Bidirectional Traceability between Datasets and ODDs: A Formal Approach for ML-Based Safety-Critical Development

Martin Rabe[1]

[1]Technische Universität Ilmenau, Germany

Abstract

The increasing use of machine learning (ML) components in safety-critical automotive systems demands rigorous engineering processes comparable to those prescribed for traditional software. Current safety standards focus on conventional software and do not define requirements for ML-based systems, leaving critical aspects such as dataset quality, traceability, and operational design domain (ODD) specification under-addressed.

Because existing standards lack ML-specific guidance, developers cannot ensure that datasets and models remain within the intended ODD throughout the development life-cycle. Traditional safety-critical processes are ill-suited for handling the unique challenges of ML, and the literature shows a concentration on model training and verification while domain specification, dataset management, and model integration remain under-represented. Consequently, there is no systematic method to achieve bidirectional traceability between datasets and ODDs, nor a clear understanding of the maturity of proposed ML-aware development methodologies.

We propose an analytical formalization of the ODD that enables bidirectional traceability between a dataset and its intended operational environment. This formalization is grounded on objectives for ML-based systems from the European Union Aviation Safety Agency (EASA) and is applied to three representative automotive use cases (autonomous emergency braking, freespace detection, and unmanned aircraft systems). In parallel, we conduct a systematic mapping study of existing ML-centric development methodologies to assess their coverage of the full development process. Building on the formalization, we introduce a novel dataset specialization process that automatically determines which data points satisfy the formalized ODD, allowing substantial dataset reduction while preserving performance.

The ODD formalization is expressive enough to capture the requirements of diverse automotive perception tasks and complies with EASA objectives, facilitating regulatory approval. The mapping study reveals that most research effort concentrates on model training and verification, whereas domain specification, dataset management, and integration receive significantly less attention, highlighting gaps for future work. Overall, the combined methodology provides a systematic, traceable, and efficient framework for developing safety-critical ML-based automotive systems.

Keywords

formal methods, dataset management, safety, certification, machine learning

1. Introduction

The automation of cars shows promise both in improvements to safety [1] and in optimizing traffic flow [2]. Key enablers on the software side are recent advances in machine learning (ML) such as object recognition [3], decision-making [4], and planning [5].

Safety Standards and their Limitations. Manufacturers in the automotive domain must comply with rigorous, safety-critical software development processes. The most relevant standard is the international ISO26262 [6] (2018), which prescribes a V-Model-based development life-cycle and a safety-case documentation approach. However, several studies have shown that the current ISO26262 framework does not readily accommodate ML-based systems [7, 8, 9], mainly because these systems follow a development process that differs from traditional software.

To address gaps in ISO26262, ISO21448 (SOTIF) was released in 2019 [10] (revised 2022). It focuses on hazards caused by performance limitations rather than faults, and introduces a cyclic development

Joint Proceedings of REFSQ-2026 Workshops, Doctoral Symposium, Posters & Tools Track, and Education and Training Track. Co-located with REFSQ 2026. Poznan, Poland, March 23-26, 2026

martin.rabe@tu-ilmenau.de (M. Rabe)

0000-0003-0697-233X (M. Rabe)

process for identifying triggering events. Nevertheless, ISO21448 still lacks explicit support for dataset management and other ML-specific needs.

In the aviation domain, the analogous standard DO-178C [11] faces the same limitation. The European Union Aviation Safety Agency (EASA) has responded with publications [12, 13] , which enumerate 72 objectives for ML-based system approval, yet a comprehensive, cross-domain solution remains missing.

Operational Design Domain (ODD). A critical yet under-represented aspect in both automotive and aviation ML development is the specification of the operational design domain (ODD). The SAEJ3016 standard defines ODD as: "Operating conditions under which a given driving automation system [...] is specifically designed to function, including but not limited to, environmental, geographical, and time-of-day restrictions, and/or the requisite presence or absence of certain traffic or roadway characteristics." Our prior work [14] highlighted the importance of a formal ODD definition for ensuring compliance throughout the ML development life-cycle, from dataset collection to model integration.

Problem Statement. Current safety-critical standards were devised for traditional software and do not sufficiently address the unique requirements of ML-based systems. In particular, the standards lack explicit mechanisms for dataset management, formal specification of the ODD, and continuous validation throughout the ML development life-cycle. This mismatch creates a gap between modern ML development practices and the certification processes needed to ensure the safe and reliable deployment of autonomous vehicles.

2. Related Work

In this section we review pertinent prior work on the specification of ODDs. One line of research closely aligned with our proposed approach, yet concentrating on specification rather than on mathematical formalization, is the ASAM OpenODD project initiated by the Association for Standardisation of Automation and Measuring Systems (ASAM) [15]. The project provides a dedicated specification language, expressed in XML, to describe driving scenarios. This language can encode attributes such as the number of vehicles present, their velocities, and the geometry and quantity of roadways. Moreover, the framework permits the definition of metrics for evaluating test-coverage, thereby allowing one to assess whether a particular scenario, e.g., a city-driving episode, lies within a vehicle's ODD. In contrast to our methodology, the ASAM effort does not address dataset management or dataset coverage, and it is primarily oriented toward automated-driving applications, whereas our work targets safety-critical systems more broadly. Irvine et al. [16] introduce a structural, natural-language-based approach for ODD specification. Their goal is to render the specification both human- and machine-readable, enabling a decisive check of whether a given situation falls within the prescribed ODD. The initial contribution delineates the rationale behind the specification language, intended for regulators and system engineers. The ODD is decomposed into an underlying domain model, capturing dynamic entities, scenery, and environmental conditions, and a set of language constructs (inclusion, exclusion, and conditional statements) that describe the scenery. Building on this foundation, Schwalb et al. [17] extends the earlier specification with formal operators that permit, for instance, the addition or removal of scenarios from an ODD definition. The authors also introduce mechanisms for modeling uncertainty and for specifying admissible risk levels. Nonetheless, neither of these publications discusses a direct linkage to the development life-cycle or to the associated data management activities. A formalization of the ODD together with an architectural design for runtime ODD assurance is proposed by Sun et al. [18]. The authors derive test cases for automated-driving systems (ADS) from a state-space representation of the dynamic environments that a vehicle may encounter during operation and testing. Unlike our approach, this work does not focus on the datasets employed to train ML-based systems. However, their methodology could be combined with ours to guarantee that the formalized ODD is respected at runtime. A methodology for identifying the ODD in which a system is currently operating is presented by Lee et al. [19]. The approach leverages statistical data and risk assessment to delineate the ODD and to verify whether the system can continue to operate safely. Similar to the previous method, it can serve as a runtime monitor. In the context of our work, such a technique could supply information

about an intended ODD, which could subsequently be formalized using our framework. Mendiboure et al. [20] conduct a review of ODD taxonomies, deriving a generic taxonomy that they apply to an automated-bus-station service scenario. This taxonomy is then used to extract system requirements from the example ODD. Scenario-based testing of ADS systems, with an emphasis on ODD coverage, is advocated by Weissensteiner et al. [21]. This perspective is orthogonal to our formalization and can be employed alongside it; our formalism can be used to define the scenarios that drive the testing process. The specification of ODDs in Gherkin syntax is proposed by Lauer et al. [22]. By encoding requirements in the Given-When-Then format of Gherkin files, the authors aim to improve stakeholder comprehension of ADS requirements, which are often expressed in natural language. This specification can serve as a starting point for defining the scenarios that underpin our formalization. Finally, Ollier et al. [23] presents a hazard-identification methodology grounded in a scenario-based ODD definition. This enables risk analysis of the specified ODD and can be employed to refine the ODD before it is formalized using our approach.

We now turn to literature concerning dataset management. Contributions in this area are often embedded within broader development methodologies. For example, Kurd et al. [24] stresses functional assurance, while [25] discuss W-Models with a focus on model development. Although dataset management is implicit in these methodologies, they lack concrete guidelines for handling datasets. Methods to improve dataset quality have also been proposed. O'Brien et al. [26] employ a fuzzy inference system to detect scenarios that violate traffic rules, e.g., tail-gating. A framework-based on statistical learning and model-based analysis is introduced by Roesener et al. [27] to evaluate the safety impact of automated driving functions and to identify relevant test scenarios for ML-based systems. While these techniques can enhance dataset management, the authors do not elaborate on practical data handling procedures. Enhancing datasets with simulated or synthesized data is a widely researched area. Rao et al. [28] propose augmenting existing datasets, particularly for rare events, using synthetic data. Similarly, Tian et al. [29], and Zhang et al. [30] employ synthetic data to generate test cases for models. However, these approaches primarily target testing rather than the improvement of model performance through dataset adaptation.

3. Research Method

The dissertation aims to bridge the identified gaps by:

1. Proposing a formal definition of ODD compliance that can be directly linked to requirements defined by the EASA.
2. Developing a framework that formalizes the ODD and integrates it into the safety-case documentation process.
3. Designing a quality-assessment process for evaluating dataset compliance with a given ODD.
4. Providing guidelines and case studies (automotive and aviation) that demonstrate how the formalization can be applied to specialize existing datasets, improve model training, and support safety-case arguments.

To achieve these goals, the research employs a scenario-driven methodology that combines several complementary activities. First, a literature review is performed by systematically mapping existing ODD specifications, dataset management practices, and safety-standard objectives such as ISO26262, ISO21448, and EASA guidelines. Second, the ODD is mathematically represented as a set of scenarios via a formal modeling step, which also applies formal quality properties. Third, an empirical evaluation applies this formalism to benchmark datasets (KITTI, Cityscapes, and nuScenes) to assess coverage, enable dataset specialization, and measure the impact on model performance. Finally, a traceability analysis subsequently links the ODD artifacts directly to the safety objectives, thereby facilitating the certification process.

4. Proposed Solution

In order to ensure that ML components for autonomous driving are developed and evaluated with respect to a well-defined ODD, we propose a systematic methodology that bridges the gap between high-level ODD specifications and concrete dataset requirements. The approach formalizes the ODD as a collection of scenarios, establishes mappings from these scenarios to data points, defines formal quality properties for assessing dataset coverage, and outlines a structured workflow for managing and reducing datasets to the essential elements required for reliable model training and testing.

4.1. Scenario Modeling

We propose scenario modeling to enable tracing from the ODD to the dataset, the model training, the model testing, and the model integration. The goal is to formalize the ODD in such a way that the formalization can be used to show compliance with requirements.

Towards this goal we argue to formally model the ODD as a set of scenarios. Each of these scenarios is composed of the ego vehicle, a set of intruders, and environmental conditions. The next step after defining these scenarios is to map data points to the defined scenarios. The data points can be either come from an existing dataset, or the set of scenarios can be used to guide the recording of data points for a new dataset. Once the data points are mapped to the scenarios we can apply quality properties to evaluate the support of the dataset in regard to the target ODD.

4.1.1. ODD Formalization Approach

We aim to develop a unified description of how dynamic entities move within an ODD. Both the ego platform, whether a ground-based vehicle or an unmanned aerial system (UAS), and any surrounding actors (the intruders) are characterized by three loosely coupled components the region of space they may occupy, the range of speeds they can attain, and the family of possible paths they might follow. In other words, each moving object is associated with a motion model that captures where it can be, how fast it can travel, and the shape of its trajectory through time.

The ego platform occupies a privileged spot in the scenario: its nominal position is the origin of the coordinate system, while its altitude (for a UAS) is explicitly represented because it directly influences the visual scene the platform perceives. All other actors, cars, drones, or any other intruders, are described by the same type of motion model, selected according to whether they move on the ground or in the air, and gathered into a collection of intruder models.

Beyond the moving participants, an ODD also includes a description of the surrounding environment. This environmental portrait captures those attributes that are relevant to the development and validation of the learning-based system, such as illumination levels, e.g., a minimum of 500lx and the type of operating area (highway, city, etc.). Each concrete environment is a point in this space, and the set of all such points constitutes the environmental domain for the ODD.

A scenario is then formed by coupling an ego motion model, a subset of intruder envelopes, and a particular environmental description. In this way, a scenario embodies a complete, self-contained story about how the ego platform and its neighbors may behave under specific conditions. The collection of all admissible scenarios constitutes the operational model for the system under study.

Because not every logical combination of motion models and environmental attributes yields a meaningful situation, e.g., a two-lane road on a highway does not accommodate a crossing car, constraints are imposed to prune the scenario space. These constraints are expressed as logical conditions that rule out implausible or irrelevant configurations.

All scenarios are grounded in empirical data. A dataset of recorded instances, each instance being a data point, provides concrete realizations of the abstract scenarios. By mapping each scenario to the subset of data points that instantiate it, we obtain a support for that scenario. This mapping, formalized as a support function, enables several practical workflows: during system development the entire dataset can be organized according to the scenario taxonomy, and later the data can be

split into training, validation, and test partitions that respect the scenario structure. In summary, the motion model framework offers a flexible, high-level language for describing dynamic entities, their interactions, and the surrounding context in a way that is both mathematically sound and accessible.

4.1.2. Formal Quality Properties

With the formalization approach, we have the means to define formal quality properties. They are based on the mapping from scenarios to data points in the dataset. With these properties, e.g., the coverage of the ODD with the given dataset can be evaluated. The goal is to facilitate at least the properties such as: completeness, which enables an initial quality assessment of the datasets regarding the coverage of the ODD; support sufficient, which allows for a minimum amount of data points to be available for a scenario; and support distribution, to assess that a given dataset represents the ODD unbiased.

4.2. Dataset Management Process

In this section we describe the proposed workflow for managing datasets used in the training of ML systems. The workflow leverages the ODD specification to derive precise dataset requirements.

The ODD constitutes a central artifact in the development of automotive ML systems and is therefore readily available to practitioners. By translating the ODD into concrete data requirements, the workflow enables the selection of only those data points that are essential for model training, thereby reducing computational load. Moreover, it highlights gaps in the existing data collection, guiding future data acquisition efforts. The process consists of five sequential steps:

Step 1: Abstract ODD Specification. An initial, high-level description of the ODD is produced in natural language, analogous to requirements elicitation. This description provides understandable overview of the intended operating environment for both domain experts and other stakeholders.

Step 2: Dimensional Decomposition. The abstract specification is analyzed to identify salient dimensions, e.g., road type, weather condition, presence of other vehicles. This ensures that each relevant dimension is represented in the dataset and forms the basis for subsequent validation activities.

Step 3: Formal ODD Specification. Building on the dimensions identified in Step2, a formal representation of the ODD is constructed. This formalization enumerates the set of scenarios that the ego vehicle may encounter, thereby defining the target operational model.

Step 4: Dataset Property Analysis. The characteristics of an existing dataset are examined with respect to the dimensions specified in Step2, e.g., ego vehicle velocity, environmental conditions such as weather, road type, or time of day. If the dataset contains samples covering all identified dimensions, it can be inferred that the dataset partially satisfies the ODD scenarios.

Step 5: Dataset Adaptation. Finally, the dataset is adapted depending on the outcome of step 4. This means on one hand that data points are pruned to retain only those required to represent the formalized ODD from Step3, using the dimensional criteria established in Step4. On the other hand if the analysis identified scenarios which are not at all, or not sufficiently covered, new data points need to be added the the dataset. This adaptation concentrates the dataset on the most relevant samples for the target domain, thereby exploiting the domain knowledge accumulated throughout the preceding steps.

5. Next Steps

The forthcoming research agenda will concentrate on translating the proposed formalism into a practical evaluation pipeline and extending its applicability across diverse learning scenarios. Key activities are:

Dataset Creation for Systematic Evaluation. The idea is to design a comprehensive, domain-specific dataset that encodes the formal ODD specifications as ground truth annotations. It shall include a balanced representation of environmental conditions, road geometries, and dynamic actor configurations to enable thorough coverage analysis.

Empirical Assessment on Established Benchmarks. In this activity we want to apply the dataset specialization process to widely used benchmarks such as KITTI, Cityscapes, and nuScenes.

Furthermore, we want to quantify reductions in dataset size, coverage metrics, and downstream model performance to demonstrate that safety-critical properties are preserved.

Evaluation Across a Broad Spectrum of ML Architectures. In this study we will train and evaluate representative architectures, specifically convolutional neural networks, transformer-based vision models, and hybrid sensor-fusion pipelines, using the adapted datasets. Subsequently, we will analyze how data adaptation driven by ODDs influences model robustness, generalization performance, and computational efficiency.

Integration with Safety-Case and Certification Workflows. We will map the formal ODD artifacts to the safety objectives stipulated by EASA and ISO, thereby producing traceability matrices for regulatory review. In parallel, we will prototype tooling that embeds the ODD verification step within continuous-integration pipelines, facilitating the development of safety-critical ML systems.

Collectively, these steps will provide a validated, standards-aligned methodology for ensuring that ML components operate safely within their intended operational design domains.

6. Conclusion

This dissertation shall address the critical gap between existing safety standards and the engineering realities of ML-based automotive systems. By introducing a mathematically-grounded formalization of the ODD, we enable bidirectional traceability from high-level safety requirements to concrete dataset elements. The formalism satisfies the EASA objectives, integrates seamlessly into safety-case documentation, and supports systematic assessment of dataset coverage through rigorously defined quality properties (completeness, sufficiency, and unbiased distribution).

The presented methodology constitutes a coherent, traceable, and scalable framework for the development of safety-critical ML components in automotive and aviation contexts. It bridges the divide between regulatory intent and practical implementation, offering a concrete pathway toward cross-domain certification and future standardization.

Declaration on Generative AI

During the preparation of this work, the author used GPT-OSS in order to: Grammar and spelling check. After using this tool, the author reviewed and edited the content and takes full responsibility for it.

References

[1] S. Kitajima, H. Chouchane, J. Antona-Makoshi, N. Uchida, J. Tajima, A Nationwide Impact Assessment of Automated Driving Systems on Traffic Safety Using Multiagent Traffic Simulations, IEEE Open Journal of Intelligent Transportation Systems 3 (2022).

[2] B. S. Kerner, Physics of automated driving in framework of three-phase traffic theory, Phys. Rev. E 97 (2018).

[3] J. Wu, J. Jiao, Q. Yang, Z.-J. Zha, X. Chen, Ground-aware point cloud semantic segmentation for autonomous driving, in: Proceedings of the 27th ACM International Conference on Multimedia, 2019.

[4] Y. Chen, C. Dong, P. Palanisamy, P. Mudalige, K. Muelling, J. M. Dolan, Attention-based hierarchical deep reinforcement learning for lane change behaviors in autonomous driving, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2019.

[5] D. Isele, R. Rahimi, A. Cosgun, K. Subramanian, K. Fujimura, Navigating occluded intersections with autonomous vehicles using deep reinforcement learning, in: IEEE International Conference on Robotics and Automation, 2018.

[6] ISO 26262 Road vehicles — Part 6: Product development at the software level, 2018.

[7] K. Heckemann, M. Gesell, T. Pfister, K. Berns, K. Schneider, M. Trapp, Safe automotive software, in: Knowledge-Based and Intelligent Information and Engineering Systems, 2011.
[8] A. Knauss, J. Schroder, C. Berger, H. Eriksson, Software-related challenges of testing automated vehicles, in: IEEE/ACM 39th International Conference on Software Engineering Companion, 2017.
[9] R. Salay, R. Queiroz, K. Czarnecki, An analysis of iso 26262: Machine learning and safety in automotive software, in: SAE Technical Paper, SAE International, 2018.
[10] ISO 21448 Road vehicles — Safety of the intended functionality, 2022.
[11] RTCA DO-178C Software Considerations in Airborne Systems and Equipment Certification, 2012.
[12] Artificial Intelligence Roadmap, Technical Report, EASA, 2020.
[13] EASA Concept Paper: First usable guidance for Level 1 machine learning applications, 2021.
[14] M. Rabe, S. Milz, P. Mäder, Development methodologies for safety critical machine learning applications in the automotive domain: A survey, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2021.
[15] ASAM OpenODD, ASAM OpenODD, https://www.asam.net/index.php?eID=dumpFile&t=f&f=4544&token=1260ce1c4f0afdbe18261f7137c689b1d9c27576, 2021.
[16] P. Irvine, X. Zhang, S. Khastgir, E. Schwalb, P. Jennings, A two-level abstraction odd definition language: Part i, in: IEEE International Conference on Systems, Man, and Cybernetics, 2021.
[17] E. Schwalb, P. Irvine, X. Zhang, S. Khastgir, P. Jennings, A two-level abstraction odd definition language: Part ii, in: IEEE International Conference on Systems, Man, and Cybernetics, 2021.
[18] C. Sun, Z. Deng, W. Chu, S. Li, D. Cao, Acclimatizing the Operational Design Domain for Autonomous Driving Systems, IEEE Intelligent Transportation Systems Magazine 14 (2022).
[19] C. W. Lee, N. Nayeer, D. E. Garcia, A. Agrawal, B. Liu, Identifying the Operational Design Domain for an Automated Driving System through Assessed Risk, in: IEEE Intelligent Vehicles Symposium, 2020.
[20] L. Mendiboure, M. L. Benzagouta, D. Gruyer, T. Sylla, M. Adedjouma, A. Hedhli, Operational Design Domain for Automated Driving Systems: Taxonomy Definition and Application, in: IEEE Intelligent Vehicles Symposium, 2023.
[21] P. Weissensteiner, G. Stettinger, S. Khastgir, D. Watzenig, Operational Design Domain-Driven Coverage for the Safety Argumentation of Automated Vehicles, IEEE Access 11 (2023).
[22] C. Lauer, C. Sippl, Benefits of Behavior Driven Development in Scenario-based Verification of Automated Driving, in: IEEE 25th International Conference on Intelligent Transportation Systems, 2022.
[23] G. Ollier, D. Razafindrabe, M. Adedjouma, S. Gerasimou, C. Mraidha, Using Operational Design Domain in Hazard Identification for Automated Systems, in: 18th European Dependable Computing Conference, 2022.
[24] Z. Kurd, T. Kelly, J. Austin, Developing artificial neural networks for safety critical systems, Neural Comput. Appl. 16 (2006).
[25] Z. Kurd, T. Kelly, Safety lifecycle for developing safety critical artificial neural networks, in: Computer Safety, Reliability, and Security, 2003.
[26] M. O'Brien, K. Neubauer, J. Van Brummelen, H. Najjaran, Analysis of driving data for autonomous vehicle applications, in: IEEE International Conference on Systems, Man, and Cybernetics, 2017.
[27] C. Roesener, J. Hiller, H. Weber, L. Eckstein, How safe is automated driving? human driver models for safety performance assessment, in: IEEE 20th International Conference on Intelligent Transportation Systems, 2017.
[28] Q. Rao, J. Frtunikj, Deep learning for self-driving cars: Chances and challenges, in: Proceedings of the 1st International Workshop on Software Engineering for AI in Autonomous Systems, 2018.
[29] Y. Tian, K. Pei, S. Jana, B. Ray, Deeptest: Automated testing of deep-neural-network-driven autonomous cars, in: Proceedings of the 40th International Conference on Software Engineering, 2018.
[30] M. Zhang, Y. Zhang, L. Zhang, C. Liu, S. Khurshid, Deeproad: Gan-based metamorphic testing and input validation framework for autonomous driving systems, in: 33rd IEEE/ACM International Conference on Automated Software Engineering, 2018.