SAFETY ASSURANCE FRAMEWORK FOR CONNECTED, AUTOMATED MOBILITY SYSTEMS

D5.2

Harmonised descriptions for content of CCAM safety assessment data framework

Project short name
SUNRISE

Project full name
Safety assUraNce fRamework for connected, automated mobIlity SystEms

Horizon Research and Innovation Actions | Project No. 101069573
Call HORIZON-CL5-2021-D6-01

Funded by the European Union

ccam-sunrise-project.eu/

Dissemination level	Public (PU) - fully open
Work package	WP5: Content harmonisation of scenario data framework
Deliverable number	D5.2: Harmonised descriptions for content of CCAM safety assessment data framework
Deliverable responsible	Jobst Beckmann, ika
Status - Version	Final – V1.0
Submission date	30/05/2025
Keywords	Ontology, Scenario Content, Scenario Formats, Scenario Description, Database content, Database formats, Exposure metrics, Common language, Interoperability, Harmonisation, Interfaces

Authors

Role	Name
Main author	Jobst Beckmann (ika)
Contributing authors	Xizhe Zhang (UoW), Erwin de Gelder (TNO), Edwin van Hassel (Siemens), Marcos Nieto (Vicomtech), Ghada Ben Nejma (VEDECOM), Maximilian Grabowski (BASt)

Quality Control

	Name	Organisation	Date
Peer review 1	Sjef van Montfort	TNO	16/05/2025
Peer review 2	Anastasia Bolovinou	ICCS	13/05/2025
Peer review 3	Stefan de Vries	IDIADA	30/05/2025

Version history

Version	Date	Author	Summary of changes
0.1	04/04/2025	Jobst Beckmann	First draft of document structure and content
0.2	02/05/2025	Jobst Beckmann	Document refinement after partner review
0.3	26/05/2025	Jobst Beckmann	Document refinement after internal review
1.0	30/05/2025	Jobst Beckmann	Document ready for submission

Legal disclaimer

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS AND ACRONYMS

Abbreviation	Meaning
AD	Automated Driving
ADAS	Advanced Driver Assistance System
ADS	Automated Driving System
AEB	Autonomous Emergency Braking
AGO	Automotive Global Ontology
API	Application Programming Interface
AQCG	Automated Query Criteria Generation
ASAM	Association for Standardizations of Automation and Measuring Systems
AUTO	Automotive Urban Traffic Ontology
BSI	British Standards Institution
CCAM	Connected, Cooperative, and Automated Mobility
COD	Current Operational Domain
COTSATO	COncretizing Test Scenarios and Associating Test Objectives
CTDO	Connected Traffic Data Ontology
DDT	Dynamic Driving Task
DF	Data Framework
DSL	Domain-Specific Language
HDF5	Hierarchical Data Format version 5
HQDM	High Quality Data Model
ISMR	In-Service Monitoring and Reporting
ITS	Intelligent Transportation Systems
JSON	JavaScript Object Notation
KB	Knowledge Base
KG	Knowledge Graph

NATM	New Assessment/Test Method for Automated Driving
ODD	Operational Design Domain
OEM	Original Equipment Manufacturer
OSI	Open Simulation Interface
PDF	Probability Density Function
RDF	Resource Description Framework
SAF	Safety Assurance Framework
SCDB	SCenario DataBase
SOSA	Sensor, Observation, Sample and Actuator
SUNRISE	Safety assUraNce fRamework for connected, automated mobIlity SystEms
SUT	System Under Test
SWRL	Semantic Web Rule Language
TOD	Target Operational Domain
TOD	Target Operational Domain
UC	Use Case
URI	Unique Resource Identifier
V&V	Verification and Validation
VSSO	Vehicle Signal and Attribute Ontology
XML	eXtensible Markup Language

EXECUTIVE SUMMARY

Safety assurance of Cooperative, Connected, and Automated Mobility (CCAM) systems is a crucial factor for their successful adoption in society, yet it remains a significant challenge. It is generally acknowledged that for higher levels of automation, the validation of these systems by conventional test methods would be infeasible. Furthermore, certification initiatives worldwide struggle to define a harmonised safety assurance approach enabling massive deployment of CCAM systems.

The **SUNRISE** project develops and demonstrates a **CCAM Safety Assurance Framework** (SAF). The overall objective of the SUNRISE project is to accelerate the large-scale and safe deployment of CCAM systems. In alignment with international twin projects and initiatives, the project aims to achieve this objective by providing a SAF consisting of three main components: a Method, a Toolchain and a Data Framework. The **Method** is established to support the SAF safety argumentation, and includes procedures for scenario selection, sub-space creation, dynamic allocation to test instances and a variety of metrics and rating procedures. The **Toolchain** contains a set of tools for safety assessment of CCAM systems, including approaches for virtual, hybrid and physical testing. The **Data Framework** provides online access, connection and harmonization of external Scenario Databases (SCDBs), allowing its users to perform query-based extraction of safety relevant scenarios, allocation of selected scenarios to a variety of test environments, and reception of the test results.

This deliverable introduces both a **harmonised ontology** and **harmonised data formats** for use within the SUNRISE project and its associated partner databases. The ontology establishes a **shared understanding** of scenario-related concepts across all components of the SAF. This common understanding is particularly vital for enabling seamless **interoperability** between the SUNRISE Method and the Data Framework, which connects and federates multiple scenario databases. In addition, this shared understanding enables the use of diverse tools within **the Harmonized V&V Simulation Framework**, all of which must align on the meaning of the scenario content to function effectively within the same simulation environment.

Within the Data Framework, the **querying process** leverages the ontology to identify and retrieve logical scenarios. To ensure that scenario concepts are interpreted **consistently** by SAF users, SUNRISE tools, and connected databases, the use of a harmonised ontology is essential. This ontology is built upon **widely accepted industry standards** to support ease of adoption and integration. While it does not aim to capture the full diversity of all SCDB-specific concepts, it defines a **common semantic foundation**.

In parallel, the adoption of harmonised formats ensures that scenario content is represented in a **consistent structure** across interfaces within the SAF. This consistency is critical for enabling tools within the SUNRISE Method to process, interpret, and exchange data reliably. This deliverable outlines the formats employed throughout the SUNRISE project, specifying where and how they are applied within the SAF. Emphasis is placed on the use of **open, widely supported standards**, which promote **interoperability, transparency, and long-term maintainability**.

1 INTRODUCTION

1.1 Project introduction

Safety assurance of Connected, Cooperative, and Automated Mobility (CCAM) systems is a crucial factor for their successful adoption in society, yet it remains a significant challenge. CCAM systems need to demonstrate reliability in all driving scenarios, requiring robust safety argumentation. It is acknowledged that for higher levels of automation, the validation of these systems by means of real test-drives would be infeasible. In consequence, a carefully designed mixture of physical and virtual testing has emerged as a promising approach, with the virtual part bearing more significant weight for cost efficiency reasons.

Worldwide, several initiatives have started to develop test and assessment methods for Automated Driving (AD) functions. These initiatives already transitioned from conventional validation to a scenario-based approach and combine different test instances (physical and virtual testing) to avoid the million-mile issue.

The initiatives mentioned above, provide new approaches to CCAM validation, and many expert groups formed by different stakeholders, are already working on CCAM systems' testing and quality assurance. Nevertheless, the lack of a common European validation framework and homogeneity regarding validation procedures to ensure safety of these complex systems, hampers the safe and large-scale deployment of CCAM solutions. In this landscape, the role of standards is paramount in establishing common ground and providing technical guidance. However, standardising the entire pipeline of CCAM validation and assurance is in its infancy, as many of the standards are under development or have been very recently published and still need time to be synchronised and established as common practice.

Scenario Databases (SCDBs) are another issue tackled by several initiatives and projects, that generally tends to silo solutions. A clear concrete approach should be used (at least at European level), dealing with scenarios of any possible variations, including the creation, editing, parameterisation, storing, exporting, importing, etc. in a universally agreed manner.

Furthermore, validation methods and testing procedures still lack appropriate safety assessment criteria to build a robust safety case. These must be set and be valid for the whole parameter space of scenarios. Another level of complexity is added, due to regional differences in traffic rules, signs, actors and situations.

Evolving from the achievements obtained in HEADSTART and taking other project initiatives as a baseline, it becomes necessary to move to the next level in the development and demonstration of a commonly accepted **Safety Assurance Framework** (**SAF**) for the safety validation of CCAM systems, including a broad portfolio of Use Cases (UCs) and comprehensive test and validation tools. This will be done in **SUNRISE**, which stands for **S**afety ass**U**ra**N**ce f**R**amework for connected, automated mob**I**lity **S**yst**E**ms.

The SAF is the main product of the SUNRISE project. As the following figure indicates, it takes a central role, fulfilling the needs of different automotive stakeholders that all have their own interests in using it.

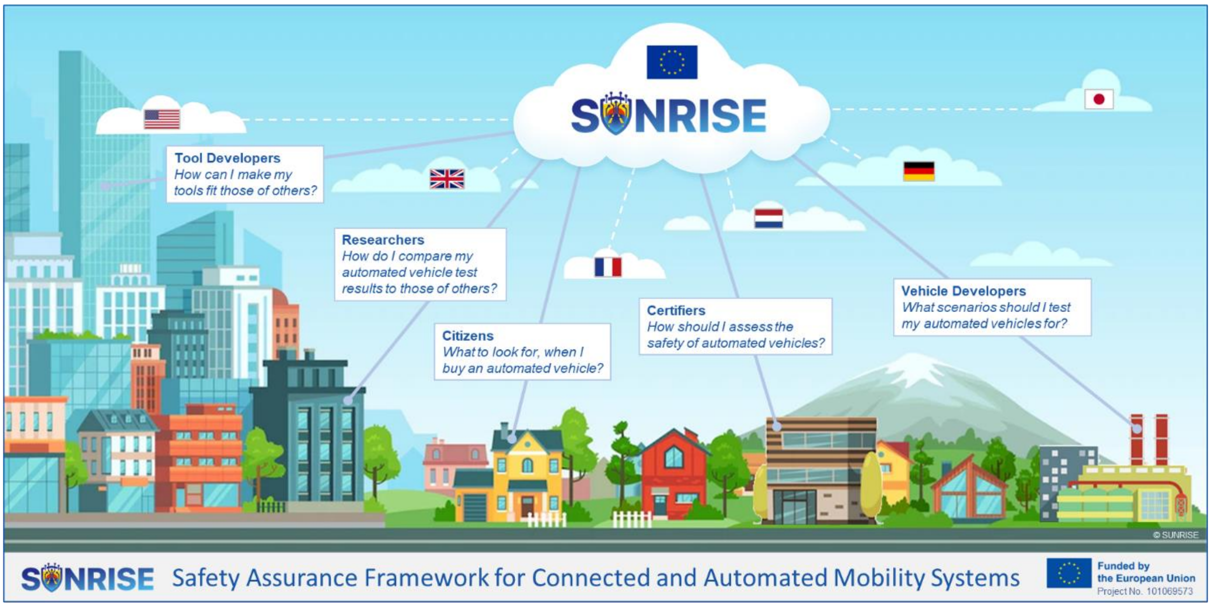

Figure 1: Safety Assurance Framework stakeholders

The **overall objective** of the SUNRISE project is to accelerate the safe deployment of innovative CCAM technologies and systems for passengers and goods by creating demonstrable and positive impact towards safety, specifically the EU's long-term goal of moving close to zero fatalities and serious injuries by 2050 (Vision Zero), and the resilience of (road) transport systems. The project aims to achieve this objective by providing a SAF consisting of three main components: a Method, a Toolchain and a Data Framework. The **Method** is established to support the SAF safety argumentation, and includes procedures for scenario selection, sub-space creation, dynamic allocation to test instances and a variety of metrics and rating procedures. The **Toolchain** contains a set of tools for safety assessment of CCAM systems, including approaches for virtual, hybrid and physical testing. The **Data Framework** provides online access, connection and harmonization of external Scenario Databases (SCDBs), allowing its users to perform query-based extraction of safety relevant scenarios, allocation of selected scenarios to a variety of test environments, and generation of the test results. The SAF will be put to the test by a series of **Use Cases demonstrations**, designed to identify and solve possible errors, gaps and improvements to the underlying methods, tools and data.

Following a common approach will be crucial for present and future activities regarding the testing and validation of CCAM systems, allowing to obtain results in a standardised way, to improve analysis and comparability, hence maximising the societal impact of the introduction of CCAM systems.

The following figure shows the general workplan of the SUNRISE project.

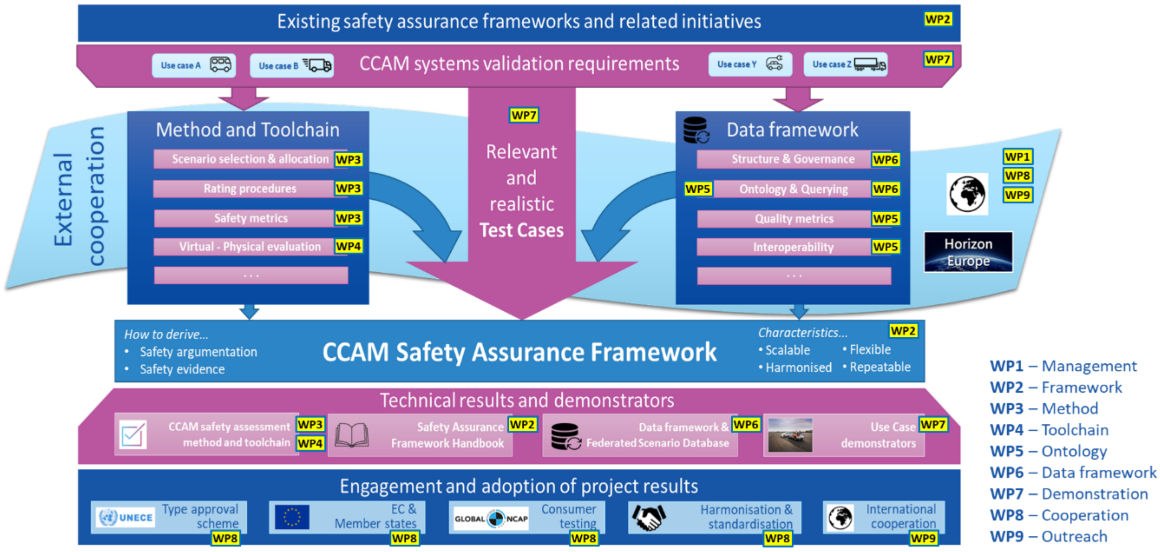

Figure 2: Workplan of the SUNRISE Project

1.2 Purpose of deliverable

This deliverable aims to establish a harmonised approach to managing the content of the SUNRISE Data Framework, ensuring consistency and interoperability across different scenario databases within the project. By defining a common structure for driving scenarios and operational design domains (ODDs), it supports the integration of various parts of the SUNRISE Safety Assurance Framework such as simulation, testing, and validation.

A key aspect of this harmonization is the development of a harmonised ontology that provides a standardized representation of driving scenarios. This ontology defines essential concepts, relationships, and attributes, ensuring that different databases can interpret and access scenario data in a consistent manner. By serving as a shared language, it facilitates data exchange and improves usability across different components of the SUNRISE SAF.

Additionally, this deliverable gives guidelines on how the ontology should be used and integrated into the querying process needed for the SUNRISE Data Framework. To facilitate this, examples are provided to integrate the ontology with the formats used within WP6 to create the SUNRISE Data Framework.

In addition to structuring the content, this deliverable also addresses the various formats used within the Data Framework to store and transfer this content. The deliverable gives guidelines on the use of such formats at various points within the SUNRISE Data Framework.

1.3 Intended audience

This deliverable is intended for all stakeholders, both internal and external, who require a clear understanding of the structure and format of data managed within the SUNRISE Safety Assurance Framework and the SUNRISE Data Framework. As such it is of particular interest to WP6, as the results of the deliverable directly influence the work to create the Data Framework in this work package. Noteworthy is especially task T6.2 which standardises outputs from SCDBs. Here, the ontology plays an important role as the basis to define relevant queries that enable search for specific scenarios in a SCDB. It is also relevant for database owners and users, providing essential guidance on the data that needs to be made available across different databases, so that a connection can be made to SUNRISE, and the data that can be accessed or requested by users.

By defining a common ontology this document ensures that stakeholders can effectively navigate the framework, facilitating seamless data exchange and interoperability.

1.4 Deliverable structure and relation to other parts of project

The content of the deliverable is divided into three main chapters. Chapter 2 describes the type of scenario content that needs to be harmonised within the SUNRISE project. Chapter 3 describes how this content is structured within a harmonised ontology. Finally, Chapter 4 describes how scenario content can be represented within different formats and gives guidelines on the various formats to be used at different points within the SAF.

This deliverable is closely related to the work performed around the SUNRISE Data Framework in WP5 and WP6. In the deliverable D5.1 requirements have been defined for the Data Framework content that are relevant for the ontology description. In WP6 input and output processes to databases connected via the SUNRISE federated layer are defined. Within this work the structure of this data and the relevant formats for its description defined in this deliverable need to be considered. In task T6.2 the queries to the individual connected databases will be defined, to enable the search functionality of the SUNRISE DF. The ontology forms the basis for what elements can be queried and is thereby a vital prerequisite for this task.

In addition, the common language provided by the ontology is essential for the Harmonized V&V Simulation Framework developed in WP4. It is critical that all tools within the Simulation Framework interpret and simulate scenario content consistently, exactly as it was originally understood when created and stored in the scenario databases. Achieving this consistency requires not only a shared ontology but also harmonized data formats that enable seamless interchange of scenario content between different tools.

2 DATA FRAMEWORK CONTENT DESCRIPTION

The SUNRISE Data Framework (DF) implements the data management layer of the SUNRISE Safety Assurance Framework (SAF), with a particular focus on the governance, structuring, and utilization of Scenario Databases (SCDBs). It provides a federated set of services that allow unified access to distributed scenario data, while respecting the autonomy and data policies of individual SCDB providers.

The Data Framework ensures that scenario data can be harmonized, queried, analysed, and utilized effectively within the SAF. Key concepts that structure the Data Framework include scenarios, the scenario databases, and the Operational Design Domain (ODD). These elements are grounded in a shared ontology, ensuring consistent interpretation across tools and stakeholders.

2.1 Scenarios

A scenario is a structured description of a temporal and spatial traffic constellation, used to assess the behaviour of automated driving systems under various conditions. Within the ISO 34501 [1] the scenario is also defined as a sequence of scenes, snapshots of all entities, that include the subject vehicle in the process of performing the dynamic driving task (DDT). The DDT includes all functions and behaviour to operate a vehicle, excluding the navigation. The scenario definition is further refined by [2] and [3], which introduce further levels of abstraction.

A scenario plays a central role in the SUNRISE Safety Assurance Framework by providing a dynamic representation of situations in which automated driving systems could operate. Unlike a static dataset, a scenario captures not only the spatial configuration and environmental conditions but also the behavioural interactions between various traffic participants, such as vehicles, pedestrians, and cyclists.

This behavioural component is essential: scenarios model how different agents act and react within a given context. For instance, a vehicle approaching a pedestrian crossing while a cyclist enters the scene is not only a configuration of objects and positions, but a sequence of decisions and actions that unfolds over time.

Scenarios can be categorized based on the degree of formalization and specificity, into four distinct types can be seen in Figure 3 (as defined in ISO 34501 and further expanded on by [2]):

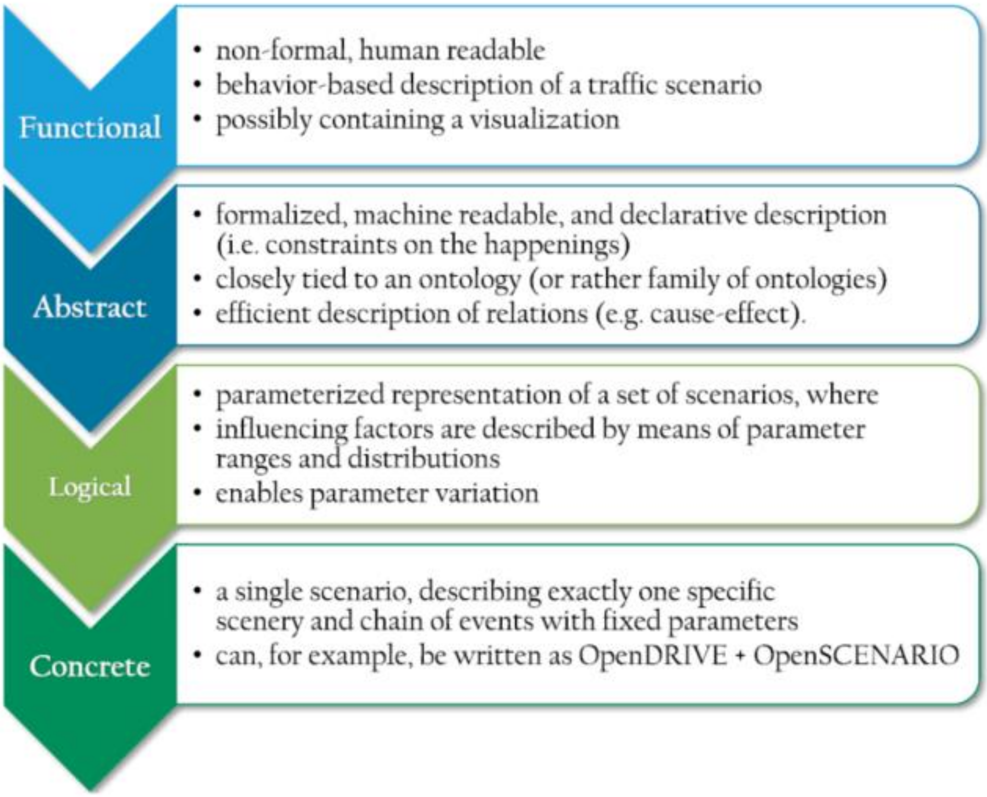

Figure 3: Different scenario abstraction levels with their characteristics [1]

This layered structure allows scenarios in different forms of abstraction to be used for high-level functional requirements through to testable, executable cases.

2.2 Scenario Databases

A Scenario Database (SCDB) is a structured repository developed to store, manage, and retrieve scenarios across various levels of abstraction. In the context of the SUNRISE Safety Assurance Framework, these databases are a central resource for enabling data-driven safety assessment of CCAM systems. The SUNRISE Data Framework provides federated access to multiple SCDBs, allowing users to interact with distributed data sources through a unified and harmonised interface. This approach preserves the governance and ownership rights of individual data providers while promoting interoperability across platforms.

The SCDBs connecting to the SUNRISE Data Framework are designed to support structured storage of scenarios along with their associated metadata. They include query capabilities that enable users to filter and select scenarios based on attributes of the Operational Design Domain (ODD), specific parameter values, or scenario categories. Access and authorisation controls are also integrated to ensure that data usage complies with the business models and data policies of third-party SCDB providers.

Interoperability is further enhanced by adhering to standardised formats such as OpenSCENARIO, and by using a shared ontology that defines a semantic schema for the content. In addition to storage and access, the Data Framework supports advanced data processing services. These services allow users to explore and filter SCDB contents, analyse search results, and prepare scenario data for use within the SAF testing pipeline.

2.3 Operational Design Domain (ODD)

The Operational Design Domain (ODD), as defined by ISO 34503 and adopted within SUNRISE, delineates the boundaries of the operational environment in which an automated driving system (ADS) is expected to perform the Dynamic Driving Task (DDT) safely. The ODD describes the specific conditions that must be satisfied for the system to function as intended, ensuring safety and regulatory compliance throughout its deployment.

An ODD encompasses a wide range of dimensions. These include road infrastructure characteristics such as the types of roads and intersections, the presence of lane markings, and other structural elements. Environmental conditions like weather, lighting, and temperature are also part of the ODD. Geospatial constraints, such as specific cities, regions, or geofenced areas, are included, as are temporal constraints that may restrict system operation to certain times of day, seasons, or other time-based limitations.

Within the SUNRISE framework, the ODD serves multiple critical roles. It acts as a filter for selecting or generating scenarios that are applicable to a given system's intended operational context. It also functions as a boundary condition for defining and validating performance claims, ensuring that systems are only evaluated within their declared domain. Furthermore, the ODD provides a reference model for tagging and structuring scenarios using the shared ontology, facilitating consistency and traceability.

By representing ODDs in a machine-interpretable form, they can be used programmatically within simulation workflows. This enables automated selection of relevant scenarios and supports systematic coverage analysis. The structure and semantics of ODD representations in SUNRISE are informed by ongoing standardisation efforts, including BSI PAS 1883 and relevant ISO initiatives, which seek to create consistent and interoperable standards across the industry.

To support simulation and scenario-based testing, the ODD must be both precise and machine-interpretable. It often forms a filtering layer for selecting or generating relevant test scenarios. For instance, only scenarios that match the defined ODD are used during virtual testing of an ADS. Efforts are ongoing to standardize the formal representation of ODDs. These initiatives aim to provide structured formats that make ODDs easier to share, interpret, and use across development tools and platforms.

While scenarios and ODDs are closely related within the SUNRISE framework, they serve distinctly different roles. Scenarios are event-driven and include time-dependent behaviour; they are used to test and validate how a system reacts to complex situations. In contrast, ODDs are environmental constraints, defining the conditions under which a system is intended to function but not how it should behave within those conditions.

Scenarios contain an explicit temporal structure, including potential interactions and safety-critical events. They are designed to simulate the decision-making environment of an ADS.

ODDs, on the other hand, form a boundary filter: a scenario is only considered valid for a given system if it is situated entirely within that system's ODD.

Other related terms are defined in ISO 34503 as well. The Target Operational Domain (TOD) refers to the set of operating conditions in which an Automated Driving System (ADS) is expected to operate, including factors such as environmental, geographical, and time-of-day restrictions, as well as the presence or absence of specific traffic or roadway characteristics. While the ODD defines the conditions the ADS is designed to handle, the TOD focuses on the location or area where the ADS will be deployed, which may include conditions outside its ODD. In contrast, the Current Operational Domain (COD**)** describes the specific set of operating conditions that exist at any given moment in the immediate vicinity of the ADS, covering the same kinds of factors, environmental, geographical, temporal, and traffic-related, but reflecting the system's present context rather than its expected or planned deployment area.

2.4 Role of Ontologies

To promote consistency, semantic clarity, and automation in scenario modelling, the SUNRISE Data Framework incorporates a dedicated ontology for traffic scenarios. This ontology provides standardised definitions for key scenario concepts such as vehicles, road elements, and manoeuvres, ensuring that all stakeholders interpret these elements in a consistent manner. Beyond terminology, the ontology captures the semantic relationships between entities, for example, causal links, dependencies, and interactions, which are essential for accurately modelling real-world driving situations.

The ontology also supports technical integration by enabling the translation of scenario definitions into scenario formats. This capability ensures compatibility with simulation tools, scenario generation engines, and other digital platforms used in the safety assurance process and the Harmonized V&V Simulation Framework within SUNRISE specifically. Moreover, the ontology serves as a schema for designing and querying scenario databases, thereby improving the efficiency and reliability of data retrieval.

By embedding ontologies into the Data Framework, the SUNRISE Safety Assurance Framework achieves semantic interoperability across its components. This ensures that scenarios, ODDs, and system specifications are interpreted consistently across tools, organisations, and development phases, ultimately strengthening the credibility and robustness of the overall safety assurance process.

3 ONTOLOGIES

3.1 Introduction

Ontologies are structured frameworks used to model the structure of a domain. They provide a formal representation of concepts, their relationships, and attributes, creating a shared understanding of a domain. Ontologies simplify complex systems by offering an abstract view of the world, where concepts are defined, organized hierarchically, and enriched with semantic relationships and rules.

An ontology consists of key components such as concepts, which define the domain's elements; relations, which establish connections between these concepts; individuals, which are instances of the concepts; and attributes, which further describe them. Additionally, axioms are included to set constraints and refine the interpretation of the concepts and their relationships.

The main advantages of ontologies lie in their ability to formalize and share knowledge. They create a common vocabulary that multiple users or systems can rely on, ensuring consistency and clarity. Ontologies are machine-readable, which allows computers to process and use the knowledge for tasks such as reasoning, decision-making, and data integration. By organizing concepts hierarchically, they enable inference, allowing new knowledge to be derived from existing relationships and rules.

In the context of autonomous driving, ontologies are particularly valuable. They are used to generate scenarios for testing automated vehicles and to evaluate the vehicle's perceived environment.

3.2 Literature Review on Ontologies

Ontologies have been extensively studied in literature, both in a general sense and specifically within the domain of driving scenarios. The following review provides an overview of key ontology concepts, methodologies, and applications, with particular attention to their relevance for the SUNRISE context.

3.2.1 Ontology Matching and Interoperability

A foundational review by [4] explores ontology matching literature from 2003 to 2013. Ontology Matching is a critical process in the field of semantic web, knowledge engineering, and information integration. It involves identifying correspondences between entities (such as classes, properties, or instances) in different ontologies. The goal is to enable interoperability and data integration across systems that use different but related ontologies. Interoperability enables the exchange and communication between ontologies and scenario databases. The literature review provides a comprehensive overview of the state of research at the time in ontology matching with matching techniques, matching systems, processing frameworks, practical applications, and evaluation. Matching techniques can be separated into top-down and bottom-up approaches, where the former starts with a high-level, abstract view of the problem and then breaks into smaller, more specific components. The latter with specific, concrete examples or techniques and then groups them into broader categories on shared characteristics. The variety of matching techniques is listed in the publication. Matching systems are tools or frameworks designed to align or match different ontologies. The publication presents an exhaustive list of matching systems that are based on different matching techniques, thus having different applications based on the ontologies.

For the harmonization of scenario databases, the method of ontology matching might be a promising approach. It could facilitate or automate the integration of different scenario databases into the SUNRISE Data Framework.

3.2.2 Ontologies for Scenario-Based Testing and Decision-Making

In [5], key representative ontologies for scenario-based testing in the field of automated and autonomous driving are summarized. Two main applications for ontologies in this field are identified with the scenario-based validation and the situation assessment & decision-making. In the scenario-based validation, ontologies are used to create scenarios with algorithms for the scenario-based testing. Ontologies used in the field of situation assessment & decision-making are to improve the situational awareness and decision-making of ego-vehicles with inferences over concepts, rules and axioms. Evaluated ontologies are classified using the dynamic and static abstraction level and placed accordingly in a two-dimensional space. The level of abstraction in each dimension is different in each ontology and ontologies should be chosen according to their application as a more abstract ontology is more lightweight, but could be sufficient for many applications and a more concrete ontology can be needed, but comes with higher computational demand.

[6] outline a method to improve the situational awareness and decision-making of AI-based automated and autonomous vehicles. They propose to use the information of the German Technical Specification on Lane Markings and formulate it into an ontology to obtain more concrete knowledge on lane markings. This ontology expands a base ontology, here the ASAM OpenXOntology, which then can be used to infer for example specific information on traffic area, location and traffic lanes. This method of knowledge extraction can be used on different Technical Specifications and might be used on similar documents in different countries.

3.2.3 Ontology Modelling and Scenario Representation

[7] offer high-level definitions of fundamental terms like *Ego Vehicle*, *Scene*, *Scenario*, and *Driving Mission*. While conceptually sound, the contribution is limited in relevance to SUNRISE, as these concepts are already formally defined within the project and the paper does not propose an ontology for scenario structuring. The relevance of this paper for SUNRISE was considered to be low, as terms like these are already defined in Sunrise, and this paper does not include a more detailed ontology on scenario level, which is the purpose of this deliverable. Furthermore, this paper is more than 10 years old, and more recent papers have been found.

[8] address ontology-based modelling for trajectory prediction, focusing on contextual factors such as road topology, lane markings, and traffic rules. The use of these data is limited by the limited availability of comprehensive representations. The authors have defined an ontology

for describing this contextual information, and used this to create a knowledge graph for the nuScenes dataset.

The ontology defined is made available online (https://tinyurl.com/5t2vv9yu). The ontology defined could be used as a starting point, or for comparison with other ontologies to check for possible gaps. It also lists other existing ontologies.

In [9], an object-oriented framework is presented that allows to instantiate scenarios, scenario categories, and their constituents as objects from classes. By representing scenarios, scenario categories, and their constituents as objects, it is possible to perform operations on them. A GitHub link is provided to an actual implementation of their framework in Python code.

[10] explore an alternative framework for scenario storage and processing using relational databases. Although this paper is more about storing scenarios using a relational database, it may also serve as a template for a way to communicate scenarios. This work also explains how it is different from OpenX standards; according to [10], the OpenX standards do not provide convenient open-source code libraries and toolkits for data conversion and later applications – instead they focus more on the scenario definition level.

As a starting point for an ontology for formalizing scenarios, the ontology creation method presented in [11] may be used. In [10], a master ontology is presented based on what is available in OpenSCENARIO and what they call "corner case taxonomy" (although it is unclear why the scenarios are necessarily "corner cases"). Based on a scenario description, they can create an ontology that is able to describe the given scenario. This generated ontology is then used to create new scenarios. An implementation is available on GitHub."

3.2.4 Ontologies and Knowledge Graph Integration

Machines can accurately process and interpret traffic situations under the requirement that elements and explicit relationships are well-defined [12]. This ability is supported by baseline data models, the data structures that give form to information, and the applied level of abstraction. Knowledge Bases (KB) have emerged as an appropriate mechanism to structure the complexity of traffic information, enabling logical deduction using written rules. KB can be materialized as Knowledge Graphs (KG) exploiting graph theory to become powerful tools [13].

Research efforts have already studied the role of KGs in autonomous driving. [14] pointed out the importance of KGs for perception, scene understanding, and motion planning. Similarly, [15] developed the Global Knowledge Graph, where data from different autonomous driving datasets were merged together to support a variety of applications. [16] proposed the Mobility Knowledge Graph, designed to predict trip destinations in public transportation. KGs have also been successfully employed in other domains like cybersecurity, healthcare, smart cities or financial services.

3.2.5 Domain-Specific Ontologies in ITS

The creation of a KG follows a structured process [17]. In first place, source data is mapped into an ontology, then converting it into Resource Description Framework (RDF) format, and finally, serializing it for publication. Ontologies are machine-readable vocabularies of classes and relationships, defining their meaning in a certain domain. When integrated with KGs, ontologies act as database schemas, providing consistency and interpretability of stored data. In the context of Intelligent Transportation Systems (ITS), ontologies operate enabling seamless data integration from different sources [18]. They have been used to define traffic scenarios, testing the performance of driving functions, and assessing safety measures.

A wide range of ontologies have been developed in the ITS domain, although most of them are specialized in particular aspects such as traffic agents or scenario elements. [19] focused on modelling driver behaviour, while [20] focused on contextual information of road scenes. Other ontologies target different applications, such as risk assessment, or route optimization. The Connected Traffic Data Ontology (CTDO) [21] is designed to model real-time traffic conditions, with emphasis on sensor data like speed and acceleration. This ontology is closely related to others like the Vehicle Signal and Attribute Ontology (VSSO) and the Sensor, Observation, Sample and Actuator (SOSA) ontology [22]. In any case, these ontologies rarely aim to cover a wider spectrum of road and traffic-related concepts. One attempt is the Automotive Global Ontology (AGO) [23], which aligns its base classes with the base elements of the ASAM OpenLABEL standard, covering a large set of objects, contexts, actions and events. Among other available ontologies for similar purposes, the Automotive Urban Traffic Ontology (AUTO) [24] stands out as one the most extensive, as it follows the 6-layer model framework, a structured approach to model scenario data.

3.3 Existing Ontologies

3.3.1 Standards

A range of standards exist that define the structure of different elements of the ODD and of driving scenarios. These standards can be used to form the basis to define a more concrete ontology and are therefore highly relevant for this deliverable.

3.3.1.1 BSI PAS 1883

The BSI PAS 1883:2020 [25] is a specification published by the British Standards Institution that sets out a standardized approach to describing the ODD of an Automated Driving System (ADS). The ODD defines the specific conditions under which an ADS can safely operate, this includes environmental factors such as road types, weather, traffic conditions, and more. By creating a common language and framework for defining these operational boundaries, PAS 1883 supports the safe development, testing, and deployment of automated vehicle technologies.

A central concept in ensuring the safe use of ADS is “informed safety,” which means that the capabilities and limitations of the system are clearly defined and communicated to users. PAS 1883 supports this by providing a taxonomy for defining the ODD of an ADS. The standard is meant for system designers and manufacturers but also for operators, regulators, and end users who need to understand and trust the performance boundaries of the vehicle.

3.3.1.2 ISO 34503

The ISO 34503 [26] standard establishes a structured approach for defining an ODD, the conditions within which an ADS is designed to function safely. It differentiates between ODD and Target Operational Domain (TOD), which represents real-world conditions that may exceed an ADS's capabilities. The standard introduces a taxonomy for ODD attributes, including scenery elements, environmental conditions, and dynamic elements, to ensure clear specification and testing. It is highly influenced by the BSI PAS 1883 standard. The standard focuses on defining attributes for static scenery content, such as the road geometry and environmental conditions like weather (See also the layers of the six-layer model). Only the type of agents and vehicles is defined in the standard and not their behaviour.

3.3.1.3 ISO 34504

ISO 34504 [27] provides guidance on categorizing (test) scenarios used for the verification of ADSs. This standard defines an approach for the categorization of the scenarios by providing tags that carry information about the scenarios. In ISO 34504, a scenario category refers to a set of scenarios that share one or more characteristics. Tags are attached to a scenario as metadata for the purpose of categorizing the scenarios. A given scenario is part of a scenario category, if all tags of the scenario category are also part of the tags that are applicable to the given scenario. Scenario categories do not need to be mutually exclusive. ISO 34504 provides a standardized set of tags.

In some cases, there is a need of having generic scenario categories – and thus a wide variety among the scenarios belonging to the scenario category – and, in other cases, there is a need of having specific scenario categories without much variety among the scenarios in the scenario category. To accommodate this, ISO 34504 structures the tags in hierarchical trees. The different layers in a tree can be regarded as different abstraction levels.

The actual implementation of the tags into a specification is out of scope of the standard. However, a few examples are provided. One option is to provide the tags to a scenario without any structure. In another example, tags that belong to a (dynamic) entity are grouped. For example, the tags "driving forward" and "following lane" are in one group related to the subject vehicle, while the tags "pedestrian", "in front of subject", and "from right" are grouped as they belong to a different dynamic entity. In another example, also the order in which tags appear in the scenario are specified. For example, in case of a cut-in scenario, a vehicle is first not the leading vehicle of the subject vehicle, while later in time, it becomes the leading vehicle once the lane change has been completed.

For the context of SUNRISE, the tags provided in the ISO 34504 standard can be used as meta data of scenarios, such that querying certain scenarios is possible. Relying on a standardized set of tags ensures consistency of the tags and clarity on the usage of the tags.

3.3.2 OpenODD

An ODD shall be valid throughout the whole vehicle's lifetime. The definition of an ODD is part of the safety concept of a vehicle. Depending on the current development step different information needs to be derived from such an overall ODD [28].

ASAM OpenODD focuses on a machine-readable format. The ODD must be represented so it can easily be used within Simulation and other machine processed environments. The content of ASAM OpenODD will be derived from any abstract ODD, providing the information in a usable manner. For the purpose of using this ODD description for simulations and post-processing the format must fulfill the following requirements: searchable, exchangeable, extensible, machine-readable, measurable and verifiable.

Readability is important: Format is readable – but main intent is for computer processing (with structure and grammar). ISO 34503 is intended to be used for human users (structured natural language).

In the scenario-based testing workflow ASAM OpenODD will play a very important role supporting the test description, defining the boundaries of what to test and achieving a good test coverage of the operational design domain and its borders.

The first version of the ASAM OpenODD project has been released in April 2025.

During the development of the project, the following aspects were addressed in particular:

- ATTRIBUTES: Provision of a base set of relevant attributes for the ASAM OpenODD format.
- SPECIFICATION: Development of semantics and syntax for the ASAM OpenODD description language, also enabling the use of different ontologies/taxonomies for the definition of ODDs.
- METRICS: Evaluation of the possibility of measurable metrics and what the ODD needs to be able to represent, so any application can perform analysis on the ODD.
- REPRESENTING UNCERTAINTY: Representation of uncertainty with the goal to enable the ODD format to handle rare events and misuse.

The ASAM OpenODD standardization initiative takes into account and aims to complement the activities of BSI (BSI PAS 1883 provides a taxonomy for ODD) and ISO (ISO 34503 uses the taxonomy to provide a high-level definition format for ODD). All three projects are in close contact to avoid contradictions.

3.3.3 OpenLABEL

ASAM OpenLABEL defines the annotation format and labelling methods for objects and scenarios. It provides a guideline on how the labelling methods and definitions should be used [29].

From working with different customers, a significant fragmentation emerged in the way each individual organization categorizes and describes the objects populating the driving environment. Such categorizations and descriptions are the fundamental building block of any Autonomous Driving System's (ADS) perception stack, since it is through them that an ADS comes to a basic and profound understanding of the status of it's surrounding.

The lack of a common labelling standard in the industry is the root cause of several different issues:

- Hampered Vehicle2Vehicle Interaction: The different descriptions and understandings of surroundings may cause casualties in complex situations involving two or more different ADSs

- Precluded sharing: It is a highly difficult if not impossible task to share data across organizations that adopted different labelling taxonomies and specifications

- Reduced annotation quality: Each individual labelling task requires ad-hoc training and even development of custom software functions that translate into a higher probability of errors and thus a threat to safety

- Deprecation of old labels: Long-term operation of ADS development implies changes in quantity and comprehensiveness of labels to be produced considering the evolution of the driving scenes, new sensors, and scenarios. As a consequence, a flexible descriptive language is required to absorb future extensions and modifications of labels and guarantee backward-compatibility.

JSON Format - The use of a standardized format will help save cost and resources in converting annotated data. ASAM OpenLABEL will be represented in a JSON format and can therefore be easily parsed by tools and applications. It will specify which coordinate systems are used as reference for the label. This already facilitates the conversion a lot.

Extended Labelling Objects - ASAM OpenLABEL will also provide methods to label objects in a scene (one point in time/ frame) as well as across multiple scenes by enhancing the methods to label actions, intentions and relations between objects.

Labelling Different Data Types - The ASAM OpenLABEL format will be capable of managing different types of labelling methods, for different types of data. This includes 2D and 3D bounding boxes, the rotation of 3D bounding boxes, semantic segmentation of images and point clouds. These semantic segmentations can be either instance classes, single/multi-class, partial or full classes.

It is important that the labelling fits into the taxonomy definitions of a user/company. For that reason, the project group intends to provide ASAM OpenLABEL with the ability to import ontologies and taxonomies for the labelling process. The ASAM OpenLABEL project group is closely interacting with the ASAM OpenXOntology project to align ASAM OpenLABEL with the OpenX domain model and to provide requirements for the ASAM OpenXOntology standard. As ASAM OpenLABEL and ASAM OpenXOntology are both currently being developed in parallel, the ASAM OpenLABEL standard will be developed with an external ontology. The experience on using ASAM OpenLABEL with different ontologies can be used to give the user a guideline on how to import their own ontology and use this with ASAM OpenLABEL. It might be possible that the use of foreign ontologies will require a certain standardized ontology format.

In addition to the object and scene data labelling, another half of the ASAM OpenLABEL standard is targeted for the scenario tagging. Its key use case is for indexing, sorting, organizing, querying and searching scenarios effectively, efficiently for scenario database owners, while using common and standardized format (as seen in Figure 4).

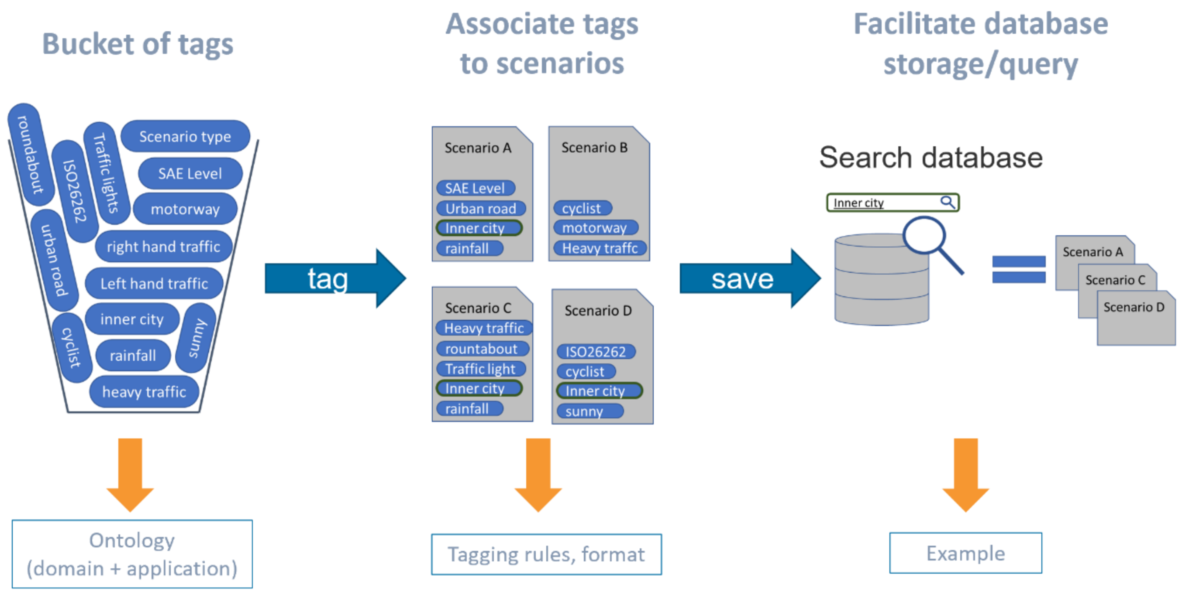

Figure 4: Illustration of the OpenLABEL scenario tagging standard in action.

At the heart of the scenario tagging section are the format and the tagging ontology. The format shares the same umbrella JSON format with the object and scene data labelling mentioned above. The tagging ontology, at the time of the v1.0 development of the standard, aligned with ASAM OpenXOntology, OpenODD, BSI 1883, which were the parallel three standards being developed at the same time. It incorporates an ODD and behaviour based ontology, as seen in Figure 5.

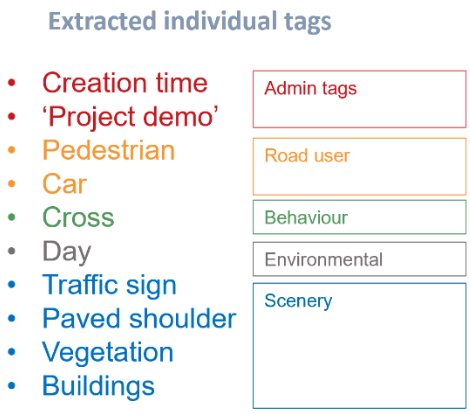

Figure 5: Example tags and their corresponding ODD & behaviour categories, plus the admin tags.

Please note that since then, OpenXOntology was positioned as a concept project for further development, OpenODD stated that it will focus on the format rather than restricting the ontology, and BSI 1883 has been input into the ISO 34503 while itself became a user guide for the ISO 34503. Furthermore, OpenLABEL supports adding ‘admin tags’ which can contain any meta data the user might want to associate with (e.g., intended SAE level for the scenario). It is highly extensible in nature, allowing user to connect any user-side extension to the baseline OpenLABEL ontology. The ontology uses the turtle format from W3C, interchangeable with OWL and other W3C formats, and the tagging file uses JSON format. Regardless, ASAM OpenLABEL v1.0 scenario tagging contains very aligned ODD taxonomy as found in ISO 34503, and a dedicated behaviour taxonomy for tagging the behaviour component of the scenarios, which are still very relevant given the updates in other standards since its release.

3.3.4 OpenXOntology

The ASAM OpenXOntology is a concept document which details a standardisation effort towards forming a harmonised ontology across ASAM OpenX standards, furthermore it includes the ontology format, as well as reasoning rules and ontology reasoning use case demonstrations. The initial input towards this standardisation activity includes OpenODD, OpenLABEL, OpenSCENARIO xml, OpenDRIVE, and Open Simulation Interface. The ASAM OpenXOntology architecture is modular and layered, comprising core, domain, and application ontologies, each building upon the last, as seen in Figure 6. The core ontology defines fundamental, abstract concepts, such as physical objects, activities, and events, based on the HQDM framework.

The domain ontology extends this with traffic-specific concepts like roads, vehicles, and weather. These two layers are linked: for example, a vehicle (domain) is modelled as a type of physical object (core). On top of this, application ontologies, such as for OpenLABEL, reuse these concepts to add task-specific elements, e.g. labelling a vehicle in sensor data. These layers combine into an application integration ontology, enabling harmonised, semantically rich data use across the OpenX standards. The domain ontology in ASAM OpenXOntology organises key traffic domain concepts into structured categories, primarily as a taxonomy. It defines entities such as road elements (roads, lanes, junctions), traffic infrastructure (signs, signals), traffic participants (vehicles, pedestrians), and environmental factors (rain, fog, lighting). Each class represents a shared concept across OpenX standards, promoting consistency. While the current version lacks object properties or axioms, it lays a foundation for linking to the core ontology’s formal structure. Future enhancements aim to enrich this taxonomy with defined relationships and semantic rules, enabling more advanced reasoning and supporting diverse simulation, labelling, and scenario modelling tasks.

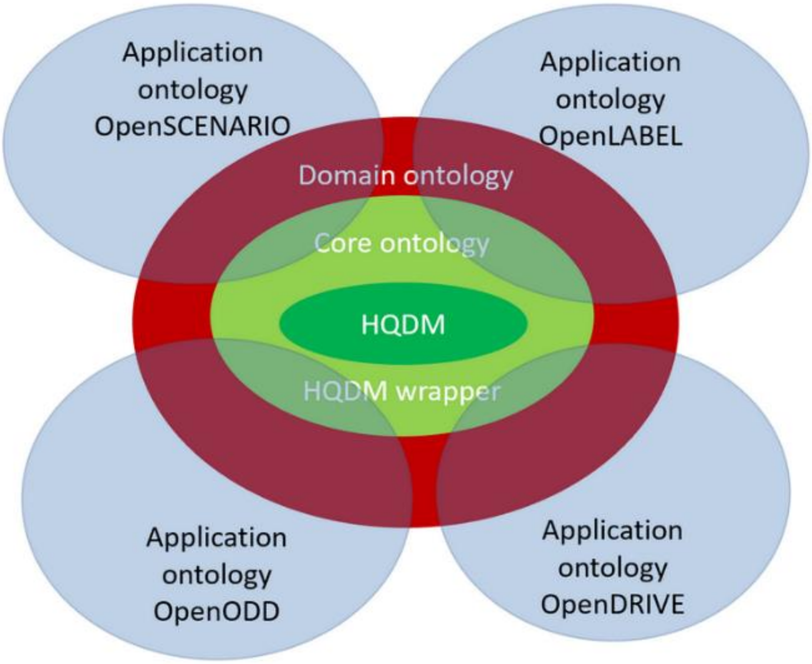

Figure 6: Example visualisation of core, domain, and application ontologies.

ASAM OpenXOntology uses **reasoning** to infer new knowledge from existing data and detect semantic inconsistencies. It leverages **SWRL (Semantic Web Rule Language)** to express logical rules that go beyond what OWL alone can define. These rules follow a structured *if, then* format and enable applications to derive facts or constraints. For instance, a rule might state: *If vehicle A is the subject of a following activity and vehicle B is the object, then vehicle A is behind vehicle B*. Such reasoning supports intelligent data validation and consistency checks. SWRL rules in the current release are limited to the **core ontology**, with domain-level rules to follow. They enable semantic applications to process relationships between individuals (ABox) and assess conformance to logical structures defined in the class model (TBox). This enhances automation in modelling complex traffic scenarios and supports alignment across tools and standards in the ASAM OpenX ecosystem.

3.4 Ontologies in the SUNRISE project

In this section we give insights into how the harmonised ontology for the SUNRISE project has been developed. We start with an overview of requirements collected in the project and how they impact the ontology. Additionally, we discuss how the ontology is structured to address these requirements, how parameters are mapped to the ontology and finally, how it is used within the project. In addition to the harmonised ontology proposal, candidates from existing standardised ontology are summarised and put forward as candidates (section 3.4.3.1) for the SUNRISE Data Framework implementation.

3.4.1 Ontology Requirements Analysis

In deliverable D5.1 [30] requirements have been defined for the Data Framework content for CCAM safety assessment. These include requirements important when defining a harmonised ontology. In the deliverable the requirements were grouped into clusters. The clusters relevant for the ontology include:

- **CL7** Structure of Scenario Representation: Involves requirements related to organising and categorising scenarios data framework.
- **CL9** Scenario Filtering/Searching: Refers to a set of requirements or functionalities related to searching and filtering scenarios.
- **CL11** Scenario Metadata Association: Involves requirements that focus on capturing and associating essential metadata elements with scenarios.
- **CL13** Scenario Tagging/Labelling: Involves requirements related to tagging and labelling scenarios within the Data Framework.

Key individual requirements from these clusters that directly influence ontology definition include:

RE25: Specifies that scenarios can be represented hierarchically or via tags, underlining the need for an ontology that supports both hierarchical classification and flexible tagging mechanisms.

RE30: Requires the Data Framework to enable querying and filtering based on an agreed taxonomy, including meta-information. This necessitates an ontology that defines these taxonomies explicitly and supports semantic querying.

RE31: Demands scenario filtering based on Operational Design Domain (ODD) attributes, which calls for a standardized, ontology-based description of ODD elements to ensure interoperability and semantic clarity.

RE38: Requires the inclusion of scenario metadata, reinforcing the need for an ontology that captures metadata types, properties, and their associations to scenarios.

RE40: Calls for the definition of mandatory and optional parameters using a taxonomy/ontology, indicating that the ontology must distinguish between core and optional scenario attributes.

RE41: Specifies detailed metadata requirements, such as scenario origin, addition date, and ADS activation status, each of which must be ontologically represented.

RE47: Stipulates compatibility with SUNRISE partner databases, implying that the ontology must align with existing schemas or support mappings to external data models.

RE49: Highlights the need to represent both logical and concrete scenarios. The ontology must thus distinguish between scenario abstraction levels and support links between them.

RE56: Requires complete, unambiguous scenario descriptions, emphasizing the ontology's role in defining precise semantics and reducing interpretation gaps.

RE57: Includes support for dynamic objects, map data, weather, and variable road networks, elements that must be defined and interrelated within the ontology.

RE61: Mandates the inclusion of V2X (vehicle-to-everything) content in scenario definitions, necessitating ontological representations of communication entities and interactions.

3.4.2 Ontology proposals within SUNRISE

3.4.2.1 Ontology candidate for scenario semantic representation

Based on the requirement analysis and existing literature and standards a harmonised ontology has been defined within the SUNRISE project. The goal of this taxonomy is to form the basis for a scenario ontology as a candidate to be used in the SUNRISE Data Framework (DF). Here, a unified description of scenario content is needed to ensure communication between the users of the DF and the databases connected to the SUNRISE Federated Layer. Additionally, the ontology is important to align understanding of scenarios by tools within the Harmonized V&V Simulation Framework. Furthermore, it is vital to align the taxonomy and ontology with current standards to ensure acceptance. Furthermore, they should not include idiosyncrasies of specific scenario databases, which for example force specific scenario concepts, but instead form the baseline description of scenario content that is aligned across all the databases connected to the SUNRISE Federated Layer. Therefore, adhering to standards is key for enabling this requirement.

To this end we have relied heavily on existing standards, especially ISO 34503 and ISO 34504, to form the groundwork for the content of the ontology and taxonomy.

The ontology is created using a tree structure. Each element in this structure, apart from the root element, has a parent element. Furthermore, each of the elements can have multiple child elements. This enables us to define broad characterizations at a top level, with more specific elements being defined the lower one goes down in this structure. At the top we differentiate between four top level elements:

- Scenery Elements: The static components of the scenario/ODD definition.
- Environmental Condition: This includes effects like weather, but also V2X communication.
- Dynamic entities: The definition of behaviour of the dynamic entities in a scenario.
- Additional Information: A taxonomy to describe meta-information of the scenario.

The ontology developed for the SUNRISE Data Framework provides a semantically rich and structurally coherent representation of all elements necessary to support comprehensive scenario-based safety assessments. Among its various branches, the Scenery Element sub-branch plays a foundational role. This part of the ontology is responsible for capturing the physical and spatial context in which driving scenarios occur. It defines the type of area being driven in, such as whether it is urban, rural, or highway, as well as more specific geographic regions. This is essential for situating a scenario within a realistic and operationally relevant setting. The ontology also details the makeup of the roads being driven on, including structural aspects such as road geometry, lane types, lane numbers, and surface composition. Such granularity is critical for supporting both logical and concrete scenario representations, as mandated by requirement RE49, and ensures the unambiguous scenario descriptions required in RE56.

Beyond roads themselves, the ontology accounts for surrounding infrastructure and objects including buildings, vegetation, and street furniture such as signage and lighting. Junctions and intersections, including roundabouts and other complex configurations, are defined as specific components that structure how different roads connect and how agents navigate the environment. The ontology does not limit itself to permanent infrastructure; it also incorporates temporary structures, including construction zones and detours. This is crucial for compliance with RE57, which emphasizes the need to model dynamic and changing road environments, and supports RE40 by distinguishing between mandatory and optional scenario elements through ontological categorization.

Complementing the scenery elements are definitions related to dynamic entities, which include objects and agents that move within the scenario and influence its evolution. These entities include vehicles, pedestrians, cyclists, animals, and inanimate obstacles, each captured as specific ontological classes. The ontology further defines the roles that such dynamic entities can assume, such as leading, following, yielding, or having no defined role. This structuring supports nuanced behaviour modelling and interaction analysis between road users, aligning with RE31's requirement for ODD-based search and filtering, and ensuring that both actor behaviour and semantic roles are clearly represented.

Dynamic behaviour is modelled through a taxonomy of actions, categorized by their direction and nature. Lateral actions such as changing lanes, swerving, or turning, longitudinal actions like accelerating, decelerating, or maintaining speed, and mixed actions such as parking manoeuvres are all accounted for in the ontology. These action definitions enable precise modelling of how scenarios evolve over time and how different agents interact within them. Additionally, the initial state of each entity, including its direction, speed, and position relative to others, is captured to ensure that concrete scenarios can be simulated without ambiguity, again supporting RE56.

To ensure that each scenario is traceable, interoperable, and searchable, the ontology also includes a robust metadata framework. Tags such as Scenario_type, Scenario_usage, Scenario_source, and Intended_execution_platform enable detailed classification and support cross-platform data sharing, directly addressing RE38 and RE47. The Abstraction_level tag helps distinguish between high-level logical scenarios and low-level instantiations, which is essential for RE49. Administrative tags define the origin, the creation or ingestion date, and whether the ADS (Automated Driving System) was active at the time, fulfilling the data traceability obligations specified in RE41. These metadata elements, when combined, facilitate advanced filtering and searching as required by RE30 and RE31, and make the ontology an effective semantic foundation for the entire scenario Data Framework.

In addition to static and behavioural elements, the ontology comprehensively models environmental and contextual conditions. Visibility is described in terms of whether objects are fully in view, partially obstructed, or completely blocked, which supports sensor-based assessments and visibility modelling required by RE56. Illumination, including both artificial and natural light sources, is included alongside cloudiness and sun position to give a realistic picture of lighting conditions. Weather is modelled in detailed terms, including rainfall, snowfall, wind, and ambient air temperature, while particulates are categorized by type, size, and intensity. These detailed environmental descriptors support realistic scenario modelling and meet the completeness requirements of RE57.

The ontology also includes connectivity and communication components to fulfil RE61, which calls for scenario definitions that include V2X content. It models communication capabilities including whether vehicles or infrastructure elements are connected and defines positioning technologies such as RTK correction. These features allow the ontology to represent modern, connected driving environments, which are essential for future-focused CCAM safety assessments.

To represent environmental concepts in the ontology, we use several key ontological elements. Each concept is modelled as a class, and classes may be related to one another through various relationships. A class can have subclasses, which represent more specific variations of the parent class. For example, a truck might be a subclass of vehicle, since the concept of a truck is a more specific form of the broader concept of a vehicle.

In addition to hierarchical relationships, classes can have attributes that describe their characteristics. For instance, temperature is not a subclass of weather, but rather an attribute of it, weather has a temperature.

Beyond classes and attributes, the ontology also includes instances (or individuals), which are concrete realizations of classes. For example, light breeze is an instance of the class wind speed, characterized by a specific range of values.

To formally model attributes in the ontology, we introduce a root class called Attribute, which has two subclasses: Quantity and Type. These help distinguish between ordinal (measurable) and nominal (categorical) values. A quantity like temperature, for example, can be represented either as a single scalar value or a range of values.

Let's examine how this structure applies to a specific case: the concept of no rain. In the ontology, no rain is defined as an instance of the class *Rainfall_intensity_scalar*, with a scalar value of zero and the unit mm/h. The class Rainfall has *Rainfall_intensity* as an attribute, and *Rainfall_intensity_scalar* is a subclass of *Rainfall_intensity*, representing one way to quantify this attribute.

Using this structure, we can define our own rainfall event as an instance of the Rainfall class. We assign it the intensity no rain, linking it to the instance we previously defined. Furthermore, we can also associate a temperature with this event, since this attribute is inherited from Weather, the parent class of Rainfall.

This modelling structure is illustrated in Figure 7, which shows the no rain instance in the context of its parent classes and attributes.

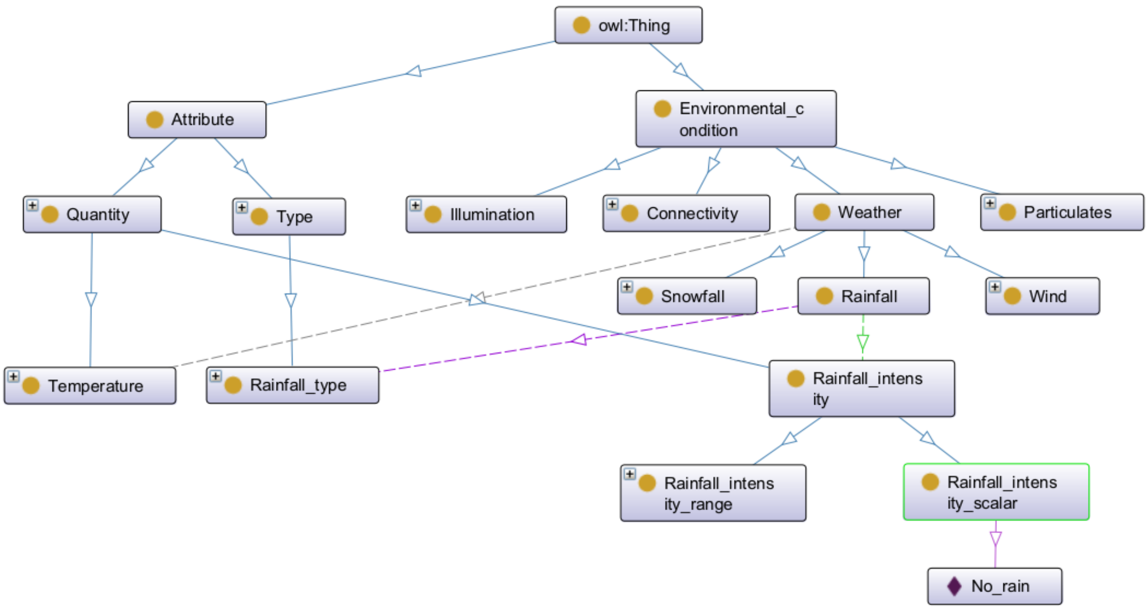

Figure 7: No_rain element ontology sub-structure.

Together, these ontological components form a harmonised framework that meets the structural and functional requirements outlined in Deliverable D5.1. The ontology is designed to capture the complexity and variability of real-world driving environments and also to support scenario tagging, metadata association, hierarchical structuring, and advanced search and filtering capabilities. As such, it enables a Data Framework that is robust, extensible, and fully aligned with the needs of scenario-based safety validation and automated driving system development.

The ontology is provided in the Resource Description Framework (RDF) format, which is a format to describe graph-based data, well suited to the ontology description. The current version of the ontology can be found under: https://github.com/ika-rwth-aachen/sunrise_ontology [31].

3.4.3 Ontology and OpenLABEL

To be able to use the ontology for the purpose of scenario tagging and querying, it must be integrated into a suitable format. This format needs to allow for the specification of tags related to the concepts present in the ontology. To address this, we propose to apply the ontology to the OpenLABEL JSON format, utilizing the tags structure.

First, OpenLABEL allows for the definition of different ontologies in the similarly named section of the standard. The SUNRISE ontology must be specified here.

OpenLABEL specifies different groups for tags, which are administrative, odd, behaviour and custom. These groups must be used to group tags of the corresponding concepts in the SUNRISE ontology. Within these groups, tags are included. For each concept from the ontology that is included, a separate tag must be defined. Additionally, a tag must be defined for each attribute of a concept that is included as well. A tag cannot be defined for the parent concepts. Each tag must include a unique name, a type corresponding to the concept of the ontology and an ontology uid that defines the concept. There are four different ways of defining tags. They differ in the way the value (val) field within the OpenLABEL specification for tags is used. They are:

1. Definition by naming instance (For concepts without attributes)
2. Definition by omission of the value field (For named instances)
3. Definition by reference
4. Definition by explicitly providing the data properties

The first option is only applicable if the concept has no associated attributes (either other concepts or data properties). In this case, the instance of the concept is well-defined by providing a unique name. The name is provided in the value field of the text element in the value field of the tag. The name refers to the concept and is not the name of the tag. The corresponding name field to the value field must contain the type of the concept. This is shown in Figure 8.

```
{
  "name": "my_rainfall_type",
  "type": "Rainfall_type",
  "ont_uid": "SUNRISE#Rainfall_type",
  "val": {
    "text": [
      {
        "name": "Rainfall_type",
        "val": "Dynamic"
      }
    ]
  }
}
```

Figure 8: Definition by naming instance.

If a named instance is already defined in the ontology and used, then the instance is well-defined without the value field and it must be omitted. This is the second case, and an example is shown in Figure 9. In this case, the uid must directly reference the named instance.

```json
{
  "name": "my_rainfall_intensity",
  "type": "Rainfall_intensity",
  "ont_uid": "SUNRISE#No_rain",
}
```

Figure 9: Example of a named instance.

The third case is applicable if the concept has attributes that are itself instances of concepts. In this case, these instances of the attributes must be defined in separate tags. They can then be referenced in the original instance. A @ symbol is to be used before the name of the tag to be referenced. The name of the nested text element within the value field must include the type of the referenced tag. This is shown in Figure 10.

```json
{
  "name": "rainfall_event",
  "type": "rainfall",
  "ont_uid": "SUNRISE#Rainfall",
  "val": {
    "text": [
      {
        "name": "Rainfall_intensity",
        "val": "@my_rainfall_intensity"
      },
      {
        "name": "Rainfall_type",
        "val": "@my_rainfall_type"
      },
      {
        "name": "Temperature",
        "val": "@my_temperature"
      }
    ]
  }
}
```

Figure 10: Example of definition by reference.

The final way is the definition of explicit data properties. This way must be used if the concept has data properties and a named instance is not used. The name of the nested text element within the value field must include the name of the data property as defined in the ontology. The corresponding value is included in the val field of the text element. The value is encoded into a string, irrespective of the actual data type. This is shown in Figure 11:

```
{
  "name": "my_temperature",
  "type": "Temperature_scalar",
  "ont_uid": "SUNRISE#Temperature_scalar",
  "val": {
    "text": [
      {
        "name": "temperatureValue",
        "val": "20"
      },
      {
        "name": "unit",
        "val": "Celsius"
      }
    ]
  }
}
```


Figure 11: Example of providing the data properties

The example in its entirety and embedded within OpenLABEL is provided in the Annex 1.

The instances of ontology concepts previously discussed can also be used to establish rules for querying scenario databases. In the following, we will present an example of a minimal implementation of such querying fully compliant with the OpenLABEL standard. The needed operators are as follows:

1. INCLUDE: The scenario database shall return scenarios that include the referenced concept instance.
2. EXCLUDE: The scenario database shall return scenarios that do not include the referenced concept instance.
3. INCLUDE IF: The scenario database shall return scenarios that include the referenced concept instance if a condition is met.
4. EXCLUDE IF: The scenario database shall return scenarios that do not include the referenced concept instance.
5. EXIST: Condition that is met if a scenario in the database contains a certain concept instance.
6. AND, OR, NOT: Logical operators to create more complex conditions.

Operators can be represented as tags within the OpenLABEL JSON format. An unbounded number of query rules and conditions can be created. It should be noted that logical contradiction can be formed using these operators. The query process shall include a mechanism to short circuit these logical contradictions. In the following we present examples of how these rules, conditions and operators can be used within the JSON format. For each rule and condition, a separate tag shall be created. Each tag shall be given a unique name. This name is used to reference the tag within other elements. The type of the tag, either query rule or condition, shall be provided. The operators are included in the value field of the tag.

Only text elements are allowed here. The instance name the operator refers to is given in the text element. This instance shall be defined in a separate tag, as discussed previously. If the inclusion or exclusion is conditional, an additional text field with an if operator shall be provided. An example is provided in Figure 12.

```
{
  "name": "include_rainfall",
  "type": "Query_rule",
  "val": {
    "text": [
      { "name": "INCLUDE", "val": "@rainfall_event" }
    ]
  }
},
{
  "name": "exclude_rainfall",
  "type": "Query_rule",
  "val": {
    "text": [
      { "name": "EXCLUDE", "val": "@rainfall_event" }
    ]
  }
},
{
  "name": "include_if_condition",
  "type": "Query_rule",
  "val": {
    "text": [
      { "name": "INCLUDE", "val": "@rainfall_event" },
      { "name": "IF", "val": "@rainfall_exists" }
    ]
  }
},
```

Figure 12: Example of query rules.

Query conditions can be defined in a similar way. An example is provided in Figure 13. When performing a logical operation on two conditions, two text elements shall be provided in the tag, for each of the conditions respectively. The first element shall be named condition, while the second elements shall be named after the logical operator.

```
{
  "name": "rainfall_exists",
  "type": "Query_condition",
  "val": {
    "text": [
      { "name": "EXIST", "val": "@rainfall_event" }
    ]
  }
},
{
  "name": "rainfall_and_temp",
  "type": "Query_condition",
  "val": {
    "text": [
      { "name": "CONDITION", "val": "@rainfall_exists" },
      { "name": "AND", "val": "@temperature_exists" }
    ]
  }
},
{
  "name": "not_rainfall",
  "type": "Query_condition",
  "val": {
    "text": [
      { "name": "NOT", "val": "@rainfall_exists" }
    ]
  }
}
```

Figure 13: Example of query conditions.

3.4.3.1 Ontology candidate for scenario tagging

ASAM OpenLABEL provides a unified framework for labelling and tagging scenarios. It supports both object-level annotations (such as bounding boxes and classifications, incorporating spatial and temporal relations) and attribute-level tagging (i.e., ODD and behaviour keywords). This dual capability makes OpenLABEL well-suited for training, validation, as well as the structured cataloguing/indexing/organisation/query of scenario database. The latter one aligns closely with the purpose the SUNRISE Data Framework is set out to achieve.

At a high level, from a scenario tagging perspective, the ASAM OpenLABEL ontology contains three layers: admin tags, ODD and behaviour.

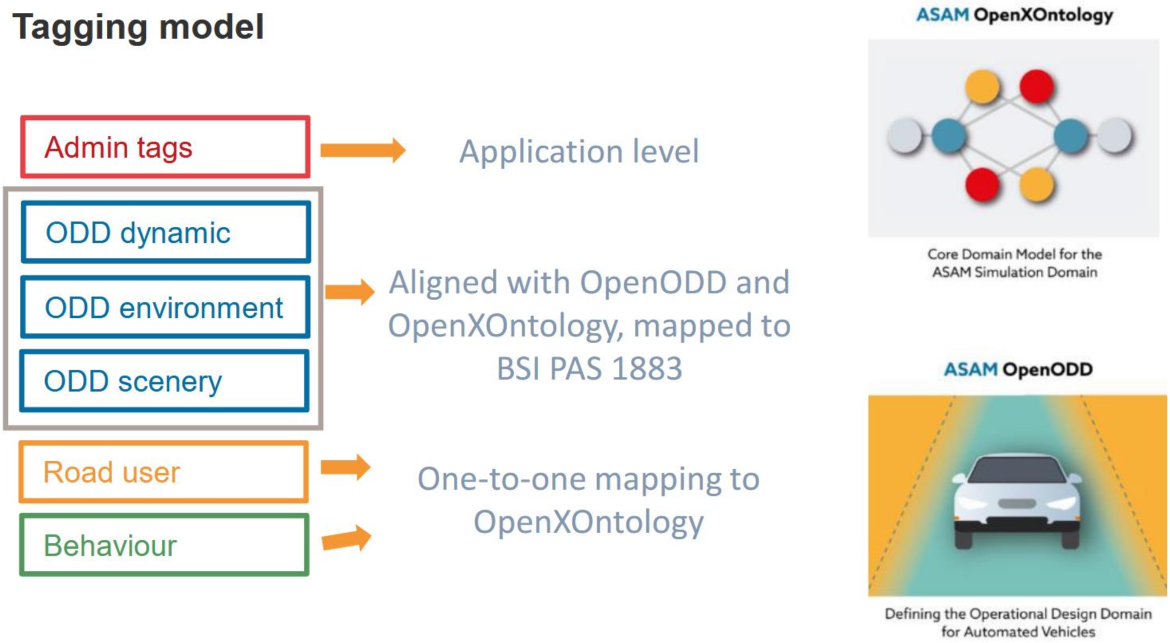

Figure 14: ASAM OpenLABEL tagging ontology composition and alignment [29]

At the scenario-independent layer, OpenLABEL tagging ontology supports a metadata structure (admin tags) that ensures each labelled scenario or dataset is traceable, version-controlled, and machine-readable. The admin tags may include: Label provenance: Information about who annotated the data, when, and using what tools. Scenario identifiers: Unique IDs for scenario instances and their components. Ontology references: Pointers to the version of the tagging ontology or external references used. Annotation status: Flags to indicate whether data is complete, reviewed, partially annotated, or uncertain. This set of admin tags is extensible and user may add any further required admin tags for efficient scenario tagging and query.

For the ODD aspect tagging of the scenarios, OpenLABEL leverages PAS 1883:2020 by adopting its structured approach to define the ODD taxonomy. Furthermore, it also aligns with the then on-going development ASAM OpenXOntology at the domain attributes level. This integration ensures that the ODD attributes found within a scenario is consistently indexed with a standardised vocabulary. As a result, the tagging enables a clear and consistent interpretation of where and how a SUT is expected to operate in a scenario, supports traceability regarding the system's operational boundaries, and allows for comprehensive coverage analysis of the dataset, such as identifying underrepresented environmental or road conditions. These structured tags are formally embedded in OpenLABEL annotation files using RDF-based ontologies, ensuring compatibility and semantic precision. *Please note, during the development of the ASAM OpenLABEL, the equivalent ODD taxonomy standard within the ISO standardisation body was still under development, which was later released post the release of ASAM OpenLABEL V1.0. ISO34503 and BSI PAS 1883 share many common taxonomy characteristics, but there are minor differences which are currently being prepared to be incorporated in the next ASAM OpenLABEL release. Also, BSI PAS 1883:2025* *(the updated version of 1883) is transformed into a user guide, the taxonomy content is superseeded by the ISO34503 content.*

OpenLABEL also includes support for tagging the behaviours of traffic participants, including both the SUT and other scenario agents. This enables the tagging of the behaviour aspect of a scenario. Behavioural tags may capture vehicle manoeuvres such as cut-ins, cut-outs, emergency braking, or walking/running pedestrians. These behaviour-based tags allow scenario datasets to express not only which entities are present in the scene, but also the types of behaviours they exhibits based on the scenarios, making them especially valuable for database querying/indexing use cases.

The OpenLABEL tagging framework is built upon a machine-readable ontology defined using RDF/Turtle. This structure enables explicit semantic relationships between tags, improving consistency and interpretability across systems. The ontology is designed to be extensible, allowing users to define custom tags as needed. For example, organisations may introduce region-specific road elements, or tailor behaviour categories to their testing focus. A detailed workflow has been previous published in [32]. The ASAM OpenLABEL V1.0.0 tagging ontology turtle (.ttl) file is hosted by ASAM and can be accessed under [33], as detailed in the previous section it utilises a json schema, which can be found [34].

Figure 15 demonstrates an example scenario tagging process using the ASAM OpenLABEL V1.0.0 tagging ontology, and the OpenLABEL Schema. For convivence, the scenario used in this example is represented as an image, it contains various ODD and behaviour attributes such as raining weather, car, minor road etc. Within the "OpenLABEL tagging instance' section, their representation using the schema is illustrated, as seen they are all part of the ontology_uid 0, which is the OpenLabel tagging ontology defined in the 'ontologies" section. Furthermore, the attribute values (if applicable as defined in the ontology) can also be associated. They include, but not limited to, discontinued range band (e.g., [3.4, 3.7], [3.9, 4.1] for two ranges of road width); descrete values (e.g., [2, 3] for number of lanes of the roads within the scenario); open-ended value range (e.g., min = 1.2, for rainfall intensity). Please note, the attribute types are all pre-defined within the tagging ontology file to prevent user from inputting wrong data type during the tagging process.

To sum up, this sub section references existing ontology candidates as they are, without modifications. This will prevent incoherent tags usage between DF and SCDBs, using standardised tags is the rationale behind SUNRISE, which will maximise the chances for external SCDBs who are already using international standards to join the SUNRISE DF.

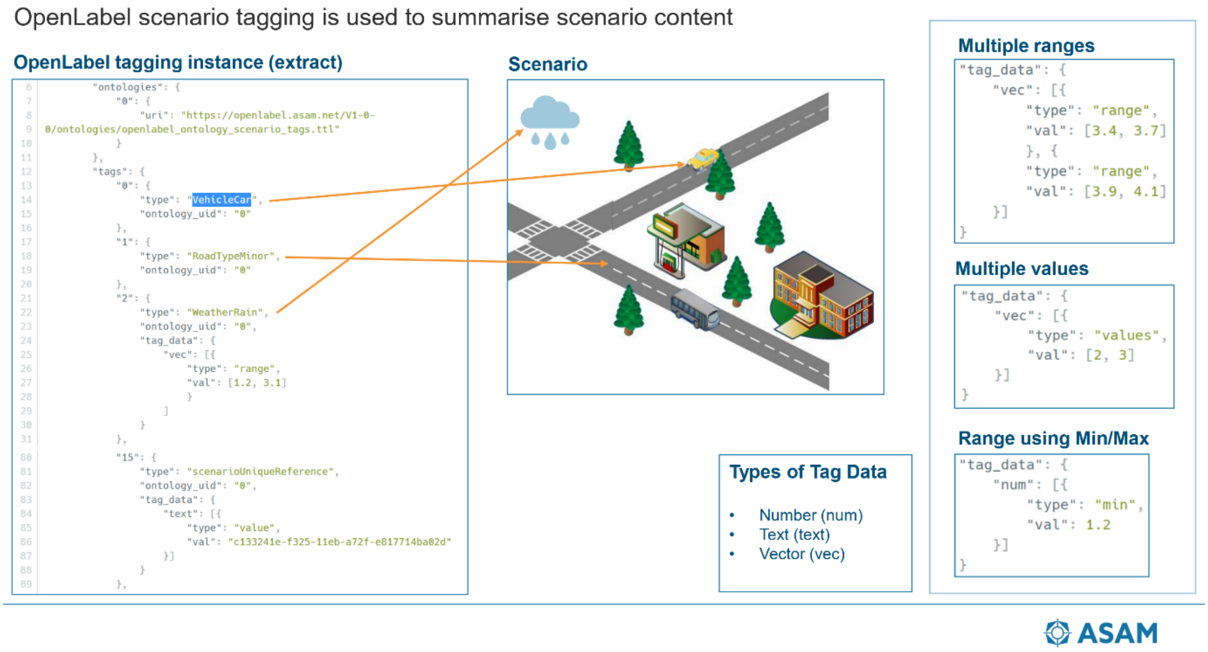

Figure 15: ASAM OpenLABEL tagging process on an example scenario [29]

3.4.4 Parameters and Ontology

To effectively match parameter definitions to a domain ontology, one must begin by understanding the precise role and meaning of each parameter within the context of the system or scenario it supports. Parameters are often defined within technical frameworks, simulation tools, or data schemas and may include a range of inputs such as vehicle speed, road type, weather condition, or scenario source. These parameters need to be semantically aligned with the concepts defined in the ontology in order to ensure consistency and interoperability across tools and platforms.

The first step in this process involves a thorough interpretation of each parameters intent. This includes identifying whether the parameter represents a physical quantity, a categorical attribute, or metadata. For example, a parameter such as “vehicle speed” clearly denotes a measurable dynamic property of a moving entity, while a parameter like “road type” serves a classificatory function, organizing scenarios based on the type of infrastructure involved. Understanding the nature of each parameter is essential for determining where it fits within the conceptual model provided by the ontology.

For these different classes of parameters, different concepts are included in the ontology to describe them. In the previous section we have already shown how they are structured and can be used.

Once the meaning of each parameter is established, the next step is to locate the corresponding concept within the ontology. This could be an ontological class, such as Vehicle, Weather Condition, or Road Structure, or it could be a specific property, such as has speed, has surface type, or occurs under weather condition. In cases where a parameter refers to a relationship between entities, such as a vehicle driving on a specific road or a scenario occurring during a particular time, the appropriate match might be an object property representing that relationship. For parameters that carry quantitative or descriptive values, such as speed, temperature, or date, the match would typically be to a data property defined within the ontology.

For categorical instances, the ontology may already define controlled vocabularies or enumerated classes which represent parameter values. For instance, a parameter with a value like "highway" can be matched to a defined instance within a Road Type class. This not only ensures semantic consistency but also supports the advanced querying and filtering capabilities specified by requirements such as RE30 and RE31. In doing so, the ontology becomes a mechanism for structuring vocabularies as well, reducing ambiguity and aligning them with the domain knowledge.

It is also important to consider the parameters position within the overall structure of the ontology. This involves tracing where the concept belongs in the hierarchy and understanding its relation to other entities. Parameters are generally divided between quantities, types, states and actions. All parameters are a specialization of the Attributes concept in the ontology. For quantities they are further subdivided into scalar values and ranges. Each parameter is then mapped with an object relation to other classes within the ontology. For example, a parameter describing temperature is associated with the concept of weather. Each individual of the weather class can therefore contain a parameter to specify this weathers temperature. This contextual alignment ensures that the parameter contributes meaningfully to the overall scenario representation and maintains logical consistency within the ontology's schema.

Beyond semantic alignment, practical aspects such as units, data types, and expected formats must also be addressed. The ontology can be used to specify these constraints, and the parameter definitions should adhere to them. This can be done by defining data properties as elements of ontological concepts. This is particularly relevant in simulation environments where mismatched units or inconsistent data types can lead to errors or misinterpretation. By enforcing alignment at this level, the ontology also serves as a specification for system integration and scenario completeness.

To formalize and the mapping between parameters and ontology elements, a structured representation was included in the ontology, so that for every parameter values and units can be well-defined.

3.4.5 Application in the Safety Assurance Framework

The ontologies described in this document, are needed to enable standardization of the query to search for scenarios in various scenario databases. This standardization enables the generation of tools to search through various scenario databases through a common front end, which is one of the main objectives of SUNRISE and further developed in work package 6. More specifically, this standardization enables the creation of an Automated Query Criteria Generation (AQCG) tool, as described in more detail in SUNRISE Deliverable D6.1 [35]. The AQCG tool aims at automatic generation of query criteria for searching of scenarios, based on inputs such as the use case ODD, test requirements, dynamic driving task (DDT), etc. The high-level concept of the proposed tool is shown in Figure 16.

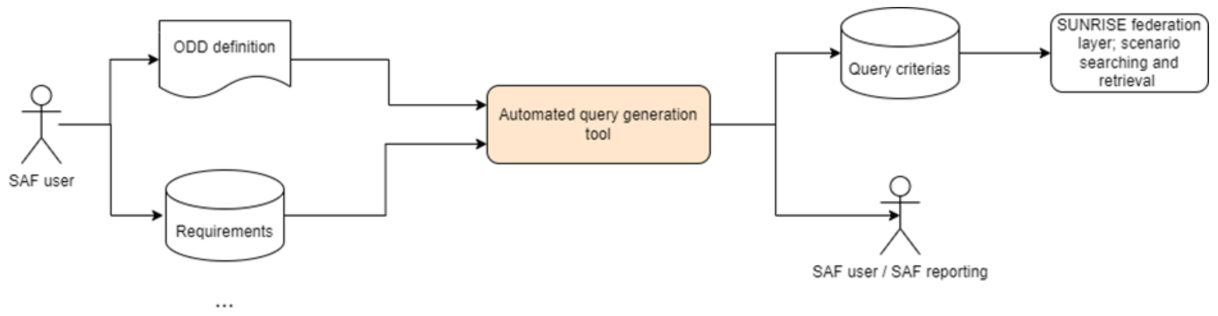

Figure 16: A high-level conceptual view of the Automated Query Criteria Generation tool.

The AQCG tool requires inputs from the user including the ODD definition and requirements. The data formats and ontologies for these inputs must be aligned with the rest of the SAF, such that there is a seamless data exchange and interfacing with different SAF modules. Likewise, the tool will output a query criterion to be used by the federation layer for searching of scenarios. Thus, the output format and ontology must match the expected query format by the federation layer.

Data formats and ontology for the **inputs**:

ODD definition: The ODD definition is expected in the ISO 34503 ODD definition language and ontology. Although the ODD definition language in this standard is natural language based, it has a simple grammar and only a few types of statements. Therefore, it can be parsed in a straightforward manner.

Requirements: For requirements, it was agreed that a JSON file format would be used, with the ODD elements covered in the requirements to be described with the ISO 34503 taxonomy and the dynamic behaviour elements to be described using the ASAM OpenLABEL standard. As the OpenLABEL standard allows for adding custom tags, these can be added if the requirements relate to behaviour which cannot be described using the existing tags.

Data formats and ontology for the **outputs**:

Query criteria: The ASAM OpenLABEL standard was chosen for the query criteria output. A JSON file format is agreed. One important note is that instead of BSI 1883 taxonomy which OpenLABEL currently supports, the ISO 34503 and ISO 34504 taxonomies will be used for the query criteria. This is motivated due to the planned revisions of both OpenLABEL and BSI 1883 to be more in line with the ISO standards. Thus, with the ISO standard taxonomies as the basis for the ontology for the query criteria, the tool is made more future-proof.

3.4.6 Exposure Metrics

In the development of a robust data framework and scenario database (SCDB) for scenario-based assessment, exposure metrics play a central role in ensuring that the collected and organized scenarios reflect the real-world conditions and distributions that connected, cooperative, and automated mobility (CCAM) systems will encounter.

An exposure metric refers to a quantitative measure of how frequently or how extensively a specific condition, event, or scenario type occurs in real-world driving or operating environments. This includes, for example, the proportion of driving time under certain weather conditions (like rain or fog), the share of kilometres driven on different road types (urban, rural, highway), the frequency of specific traffic interactions (such as lane changes, cut-ins, or merges), or the encounters with vulnerable road users (like pedestrians and cyclists).

These metrics are crucial because they help prioritize the importance and relevance of scenarios within the SCDB. Without considering exposure, a database might overemphasize rare or exotic edge cases while underrepresenting the everyday situations that CCAM systems encounter most frequently. Properly applied exposure metrics ensure that scenario-based assessments are balanced, realistic, and representative of ODDs.

To structure and formalize the application of exposure metrics, it is highly beneficial to tie them to elements within the SUNRISE ontology. The ontology provides a structured, formalized description of the key concepts, entities, relationships, and properties relevant to the system under study, such as weather conditions, road types, road user categories, manoeuvres, traffic interactions, and environmental factors. Each scenario in the SCDB can be annotated using terms from this ontology, ensuring semantic consistency and machine-readability across the entire SAF.

Some exposure metrics to consider are:

Table 1: Exposure Metrics

Metric Name	Description	Example
FrequencyInSCDB	Fraction or count of how often this element appears in the current scenario database (SCDB).	150 rainy scenarios out of 1000 total → 15%
FrequencyPerDrivenKm	Number of times this element occurs per 1000 km of recorded driving.	8 lane changes per 1000 km
FrequencyPerDrivenHour	Number of times this element occurs per hour of driving time.	3 cut-in events per hour
ExposureProbability	Probability that, at any given moment, the system is exposed to this element.	Night driving: ~18% chance at any moment

EncounterRatePerKm	Rate of encountering a specific actor or object per km (mainly for road users, obstacles).	5 pedestrian encounters per km in cities
EncounterRatePerHour	Rate of encountering a specific actor or object per hour.	2 cyclists encountered per hour
AverageDuration	Average time or distance during which this element is active when present.	Rain events last on average 20 minutes

These exposure metrics are directly included into the ontology. They inherit from a main Exposure Metric element. They are further connected to all other classes with object properties. This allows the definition of class instances within the ontology that directly follow certain exposure characteristics. Additionally, a coverage analysis can be performed using the ontology with exposure metrics, to identify underrepresented elements.

3.4.7 Future Proofing the Ontology

Ensuring the long-term applicability and adaptability of the ontology is critical for maintaining its value as the SCDB and associated data frameworks evolve. The ontology must not only align with current best practices but also remain robust and extensible in the face of emerging standards, technologies, and scenario types. To achieve this, several key strategies are applied.

The ontology is designed using a taxonomy grounded in widely accepted standards, such as ISO 34503 (Scenario Definitions) and related ISO and BSI norms. This ensures compatibility with ongoing international activities and facilitates integration with other tools, datasets, and regulatory frameworks.

The ontology is structured in a modular way, allowing new classes, properties, and relationships to be added without disrupting the existing framework. For example, if new vehicle types (e.g., automated delivery robots), road user categories (e.g., micromobility devices), or environmental factors (e.g., dynamic infrastructure elements) emerge, they can be integrated as additional modules or subclasses.

Attributes (properties, annotations, metrics) in the ontology are designed with flexibility in mind, using generalized property patterns that can accommodate new content types or descriptors.

As the SCDB expands to include new kinds of data, the ontology can be adapted by defining new elements or extending existing hierarchies. For example, if a new scenario standard defines a novel category of road environment or traffic interaction, it can be represented by adding new subclasses under existing high-level classes.

4 DATA FORMATS

Different data formats play a crucial role to enable the transfer of content between different scenario databases. It is important to ensure that the description of driving scenarios and scenario content is standardized. Furthermore, thought has to be put into how different data formats are allowing for the storage of this content in databases and how the content can be transmitted between database and user, or within the toolchain of a safety assurance process.

4.1 Existing Formats

This section gives an overview over the most important formats that have been considered for use in the SUNRISE project.

4.1.1 OpenSCENARIO

ASAM OpenSCENARIO defines a standard for describing the dynamic elements of driving scenarios, including the behaviours of vehicles, pedestrians, and other entities within a simulation. This standard is especially valuable for testing and validating automated driving functions in a controlled, repeatable, and structured way. The standard is divided into two main branches: OpenSCENARIO XML and OpenSCENARIO DSL, each tailored to different modelling needs and development preferences.

4.1.1.1 OpenSCENARIO XML

OpenSCENARIO XML is the continuation of earlier versions of the standard and is based on XML (eXtensible Markup Language), a flexible, text-based format for storing and transporting structured data. XML uses custom tags to define elements and their hierarchical relationships, making it both human-readable and machine-readable. Its widespread adoption in system integration makes it ideal for scenarios where interoperability is key.

An OpenSCENARIO XML file is organized to define driving behaviour in a modular, time-sequenced manner. At its core is the <*OpenSCENARIO*> root element, which encapsulates the entire scenario description. Within it, the <*FileHeader*> contains metadata such as author, creation date, and version information, aiding traceability and documentation.

Optional <*CatalogLocations*> allow referencing external libraries of predefined components like vehicles or controllers, promoting reuse and consistency across scenarios.

The <*RoadNetwork*> section typically links to an OpenDRIVE file, which defines the physical layout and topology of the road environment. This provides the necessary context in which the scenario takes place.

Entities, such as vehicles, pedestrians, and other participants, are defined under the <*Entities*> section. Each includes details like the model type and controller configuration, ensuring accurate simulation of behaviour.

At the heart of scenario logic is the <*Storyboard*>, which orchestrates how events unfold during the simulation. It begins with an <*Init*> section that sets the initial conditions, including entity positions, speeds, and orientations.

The <*Story*> element introduces a hierarchical structure to model temporal behaviour. It contains one or more Acts, each activated by specific triggers. Within Acts, ManeuverGroups assign actions to entities, organizing them into Maneuvers, which in turn are composed of Events. Each event executes an Action, such as accelerating, changing lanes, or stopping. This layered approach allows for precise and detailed modeling of dynamic interactions.

Finally, the <*Storyboard*> concludes with a <*StopTrigger*>, which defines when the scenario ends, based on elapsed time, distances, or completion of specified behaviours.

4.1.1.2 OpenSCENARIO DSL

OpenSCENARIO DSL, introduced with version 2.0, takes a more modern and expressive approach. Instead of XML, it uses a domain-specific language (DSL) designed to be human-readable and intuitive, while also supporting complex, programmable behaviours. The DSL brings a scripting-style syntax that introduces high-level constructs like actors, actions, conditions, and sequences, along with programming features such as modularity, parameterization, and control flow.

A typical DSL script begins by defining scenario parameters, for example, vehicle types, initial conditions, and environmental variables. These parameters are reusable and easily configurable, making the same scenario adaptable to different contexts or variations.

Entities are then instantiated and initialized, often with precise locations mapped to a road network via OpenDRIVE. The scenario logic unfolds through stories, episodes, and actions, narrative constructs that clearly describe temporal behaviour. For example, one vehicle might be directed to follow another, accelerate to a target speed, or yield at an intersection based on defined triggers.

One of the key strengths of the DSL is its ability to handle conditional logic and event-driven behaviour. Developers can include if-else conditions, wait commands, and runtime constraints, enabling highly responsive, adaptive scenarios. Features like randomization and sampling allow for variability in testing, supporting broader coverage and robustness analysis.

Unlike the more rigid XML format, the DSL is built for readability, collaboration, and scalability. It integrates well into modern development workflows and supports advanced features such as inheritance, composition, and constraint checking, making it ideal for large-scale scenario development and automated testing pipelines.

4.1.2 OpenDRIVE

OpenDRIVE is an open standard developed by ASAM for the detailed description of road networks, primarily used in simulation environments for autonomous driving, driver assistance systems, and traffic modelling. It provides a consistent and machine-readable format for defining the geometry, topology, and semantics of road infrastructure, ensuring that simulated environments closely mirror real-world road conditions.

At its core, OpenDRIVE describes the physical layout of roads through a structured XML format. The standard enables the definition of complex road geometries, including curves, elevations, lane widths, junctions, road markings, and traffic signals. These elements are essential for creating realistic simulation environments in which automated driving systems can be accurately tested and validated.

The road network in OpenDRIVE is composed of interconnected road segments, each defined by a unique identifier. Every road is described in terms of its reference line geometry, which forms the backbone of the road and is defined using primitives such as lines, arcs, and spirals. The standard also allows for the specification of elevation profiles and superelevation, ensuring a fully three-dimensional representation of the road surface.

Each road segment contains one or more lane sections, which further subdivide the road into individual lanes. Lanes are classified by type, such as driving, shoulder, sidewalk, or bike lane, and are described with their own width profiles, lane markings, and connectivity information. This fine-grained level of detail is crucial for the accurate modelling of vehicle dynamics and lane-level behaviour.

OpenDRIVE also supports the definition of junctions, where two or more roads connect. Junctions include detailed information about the paths vehicles can take through intersections, including allowed turn movements and conflict points. This enables precise modelling of complex traffic interactions in intersections, roundabouts, and highway merges.

Additionally, OpenDRIVE integrates road objects and signals, such as traffic signs, lights, poles, and barriers. These can be positioned along roads or lanes with high precision, allowing for realistic simulation of traffic regulations and roadside infrastructure.

One of the strengths of OpenDRIVE is its deterministic and geometry-focused structure, which allows it to serve as a reliable digital twin of real-world road networks. It is often used in combination with other ASAM standards, most notably OpenSCENARIO, which overlays dynamic behaviour on top of the static road environment defined by OpenDRIVE.

4.1.3 OpenLABEL

ASAM OpenLABEL defines a standard format for multi-sensor data labelling and scenario tagging. It provides a structured way to define annotations, data types, and conventions (formats, ontologies), while also specifying the serialization of the content into JSON files.

The JSON schema of OpenLABEL [34] serves both as a data model and a file format definition. A JSON schema describes the expected structure of JSON files, including required fields, data types, constraints and attributes. By defining a JSON schema, OpenLABEL aims to define the basic data types while at the same time being flexible enough to let the user define custom structures.

Also, the JSON schema enforces validation rules to ensure correctness and consistency across annotation files. Usable as a formal documentation, the schema can be used with software packages to read and validate the structure of JSON files.

The OpenLABEL JSON structure consists of various metadata groups which are used to describe objects, actions, events, contexts, relations, frames, tags, ontologies, streams and coordinate systems. Objects, for instance, represent physical entities in the real world and their manifestations under different sensor views. Unique identifiers are used and links to ontologies provide semantic definitions for object types.

Another key feature of OpenLABEL is scenario tagging, which allows to attach tags to external files (like scenario files) using the Tags elements. Tags are primarily conceived to annotate scenario files, such as OpenSCENARIO files, by providing scenario-relevant concepts. These tags can include environment conditions, road types, or other scenario-specific attributes. OpenLABEL allows users to select and refer to external ontologies (i.e. specifying their URI, Unique Resource Identifier), where tag terms are defined. By default, OpenLABEL also includes an ontology for scenario tagging [33] strongly based on BSI/PAS 1883. Tags apply to entire files, but can also be defined with nested properties, like numerical ranges or categorical enumerations, that allow finer scoping to elements inside the scenario file. This flexibility makes OpenLABEL an effective standard to tag or label content in a flexible way, facilitating interoperability between different tools and datasets.

4.1.4 Open Simulation Interface

The Open Simulation Interface (OSI) is an open source, standardized interface designed to enable seamless communication between components in automotive simulation environments. Developed under the umbrella of ASAM, the Association for Standardization of Automation and Measuring Systems, OSI plays a critical role in supporting a modular and interoperable simulation framework, particularly for testing advanced driver assistance systems and automated driving systems.

At its core, OSI defines a set of data structures and messages that represent the perception of the simulated environment as experienced by a vehicle's sensors. These messages are exchanged between various simulation components, such as sensor models, traffic simulators, vehicle dynamics models, and control algorithms, allowing them to interact in a synchronized and realistic way.

OSI is built on top of Google Protocol Buffers, a language neutral and platform independent serialization format. This foundation makes OSI efficient, extensible, and easy to integrate across a wide range of simulation tools and development environments. OSI has some of the following key features:

Sensor Interface: OSI provides a standardized method for simulating sensor data, including camera images, radar returns, and LiDAR point clouds. It also supports abstracted data formats such as detected objects and semantic information.

Ground Truth Data: The interface includes comprehensive representations of the simulated world, often referred to as ground truth. This includes all relevant objects, roads, participants, and environmental elements.

Scenario Information: OSI messages can carry dynamic scenario details, such as positions and velocities of vehicles, traffic signs, road markings, and traffic signals. This allows for complete and precise scenario modelling.

Sensor Configuration: OSI allows users to define virtual sensor characteristics, including position, orientation, and field of view, which enables accurate modelling of real-world sensor setups.

OSI is commonly used in closed loop simulations, where virtual vehicle subsystems operate in real time and respond to changing conditions in the simulated environment. For example, a traffic simulation tool may transmit road user information to a sensor model, which generates synthetic sensor output that is then processed by the vehicle's automated system. Because of its modular structure, OSI allows individual simulation components to be updated or replaced independently.

This flexibility has made OSI popular in both industry and research. It enables integration across tools from different vendors and supports a wide range of testing environments, including hardware-in-the-loop (HiL), software-in-the-loop (SiL), and cloud-based simulation setups for large-scale testing. OSI also works well in combination with other ASAM standards, such as OpenDRIVE for describing road geometry and OpenSCENARIO for defining dynamic scenarios.

4.2 Formats in the SUNRISE project

4.2.1 Formats Requirements Analysis

In deliverable D5.1 requirements have been defined for the Data Framework content for CCAM safety assessment. These include requirements important when defining harmonised formats for the SUNRISE project. In the deliverable the requirements were grouped into clusters. The clusters relevant for the ontology include:

- CL7 Structure of Scenario Representation: Involves requirements related to organising and categorising scenarios Data Framework.

- CL9 Scenario Filtering/Searching: Refers to a set of requirements or functionalities related to searching and filtering scenarios.

- CL11 Scenario Metadata Association: Involves requirements that focus on capturing and associating essential metadata elements with scenarios.
- CL13 Scenario Tagging/Labelling: Involves requirements related to tagging and labelling scenarios within the Data Framework.

Key individual requirements from these clusters that directly influence format definition include:

RE38: Requires that scenario metadata be included as part of the scenario representation.

RE41: Specifies that detailed metadata, such as scenario origin, date of addition, and ADS activation status, must be representable within the chosen format.

RE47: Stipulates that the scenario format must be compatible with the database formats used by SUNRISE partners.

RE49: Requires that the format support both logical and concrete scenarios, enabling representation at varying levels of abstraction.

RE56: Demands complete and unambiguous scenario descriptions, meaning the format must support all concepts defined in the SUNRISE ontology.

RE57: Requires that the scenario format include representations for dynamic objects, map data, weather conditions, and variable road infrastructures.

RE61: Mandates that V2X (vehicle-to-everything) communications be representable in the scenario format, including all relevant data content.

4.2.2 Harmonised SUNRISE Formats

To enable **seamless integration** and **consistent interpretation** of data across the SUNRISE SAF activities, various **data formats** are required at different **stages** and **interfaces** of the SAF. These formats serve distinct purposes and must be designed to accommodate the **technical** and **semantic requirements** of each step in the process.

One critical area where format definition is necessary, is in the output format for scenarios retrieved from databases. This format must present scenario data in a **clear, interpretable,** and **machine-readable** form that supports further use, whether in simulation, test case generation, or analysis. It should be flexible enough to support **filtered** or **partial retrievals**, include **comprehensive metadata**, and be suitable for export into various tools used across different SUNRISE partners. This output format also plays a key role in ensuring **reproducibility** across SAF workflows.

Also important is the input of scenarios into databases. These input formats must support the ingestion of structured scenario data, including both **logical** and **concrete representations**. They must capture essential metadata, scenario components (such as road layout, dynamic agents, and environmental conditions), and be compatible with **existing** scenario description **standards**. Moreover, the input format must be aligned with the SUNRISE ontology to ensure **semantic consistency** and support downstream processing.

In addition, there is a need for a **querying format** that enables users or systems to **search** for scenarios through the SUNRISE Federated Layer. This **federated querying mechanism** must support both keyword-based and semantically driven queries. It must allow filtering based on **parameters** defined in the ontology, such as scenario type, ODD attributes, road characteristics, and involved agents. The format for expressing these queries must be **interoperable** with the underlying data federation technologies and sufficiently expressive to support **advanced filtering**, ranking, or composition operations.

A separate format is required for **simulation execution**. Scenarios used in simulation must be translated into a format that is compatible with various **simulation platforms**. This format must not only include geometrical and temporal scenario data but also capture behaviour, initial conditions, environmental settings, and system configuration parameters. It must also accommodate format variations across different simulation engines, requiring either a **common core representation** or **mappings** to platform-specific schemas.

4.2.3 Application in the Safety Assurance Framework

Different formats are used all over the Safety Assurance Framework of the SUNRISE project. The relevant categories to consider within the SAF are the Scenario, Environment and Safety Argument blocks. An overview of the SAF is given in Figure 17.

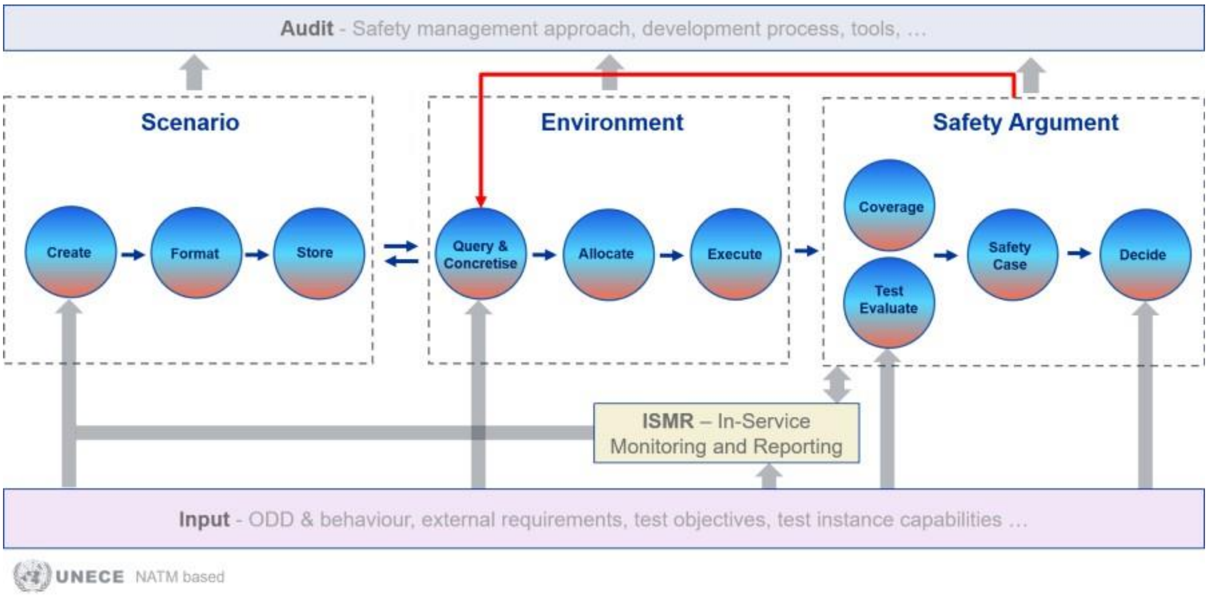

Figure 17: SAF overview.

The SUNRISE Data Framework exist at the interface between the Scenario and Environment block within the SUNRISE SAF. Each of these databases stores information on scenarios internally and allows access to this information to SAF via the DF to SUNRISE users and other external users.

Table 2: Data Formats of Connected Databases.

Database	Input Format	Output Format
ADSCENE	JSON	JSON, PDF
SafetyPool	OpenLABEL (JSON), OpenSCENARIO, OpenDRIVE, BSI 1889,	OpenLABEL (JSON), OpenSCENARIO, OpenDRIVE, BSI 1889, Videos, images
Scenius	OpenSCENARIO, OpenDRIVE	OpenSCENARIO, OpenDRIVE
StreetWise	n/a	JSON, OpenSCENARIO, OpenDRIVE
ScenarioCenter	OMEGA Format	OpenSCENARIO, OpenDRIVE

We can observe in Table 2 that different databases use different standards internally. This makes the use of common tools and methods for safety assurance like scenario-based testing more difficult, as translations have to be done. Given the likelihood of ongoing internal changes to formats and ontologies across various databases and their entities, enforcing a specific data representation format through the SUNRISE Data Framework would prevent achieving full harmonization. Instead, it is much more feasible to specify a harmonised set of standards within the SUNRISE Data Framework and let the translation to the internal content of databases be performed at the database level instead.

We can identify certain blocks and intersections within the SAF, where harmonised standards make sense. In previous sections, we have already discussed the intersection between the Scenario and Environment blocks via the querying process. We already have identified OpenLABEL as the most sensible format to enable this interchange of information between individual databases and the Data Framework. Additionally, OpenLABEL does also allow for the exchange of meta-data information if required.

As for the scenario content itself, the OpenSCENARIO and OpenDRIVE formats are by far the best choice for harmonisation, as these are the formats most databases do already support for data output and that meet the requirements set out in Deliverable D5.1. We have to further specify that OpenSCENARIO XML is to be chosen instead of DSL, which is not yet widely adopted. OpenSCENARIO XML has the benefit that the XML structure is easily parsed by a computer. While it is not as human-readable as OpenSCENARIO DSL, this is not a major concern within the SUNRISE Data Framework, as the scenarios are already defined and direct user contact to the data itself should be minimal.

Additionally, OpenSCENARIO and OpenDRIVE are a good choice since they are supported by the majority of simulation based test instances to be used within the Environment block of the SAF. This will allow for easier integration of the scenario content into the test that are performed in the Execute block.

Within the Execute block it is sensible to harmonise the format to connect individual components of the SUNRISE Simulation Framework as defined in deliverable D4.1. Here OSI is an excellent option. A use of such a format should not be mandatory, as the SAF is test instance agnostic. Users of the SAF should however consider the use of a harmonised format, as this will facilitate the use of a variety of simulation tools and models, especially those open source, that might be required depending on the use case.

Using formats within the OpenX family has the additional benefit that these formats are harmonized with the ISO standards that are also used as taxonomies of the SUNRISE ontology, ensuring harmonization between format and ontology terminology.

When it comes the creation and formatting of scenarios going into the scenario database, multiple possible solutions exists. One solution is the use of OSI, which we have discussed in a previous section. Alternatively the open source OMEGA Format could be used, which stores reference and perception data in an object-based list, combining trajectory information of dynamic objects with map data, using the HDF5 standard. Scenario creation is not a focus of the SUNRISE project. Instead the data formats being chosen in the SUNRISE sister project SYNERGIES [36] should be considered, as this project focuses on the creation of scenarios and the integration of these processes within the SAF.

5 CONCLUSIONS

In this deliverable, a **harmonised ontology** is introduced, designed for use within the SUNRISE SAF. This ontology represents a core component of the **SUNRISE Data Framework**, providing a common understanding of scenario-related content and enabling semantic consistency across systems, users, and databases. It establishes the conceptual foundation required for describing both **scenarios** and **Operational Design Domains (ODDs)** in a structured and **interoperable** manner.

The document presents general principles underpinning the ontology and includes a **comprehensive literature review** in Section 3.2 covering ontology usage in the domain of scenario-based testing, as well as an overview of existing **standards** and **formats** for ontologies and taxonomies in Section 3.3. The ontology itself is based on widely accepted industry standards and supports the formal definition of domain-relevant **concepts**, **relationships**, and **data properties**.

Within the SUNRISE Data Framework, the ontology is used to construct and execute **queries** through the Federated Layer. This enables users to search **connected scenario databases** based on specific use cases and functional requirements. By introducing a **unified semantic layer**, the ontology ensures **interoperability** across diverse toolchains, systems, and data providers. It links and formalises concepts defined in **established standards** by providing **logical relationships** and extending them with custom data properties and instance definitions. Implementation is carried out using **open, extensible formats** to ensure long-term usability and adaptability. The ontology proposed within the SUNRISE project is accessible under: https://github.com/ika-rwth-aachen/sunrise_ontology.

To operationalise the ontology within the Data Framework, a **query schema** based on the OpenLABEL format was developed. This schema enables the expression of ontology-defined concepts as **tags** and supports ontology-based **query construction** and **extension**. Queries can be further refined using parameter filters, allowing users to narrow their search scope based on scenario attributes. The use of JSON as the underlying format ensures efficient data exchange and supports integration into different external databases connecting to the SUNRISE DF. This approach also allows for the incorporation of **custom ontologies** from specific databases, enabling more detailed and context-specific querying where required.

In addition, guidelines are provided for **data interchange formats** at critical integration points within the Safety Assurance Framework. These recommendations are based on both project-specific requirements and the capabilities and constraints of participating databases.

The harmonised ontology and associated formats will be implemented in **Work Package 6 (WP6)** as part of the development of the Data Framework. This implementation will be described in in the upcoming deliverable for task T6.2, which details how the ontology can be used to construct queries for complex searching of scenario databases. They will also support activities in other tasks and work packages, helping to ensure a shared understanding of scenario content and **consistent data representation** across the SUNRISE toolchain. An example is harmonisation of concepts and formats in the test execution within the **Harmonized V&V Simulation Framework**.

A dedicated ontology for SUNRISE was developed as no publicly available ontology at the time sufficiently covered the necessary scope of scenario content. As standards like ASAM OpenXOntology continue to evolve, aligning with these initiatives in the future is recommended to enhance **broader compatibility** and **acceptance** within the domain of safety assurance and scenario-based testing for Connected, Cooperative, and Automated Mobility (CCAM). The ontology presented in this deliverable can serve as **valuable input** to ongoing efforts such as ASAM OpenXOntology.

6 REFERENCES

[1] ISO, „ISO 34501: Road vehicles — Test scenarios for automated driving systems," 2022.

[2] T. Menzel, G. Bagschik und M. Maurer, „Scenarios for Development, Test and Validation of Automated Vehicles," in *IEEE Intelligent Vehicles Symposium*, 2018.

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

[4] L. Otero-Cerdeira, F. J. Rodríguez-Martínez und A. Gómez-Rodríguez, „Ontology matching: A literature review," *Expert Systems with Applications,* Bd. 42, p. 949–971, 2015.

[5] M. Zipfl, N. Koch und J. M. Zöllner, *A Comprehensive Review on Ontologies for Scenario-based Testing in the Context of Autonomous Driving,* 2023.

[6] M. Grabowski und Y. Wang, „A Concept to Support AI Models by using Ontologies – Presented on the Basis of German Technical Specifications for Lane Markings," in *Proceedings of the 27th International Technical Conference on the Enhanced Safety of Vehicles (ESV)*, Yokohama, 2023.

[7] S. Geyer, M. Baltzer, B. Franz, S. Hakuli, M. Kauer, M. Kienle, S. Meier, T. Weißgerber, K. Bengler, R. Bruder, F. Flemisch und H. Winner, „Concept and development of a unified ontology for generating test and use-case catalogues for assisted and automated vehicle guidance," *IET Intelligent Transport Systems,* Bd. 8, p. 183–189, 2014.

[8] L. Mlodzian, Z. Sun, H. Berkemeyer, S. Monka, Z. Wang, S. Dietze, L. Halilaj und J. Luettin, „nuScenes Knowledge Graph - A Comprehensive Semantic Representation of Traffic Scenes for Trajectory Prediction," in *IEEE/CVF International Conference on Computer Vision (ICCV) Workshops – Scene Graphs and Graph Representation Learning (SG2RL)*, 2023.

[9] E. de Gelder, J.-P. Paardekooper, A. Khabbaz Saberi, H. Elrofai, O. Op den Camp, S. Kraines, J. Ploeg und B. De Schutter, „Towards an Ontology for Scenario Definition for the Assessment of Automated Vehicles: An Object-Oriented Framework," *IEEE Transactions on Intelligent Vehicles,* Bd. 7, p. 300–314, 2022.

[10] C. Chang, D. Cao, L. Chen, K. Su, K. Su, Y. Su, F.-Y. Wang, J. Wang, P. Wang, J. Wei, G. Wu, X. Wu, H. Xu, N. Zheng und L. Li, „MetaScenario: A Framework for Driving Scenario Data Description, Storage and Indexing," *IEEE Transactions on Intelligent Vehicles,* Bd. 8, p. 1156–1175, 2022.

[11] D. Bogdoll, S. Guneshka und J. M. Zöllner, „One Ontology to Rule Them All: Corner Case Scenarios for Autonomous Driving," in *European Conference on Computer Vision (ECCV)*, 2022.

[12] Z.-X. Xia, W.-C. Lai, L.-W. Tsao, L.-F. Hsu, C.-C. H. Yu, H.-H. Shuai und W.-H. Cheng, „A Human-Like Traffic Scene Understanding System: A Survey," *IEEE Industrial Electronics Magazine,* Bd. 15, p. 6–15, 2021.

[13] M. Tibau, S. Siqueira und B. P. Nunes, „Semantic Data Structures for Knowledge Generation in OpenWorld Information System," in *Proceedings of the XVI Brazilian Symposium on Information Systems*, 2020.

[14] J. Luettin, S. Monka, C. Henson und L. Halilaj, *A Survey on Knowledge Graph-based Methods for Automated Driving,* 2022.

[15] L. Halilaj, J. Luettin, C. Henson und S. Monka, „Knowledge Graphs for Automated Driving," in *2022 IEEE Fifth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, 2022.

[16] Q. Zhang, Z. Ma, P. Zhang und E. Jenelius, „Mobility knowledge graph: review and its application in public transport," *Transportation,* 2023.

[17] V. Ryen, A. Soylu und D. Roman, „Building Semantic Knowledge Graphs from (Semi-)Structured Data: A Review," *Future Internet,* Bd. 14, 2022.

[18] P. Bellini, S. Bilotta, E. Collini, M. Fanfani und P. Nesi, „Data Sources and Models for Integrated Mobility and Transport Solutions," *Sensors,* Bd. 24, p. 441, 2024.

[19] S. Sarwar, S. Zia, Z. Qayyum, M. Iqbal, M. Safyan, S. Mumtaz, R. G. Castro und K. Kostromitin, „Context aware ontology-based hybrid intelligent framework for vehicle driver categorization," *Transactions on Emerging Telecommunications Technologies,* Bd. 33, 2019.

[20] M. Ryu und S.-H. Cha, „Context-Awareness based Driving Assistance System for Autonomous Vehicles," *International Journal of Control and Automation,* Bd. 11, p. 153–162, 2018.

[21] M. Viktorović, D. Yang und B. de Vries, „Connected Traffic Data Ontology (CTDO) for Intelligent Urban Traffic Systems Focused on Connected (Semi) Autonomous Vehicles," *Sensors,* Bd. 20, p. 2961, 2020.

[22] K. Janowicz, A. Haller, S. J. D. Cox, D. L. Phuoc und M. Lefrançois, „SOSA: A lightweight ontology for sensors, observations, samples, and actuators," *Journal of Web Semantics,* Bd. 56, p. 1–10, 2019.

[23] I. Urbieta, M. Nieto, M. García und O. Otaegui, „Design and Implementation of an Ontology for Semantic Labeling and Testing: Automotive Global Ontology (AGO)," *Applied Sciences,* Bd. 11, p. 7782, 2021.

[24] H. Toulni, B. Nsiri, M. Boulmalf und S. Tayeb, „An ontology based approach to traffic management in urban areas," *International Journal of Systems,* Bd. 9, p. 54–59, 2015.

[25] BSI, „BSI PAS 1883: Operational Design Domain (ODD) taxonomy for an automated driving system (ADS) - Specification," 2020.

[26] ISO, „ISO 34503: Road Vehicles — Test scenarios for automated driving systems — Specification for operational design domain," 2023.

[27] ISO, „ISO 34504: Road vehicles — Test scenarios for automated driving systems — Scenario categorization," 2024.

[28] ASAM, „ASAM OpenODD," [Online]. Available: https://www.asam.net/project-detail/asam-openodd-v100/. [Zugriff am 29 05 2025].

[29] ASAM, „OpenLABEL," [Online]. Available: https://www.asam.net/standards/detail/openlabel. [Zugriff am 20 05 2025].

[30] F. Alakkad, X. Zhang, S. Khastgir, E. de Gelder, S. van Montfort, J. L. Mallada, E. van Hassel, M. Grabowski, J. Beckmann, X. Boaben, A. Nogues und S. Vidal, „D5.1 Requirements for CCAM safety assessment data framework content," SUNRISE, 2023.

[31] SUNRISE Project, „SUNRISE Ontology," [Online]. Available: https://github.com/ika-rwth-aachen/sunrise_ontology.

[32] M. Freeman, X. Zhang, S. Khastgir und P. Jennings, „Tagging Scenarios With Custom Tags Whilst Retaining Scenario Exchangeability," in *DSC 2023 Europe VR*, 2023.

[33] ASAM, „ASAM OpenLABEL Ontology," [Online]. Available: https://openlabel.asam.net/V1-0-0/ontologies/openlabel_ontology_scenario_tags.ttl. [Zugriff am 29 05 2025].

[34] ASAM, „OpenLABEL JSON Schema," [Online]. Available: https://openlabel.asam.net/V1-0-0/schema/openlabel_json_schema.json. [Zugriff am 29 05 2025].

[35] F. Alakkad, X. Zhang, S. Khastgir, J. Beckmann, T. Menzel, X. Boaben, S. Vidal, E. Mungan, T. Singh, E. van Hassel, M. Grabowski, J. Broos, M. Nieto, J. Valdivieso und A. Ballis, „D6.1: Methodology for SCDB application for generic use cases,“ 2024.

[36] SYNERGIES Consortium, „SYNERGIES Project,“ [Online]. Available: https://synergies-project.eu/. [Zugriff am 21 05 2025].

ANNEX 1: ONTOLOGY MAPPING EXAMPLE

Annex 1 contains a possible mapping of the ontology to a query format. It can give some guidance on how to structure a query based on the SUNRISE ontology.


```
{
  "openlabel": {
    "tags": {
      "odd": [
        {
          "name": "rainfall_event",
          "type": "rainfall",
          "ont_uid": "SUNRISE#Rainfall",
          "val": {
            "text": [
              {
                "name": "Rainfall_intensity",
                "val": "@my_rainfall_intensity"
              },
              {
                "name": "Rainfall_type",
                "val": "@my_rainfall_type"
              },
              {
                "name": "Temperature",
                "val": "@my_temperature"
              }
            ]
          }
        },
        {
          "name": "my_rainfall_intensity",
          "type": "Rainfall_intensity",
          "ont_uid": "SUNRISE#No_rain",
        },
        {
          "name": "my_rainfall_type",
          "type": "Rainfall_type",
          "ont_uid": "SUNRISE#Dynamic_rain",
        },
        {
          "name": "my_temperature",
          "type": "Temperature_scalar",
          "ont_uid": "SUNRISE#Temperature_scalar",
          "val": {
            "text": [
              {
                "name": "temperatureValue",
                "val": "20"
              },
              {
                "name": "unit",
                "val": "Celsius"
              }
            ]
          }
        }
      ],
      "administrative": [],
      "behaviour": [],
      "custom": []
    },
    "ontologies": {
      "SUNRISE": "https://ccam-sunrise-project.eu/ontology/"
    }
  }
}
```