic interfaces for controlling their state during simulation.

OpenSCENARIO is a standardized description language for specifying dynamic scenario content, including actor initialization, maneuvers, triggers, and environmental conditions. In the present work, OpenSCENARIO version 1.x descriptions are processed via the CARLA ScenarioRunner component. ScenarioRunner provides execution support for a subset of OpenSCENARIO 1.x constructs in CARLA 0.9.14, primarily for vehicle-oriented interactions.

Dynamic behaviours that are not covered by the OpenSCENARIO execution capabilities available in CARLA 0.9.14, including pedestrian behavior and railway-specific interaction patterns, are implemented using custom Python-based control logic in combination with structured scenario configuration files.

3. Use Case: Example scenario

This section presents an example scenario that instantiates the ODD-based scenario generation pipeline described in Section 2. The use case demonstrates how the defined ODD elements, scenario–ODD mappings, map artifacts, and scenario configuration mechanisms are applied to derive and execute a concrete simulation scenario in the railway context.

3.1. *Base Scenario and automatic scenario creation*

The use case is defined by a base scenario that specifies the operational situation to be analyzed. The base scenario is formulated as a parameterized textual description in which placeholders denote ODD elements and associated attributes. The description captures the interaction structure of the scenario without fixing concrete instantiations and serves as the starting point for scenario generation.

The base scenario textual description is: *"Train driving at* **{*speed(m/s)*}**. *The train is approaching a Curved Track with* **{*radius(m)*}**, *there is a* **{*Street Vehicle*}** *on the track".*

Concrete scenario instances are derived from the base scenario by instantiating the placeholders defined in the textual description. The elements indicated in bold and braces correspond to ODD elements and attributes introduced in Section 2.1, and their combination follows the ODD-to-scenario mapping procedure described in Section 2.2.

3.2. *Automatic scenario creation*

Concrete scenario instances are derived from the base scenario by instantiating the placeholders defined in the textual description. The elements indicated in bold and braces correspond to ODD elements and attributes introduced in Section 2.1, and their combination follows the ODD-to-scenario mapping procedure described in Section 2.2.

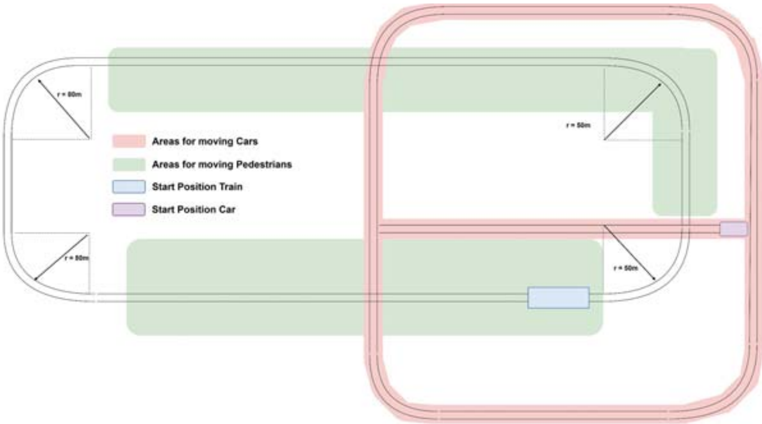

Fig. 4. Map layout used for the use case, showing the curved railway track, the intersecting road at the level crossing, and areas designated for street vehicles and human actors.

In this use case, **{Street Vehicle}** is instantiated by selecting a road-vehicle class (e.g. passenger car or larger road vehicle), while the attribute **speed** is assigned to the **Ego Train** ODD element. The resulting combinations define a set of concrete scenario instances derived from the same base scenario and are instantiated on a map that satisfies the base-scenario infrastructure requirement **{radius(m)}**.

3.3. *Map instantiation*

The map presented in Figure 4 shows the layout used for the use case, consisting of a railway track on which the **Ego Train** operates and a road that intersects the track at a level crossing. The layout includes lanes for **Street Vehicles**, the railway track geometry for the train trajectory, and designated areas for **Human Actors**.

Elements highlighted in red correspond to objects created using RoadRunner, including the road network and navigation infrastructure for street vehicles. The green areas and the railway track elements were created directly in Unreal Engine, as these object types are not directly supported by RoadRunner. The **Ego Train** and the crossing street vehicle, shown in blue, are instantiated dynamically during scenario setup and constitute the dynamic elements whose state evolves during scenario execution.

3.4. *Scenario execution*

Scenario generation for the use case is realised by combining OpenSCENARIO-based configuration with custom scenario control components, following the pipeline described in Section 2. OpenSCENARIO is used to initialise actors and to specify the vehicle-centric interaction at the level crossing. ODD elements corresponding to **Street Vehicles** and **Human Actors** are instantiated using either CARLA-provided actor models or custom three-dimensional assets created *apriori*. Each actor instance is referenced by a unique identifier within the OpenSCENARIO description. Scenario parameters derived from the base scenario instantiation, such as the train speed, are applied during scenario setup.

Within the present setup, OpenSCENARIO specifies the behavior of road vehicles and their interaction with the static environment. While sufficient for street vehicles, more complex behaviors, particularly pedestrian motion and railway-specific interactions, are configured using custom Python-based control logic. These behaviors are parameterized through YAML configuration files with a project-specific schema, which complements the OpenSCENARIO description without duplicating its functionality.

Figure 5 illustrates one instantiated scenario at the level crossing. Images A-D show the street

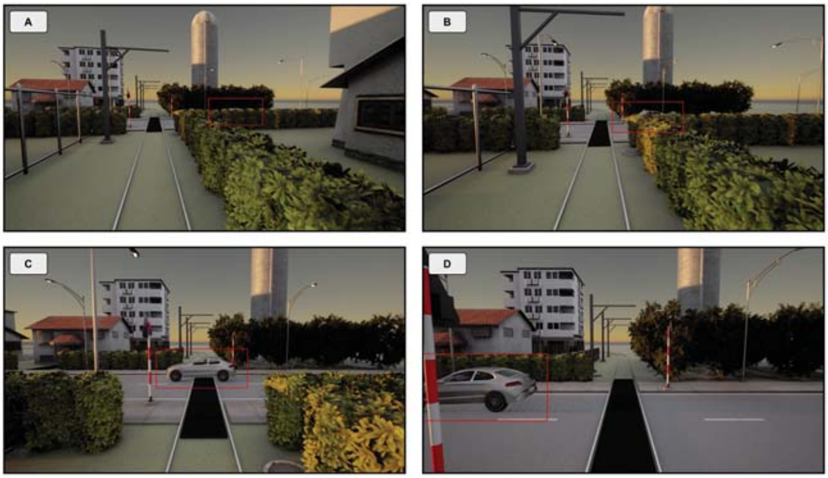

Fig. 5. Temporal progression of the interaction at the level crossing for one instantiated scenario, showing the street vehicle approaching, occupying, and clearing the crossing area while the train is approaching along the track.

vehicle approaching the crossing, occupying the track area, and clearing the crossing as the train approaches.

4. Discussion, Limitations and future work

This paper presented an ODD-driven pipeline that maps ODD elements to simulation artifacts to generate traceable and scalable railway scenarios in CARLA. The implementation shows that the pipeline preserves traceability, reduces manual scenario effort, and automatically produces consistent scenario variants from a single base description. Despite these benefits, complex railway behaviors still require custom logic, and integrating rail infrastructure remains more challenging than road networks. Future research will focus on leveraging GIS data for automated map generation and developing ODD coverage metrics to identify testing gaps. Expanding the scenario catalog to include degraded operational modes will further enhance the robustness of autonomous rail validation.

Acknowledgement

This work was conducted within the Innosuisse-funded project VRAIL, a collaboration between Stadler Signalling and ZHAW.

References

ANSI (2023). ANSI/UL 4600:2023, Standard for Evaluation of Autonomous Products.

ASAM (2025). ASAM OpenODD® 1.0.0.

BSI (2020). *Operational design domain (ODD) taxonomy for an automated driving system (ADS)*.

DIN-DKE (2025). DIN DKE SPEC 99004:2025, Specification of Operational Design Domain in Rail.

Dosovitskiy, A., G. Ros, F. Codevilla, A. Lopez, and V. Koltun (2017). CARLA: An Open Urban Driving Simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*.

ISO (2023). ISO 34503:2023, Road Vehicles - Test scenarios for automated driving systems - Specification for operational design domain.

SAE (2021). Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles.

Zeller, M., T. Waschulzik, R. Schmid, and C. Bahlmann (2024). Toward a safe mlops process for the continuous development and safety assurance of ml-based systems in the railway domain. *AI and Ethics 4*(1), 123–130.