23rd International Conference on Modeling & Applied Simulation, 008
21st International Multidisciplinary Modeling & Simulation Multiconference

2724-0037 © 2024 The Authors.
doi: 10.46354/i3m.2024.mas.008

Scenario Modeling for the Virtual Validation and Verification of Cloud-based Mobility Services

Philipp Heisig[1,*] and Sabine Sachweh[1]

[1]Dortmund University of Applied Sciences and Arts, Sonnenstraße 96, Dortmund, 44139, Germany

[*]Corresponding author. Email address: philipp.heisig@fh-dortmund.de

Abstract

The ever-evolving automotive domain undergoes a fundamental shift from hardware-based to software-centric high-tech products with a high level of service integration. Connected and automated vehicles enable data-driven innovations through modern, upgradeable, flexible, and extensible software architectures, artificial intelligence, and specialized hardware with built-in connectivity. Such vehicles operate in a safety-critical and time-sensitive environment and are subjected to various nonfunctional requirements including quality, reliability, security, and safety. However, testing these requirements poses several challenges and requires large heterogeneous data sets from real-world scenarios. While on-road testing is a huge effort in both time and cost, traffic simulations in combination with further simulations allow to prove the technical feasibility and reduce the risks for sophisticated and expensive software developments. Currently, a lot of research is conducted in the domain of automated driving systems, but there is a lack of simulation-based testing approaches that focus on the connectivity dimension and, in particular, mobility services running in the cloud and serving multiple vehicles at scale. Therefore, we propose a modeling approach to describe scenarios that involve cloud-based mobility services. More precisely, an ontology along with a corresponding domain-specific language is introduced that allows one to formally represent the domain concepts, their characteristics, and interrelationships through model-based scenario descriptions. Furthermore, we discuss the notation for the language and propose a web-based user interface that abstracts domain complexity.

Keywords: Connected Vehicles; Traffic Simulation; Cloud Computing; Model-driven Engineering

1. Introduction

Technological advances, digitization, and area-wide mobile Internet have transformed traditional vehicles into software-based high tech products with built-in connectivity and autonomous driving features. Connected and Automated Vehicles (CAVs) will have a large share in the transition toward an efficient transport system that provides good and safe transport services to all. CAVs are characterized by the increasing usage of complex software such as deep learning, high-performance computing, and the massive amounts of multi-modal data emitted by hundreds of various sensors. Especially range sensors such as Radar and LiDAR as well as cameras produce a lot of high-quality data to provide context information about the vehicle itself and its environment, e. g. to detect road conditions, measure distance to other vehicles, or recognize driver fatigue. While traditional vehicles rely on data processed locally in the vehicle, CAVs integrate an additional connectivity dimension to share data beyond the confines of a single vehicle. Vehicular communication allows CAVs to receive, for example, additional information about the state and intentions of other vehicles, thus fostering cooperation among each other and providing better vehicle awareness. With the evolution of mobile cell communication, CAVs are even capable of ubiquitously exposing cloud computing resources by sending and receiving data to and from cloud-based mobility services. Such mobility services facilitate vehicle data collection and processing at scale in multiple and simultaneously operating cloud services. In particular areas like road safety benefit from an additional connectivity dimension, e. g. by sharing information about road conditions with upcoming vehicles.

As vehicles operate in a safety-critical and time-sensitive environment with changing conditions, cloud-based mobility services pose several challenges that go beyond the requirements of other Internet of Things (IoT) domains. These services must scale with the increasing number of CAVs on the road and provide functionality in a reliable way, especially when dealing with safety-related functions. For example, unreliable vehicle connectivity with changing data transfer rates, especially in rural areas, must be expected. Consequently, the dynamic nature of CAVs poses a significant challenge for the design and implementation of cloud-based mobility services, but also for the validation and verification (V&V) process. In contrast to traditional testing approaches for automotive embedded software systems, it is not enough to test only the single vehicle itself and its behavior. Mobility services deployed in the cloud serve multiple and a varying number of participants on demand. Thus, the service must be tested with a wide range of various CAVs, including their interaction with the environment and other vehicles in traffic. Heterogeneous data sets at scale from real-world scenarios are required instead of randomly generated fake test data to ensure both the proper functionality of a mobility service and that the architecture fulfills all quality of service requirements for a varying number of different vehicle types. Virtual testing by means of traffic simulations in combination with further simulations allows to prove the technical feasibility and reduce the risks for sophisticated and expensive software developments. Setting up an appropriate testing environment based on simulation is, however, not an easy task and requires expert and domain knowledge, which may prevent Small- and Medium-sized Enterprises (SMEs), cities and municipalities, or third-party service provider yet outside of the automotive domain, such as an insurance, from realizing innovative and cross-domain mobility services. An important aspect for such a virtual proof-of-concept is the initial description of the scenario and its different settings. Existing scenario modeling approaches for CAVs put a strong focus on automated driving systems (ADS) and thus the vehicle itself, but often neglect connectivity and in particular the cloud dimension.

In the following, we present an explicit designed scenario modeling approach for the V&V process of cloud-based mobility services. Based on a domain ontology, a domain-specific language (DSL) is introduced to formally describe different CAV scenarios with respect to the cloud dimension. In addition, we provide an overview of the state of research for testing cloud-based mobility services via simulations and discuss potential limitations of our approach. Initially, Section 2 provides background information on cloud-based mobility services, while Section 3 gives an overview of related work. Section 4 then introduces an ontology specifically designed for cloud-based mobility services, which act as the basis for the design of a model-based scenario description in Section 5. Finally, Section 6 discusses the results and drawbacks, before Section 7 concludes our work.

2. Background

The rapid growth and the tremendous number of CAVs on the road make them a major element of IoT and enable vehicles to share data with each other and access cloud resources to manage data, enable advanced analytics, and provide new services and applications.

2.1. V2X Communication

An increasing distribution of machine-to-machine communication has commoditized cellular bandwidth and hardware. The automotive domain also has been adapted to the lower cost structure and most vehicles are shipped with an integrated modem nowadays. The built-in connectivity allows vehicles to connect with each other and, on the basis of cellular networks, to remote servers. This enables CAVs to receive additional context information to create a comprehensive and dynamic understanding of the environment, e. g. sharing information about upcoming bad weather conditions. In general, inter-vehicle connectivity can be summarized under the term V2X communication, where X can stand for any entity such as vehicle, cloud, infrastructure, or even vulnerable road users. MacHardy et al. (2018) provide a detailed overview of V2X communication and its historical development. V2X communication can be technical relying on a variety of wireless communication protocols and it is still an ongoing topic in both research and practise on which technology to use (Ali et al., 2024). The three technologies most discussed are DSRC, LTE C-V2X, and the emerging 5G NR V2X. A recent study by Ali et al. (2024) compare these technologies with detailed information on its applicability and drawbacks. Although there is no clear definition among researchers regarding the different V2X terms, we will use the term vehicle-to-cloud (V2C) in the following to refer to long-range communication based on cellular networks between vehicles and cloud servers.

2.2. Cloud-based Mobility Services

With the evolvement of V2C communication, CAVs are capable of ubiquitously exposing cloud computing resources to cope with exponential growth in complexity and volume of data (Alvarez-Coello et al., 2021). Compared to In-vehicle processing, cloud computing enables CAVs to offload computation tasks and vehicle data to utilize parallel processing, data persistence, and accessibility (Mostefaoui et al., 2022). Cloud-based mobility services operate upon networks of vehicles and infrastructure devices to serve a variety of vehicles on-demand. Especially aggregated data from a vehicle fleet and multi-source data fusion allow us to generate novel knowledge. Due to their integration in IoT ecosystems, cloud-based mobility services can not only leverage vehicle data, but also consider further context information, e. g. weather forecasts, traffic conditions, or data from roadside infrastructure, smart cities, etc. Early implementations of cloud-based mobility services focused on telematics applications, e. g. diagnostic electronics, whereas future CAVs will exhibit a cloud-based vehicle architecture to allow for a much higher level of service integration and pave the way to an increased number of use cases, spanning from road safety over smart, efficient, and green transportation to location-dependent services. Although cloud-based mobility services offer many advantages and opportunities for next-generation mobility, they require a sophisticated design, development, deployment, and operation to handle automotive big data (Mostefaoui et al., 2022). Currently, the design and implementation of cloud-based mobility services is often based on the microservice architecture (MSA) style, as it features, among other things, scalability, modularity, and flexibility.

3. Related Work

This section provides an overview of related work on scenario-based testing for CAVs. To the best of our knowledge, there is currently no approach available that focus on scenario-based testing for the cloud dimension of CAVs.

3.1. Scenario-based Testing

Scenario-based testing is an established testing approach within the automotive industry and has also gained a lot of attraction in research (Junietz et al., 2018), especially for the validation of software in the ADS domain (Bach et al., 2016; Hallerbach, 2020; Irvine et al., 2022; Reichsöllner et al., 2022). For example, Bach et al. (2016) propose a model-based scenario specification for ADS functions. One of the main challenges for scenario-based testing is the definition of a structure to capture the complexity of reality (de Gelder et al., 2022). A good overview regarding the state-of-the-art for virtual, scenario-based V&V of highly automated vehicles is given by Batsch et al. (2021).

Reichsöllner et al. (2022) developed a simulation environment to analyze autonomous driving scenarios in cities with a particular focus on shared autonomous vehicle fleets. Their tool, called SUMO4AV, comprises the OSMWebWizard to import road networks and Point of Interest (POI) from Open Street Map (OSM) and the Eclipse SUMO traffic simulator for running the simulation. In addition, the authors provided a workaround to interact with map entities, in particular POIs and parking areas, during the simulation. They evaluated their approach with map data from the city of Mannheim (Germany) and different routing strategies. To overcome current limitations regarding the V&V process for CAVs, Irvine et al. (2022) propose a V2X extension that enables communication between vehicles, infrastructures, and other entities for an ADS scenario description language. Their extension includes eight attributes relevant for V2X communication: Communications Capability, Transceiver Directionality, Casting Type, Transmission Type, Transmission Size, Transmission Time (of flight), Transmission Signal Strength, and Message Type. However, the focus is still on the description of complex cooperative automated driving scenarios that involve V2X communication and do not include the cloud dimension.

3.2. Ontologies

According to Studer et al. (1998), an ontology is "a formal, explicit specification of a shared conceptualization.", where conceptualization can be defined as "an abstract, simplified view of the world that we wish to represent for some purpose." (Genesereth and Nilsson, 2012). Basically, an ontology includes the vocabulary and definition of concepts and their relationships for a given domain.

Zipfl et al. (2023) provide an overview of key representative ontologies for scenario-based testing in the field of autonomous driving. In general, such ontologies aim at describing test scenarios for ego vehicles in temporal scenes on a detailed level, e. g. geometry information of objects like road surfaces. This level of detail is necessary to validate the functionality of ADS-based decision making. As stated by Yazdizadeh and Farooq (2020), the inclusion of concepts and entities related to connected roadways and IoT technologies in a transportation ontology has not yet been addressed. Katsumi and Fox (2018) provide a comprehensive overview of transportation ontologies and evaluated them based on different criteria. However, the ontologies considered therein are rather abstract or designed for specific applications such as road accidents, city logistics, or public transport monitoring. The Vehicle Signal Specification (VSS) ontology (Klotz et al., 2018) is built on top of the VSS data model and extends it with more expressive semantics to describe and interact with vehicle data, which is especially useful when connecting vehicle signals to other domains. Viktorović et al. (2020) introduce the Connected Traffic Data Ontology for the CAV data layer with a particular focus on large volumes of time-sensitive data, i. e. sensor and geospatial data. They considered the speed, acceleration, and geolocation of vehicles for their ontology and validated their approach with vehicle data from a running SUMO simulation. The recently published standard ISO 34503 defines a taxonomy to enable the safe deployment of a level 3 and level 4 ADS. Although the focus is on the vehicle itself, some of the static and dynamic attributes are also of interest.

As CAVs operate in a cross-domain environment, ontologies that are not directly associated with the automotive domain should also be considered. Yazdizadeh and Farooq (2020) give an overview on ontologies for smart mobility in general and across different domains such as transportation, smart cities, and sensors. For geospatial data, OSM (Stadler et al., 2012) and GeoSPARQL (Battle and Kolas, 2011) are the most prominent ontologies. While both allow to store geo-data, GeoSPARQL likewise supports the handling of data. Weather, as another typical cross-domain aspect, has also been specified by distinct ontologies. For example, Chen and Kloul (2018) formulated a weather ontology tailored towards ADS. A more intricate weather ontology within the realm of smart cities is given by Bellini et al. (2014). Their ontology was meticulously constructed via real-time weather data.

3.3. Traffic Scenarios

Existing realistic traffic scenarios for city-wide areas that are freely available are rare. Schrab et al. (2022) provide a large-scale traffic scenario for SUMO that illustrates an entire day of private motorized traffic in the urban area of Berlin with more than 2.2 million trips within an area of 800 km^2. In the same way, Yamazaki et al. (2023) provide a full-scale SUMO traffic scenario for a residential district within the city of Tokyo (32.22 km^2) with 298,310 trips. Other traffic scenarios are available for the cities of Bologna (Bieker et al., 2015), Ingolstadt (Lobo et al., 2020), Luxembourg (Codecá et al., 2017), Monaco (Codecá and Härri, 2017), or Turin (Rapelli et al., 2021). Although these scenarios provide a realistic traffic demand at scale, they are specific to a certain area and point in time with predefined vehicles and attributes.

4. Domain Ontology

This section introduces an ontology for the domain of CAVs with a particular focus on scenarios involving cloud-based mobility services. The domain ontology defined here helps to specify relevant domain entities including their attributes, encapsulate the most precise knowledge from the viewpoint of that domain, and provide a common understanding among different stakeholders. As scenarios act as a basis for the simulation-based generation of large synthetic test data, we also consider aspects related to traffic simulation and network simulation. Following a top-down approach, we first define the general concepts and subsequently specify them. The ontology is classified into four aspects, namely Traffic, Vehicle, Network & Communication, and Environment. Domain knowledge is mainly extracted from related work and existing traffic scenarios, but also from other ontologies if applicable.

4.1. Traffic

The term **traffic** represents a mixed traffic flow with various vehicles at scale for a given area. Today, a wide variety of vehicle types are on the road with different purposes and characteristics, e. g. connected or nonconnected, motorised or nonmotorised, and private or public. Typical **vehicle types** in a CAV scenario include cars, trucks, motorcycles, emergency vehicles, and public transport such as buses. Meanwhile, two-wheelers, e. g. electric bicycles or electric scooters, are also digitized and capable of sensing the environment, processing data, and communicating with remote servers. Thus, they are an inevitable part of today's traffic as well. Vehicle types consist of some static vehicle type values such as length, height, weight, or person capacity. **Traffic volume** represents the number of vehicles of all types for a given area and a certain period of time, e. g. vehicle per hour. Typically, traffic volume varies considerably during the day with peak times in the morning and evening. To achieve precision when simulating a traffic system, it is crucial to have timely data on **traffic demand**, which can be succinctly described as the set of all vehicles within a traffic system along with their origins, destinations, and start times. This information is accessible through city municipalities, collected via sensors embedded in the urban environment, or based on travel demand models (López Díaz and Tundis, 2023). For testing cloud-based mobility scenarios, it is sufficient to provide an almost realistic traffic demand through simulations with the option to integrate existing real-world scenarios for specific areas. **Vehicular routing** in traffic simulation aims at routing vehicles from their respective origin to their destination. Again, for the sake of simplicity, the simulator's default routing algorithms can be used to find the shortest path for each vehicle and a given road network. For specific scenarios it is still possible to alter the vehicle routing by sending driving commands to vehicles.

4.2. Vehicle

Vehicles are the host of drivers and passengers and the fundamental component of all CAV scenarios. Vehicles are aware of their own state, e. g. data from physical components such as position, speed, and acceleration. In addition, they can perceive the environment by sampling and collecting data from built-in sensors such as cameras, radar, or ultrasonic. CAVs represent vehicles that can share these sensor measurements not only with other vehicles but also with the cloud. They are also able to receive commands and additional context information outside of a vehicle's line of sight from specific cloud-based mobility services. Therefore, vehicles integrate a communication module and employ a distinct **messaging protocol**, such as HTTP, MQTT, uProtocol, or Zenoh, to send and receive data via V2C communication. Each individual vehicle can generate and share a large amount of various **vehicle-specific data**. Typical dynamic vehicle telemetry data that are supported by all traffic simulators include driving data such as acceleration and speed. As every entity in a scenario should be locable on a map through a spatial property, vehicles are also aware of their **geolocation** that is typically represented by longitude and latitude coordinates. Simulation of emission data (e. g. CO_2, NO_x, noise, fuel consumption) is also supported by certain traffic simulators such as Eclipse SUMO. Despite this, vehicle data that are relevant in the scope of cloud-based mobility services are multiple and can become very detailed, e. g. data regarding vehicle dynamics for predictive maintenance. To avoid costly simulation steps, it is necessary to provide certain vehicle data in high granularity. For example, weather-related data (temperature, rain, etc.), measurements about road conditions, and data on vehicle health status indicated by error codes, e. g. OBD2 diagnostic trouble codes, are often part of mobility scenarios and should be provided at a high abstraction level as this information detail is sufficient for further processing at the cloud.

4.3. Network & Communication

V2X communication enable vehicles to interact with different entities based on different types of communication. Consequently, the connectivity dimension plays an important role in almost every CAV scenario. Despite the underlying technology, i. e. DSRC or C-V2X, V2X networks generally face stringent bandwidth limitations that can substantially restrict the amount of data that can be transmitted. Also, messages may have a delay or never arrive and get lost. Such aspects must be thoroughly considered and tested when designing and operating cloud-based mobility services as otherwise services may not work as expected. Although V2V communication among vehicles is undeniably an inevitable part of future mobility, it can be typically neglected for testing the cloud dimension in CAV scenarios. Instead, **V2C** communication based on cellular networks is the most relevant because it connects vehicles to cloud computing. Mobility services that are operated in the cloud are often far away from vehicles which may result in significant latency due to network congestion or queuing. A crucial aspect of any cellular network simulation tool is its ability to replicate authentic propagation models, considering various factors that influence wireless signals, including terrain, buildings, vegetation, weather conditions, interference, and signal fading. However, these aspects are more relevant for the planning or optimization of existing cellular networks and can be neglected for the V&V process of cloud-based mobility services.

4.4. Environment

The environment consists of all static and dynamic physical objects with which vehicles interact (de Gelder et al., 2022). The static environment refers to objects that do not change during a scenario simulation and includes geospatially stationary elements like the road network, POIs, or topological information about the road surface. In contrast to the static environment, dynamic parts of the environment can change during the run-time of a scenario, e. g. environmental conditions such as lighting or weather. As scenarios are snapshots of reality, the description of both static and dynamic environments can become rich in detail and complex. Although the testing of in-vehicle functionality, in particular the testing of an ADS, requires a very detailed environment simulation, such details are in general not relevant for assessing cloud-based mobility services and the performance of its underlying software architecture. Thus, we can neglect most of the details of the environment or provide them in high granularity. For example, let us assume a scenario in which vehicles automatically detect and assess road conditions through their integrated sensors and share this information via the cloud with the municipal government for road maintenance or other upcoming vehicles as hazard warning. The testing process of the In-vehicle software component for the detection and assessment of road conditions would require some detailed and steady simulation of road surfaces including aspects such as hilliness and cross-slope. In contrast, the level of detail of the data sent to the cloud for data processing and storage is relatively low, i. e. it would be enough to send the geolocation and the type of road condition observed (dry, wet, slick, damage, obstacle). Thus, it is sufficient to provide an abstract description of the road condition and to simulate certain events in time when, for example, an obstacle is detected. The **road network** encompasses the layout of roads and lanes and can be classified into different types of road such as rural, urban, and highway, among others. Realistic traffic scenarios include real-world road topologies in which motorized vehicles move. Cycling and pedestrian networks are also relevant as they are often an inevitable part of road networks and vehicles interact with entities on them. For testing cloud-based mobility services, it is sufficient to abstract high-detailed road networks with, for example, lane markings and use the road network only for vehicle routing. In addition, the road network can be extended with services areas that are annexed to the road network and offer some kind of service, e. g. gas station or parking. Other types of transportation network, such as railways, waterways, airways, or cableways, will be neglected for now due to the main focus on road vehicles. **Weather conditions**, like sun, wind, ice, fog, etc., can significantly affect a CAV scenario and are relevant for several use cases (Grimm et al., 2023; Jiang et al., 2023; Marosi et al., 2018; Mostefaoui et al., 2022; Rahman et al., 2018). Thus, we will also integrate the option to describe different and changing weather conditions for a scenario. This affects primarily road conditions, but can also be used for other use cases. Traffic infrastructure, such as traffic signals, roadside units, camera/radar/loop detectors, and electronic traffic signs, will not be considered for now as it is more related to edge computing.

5. Scenario Modeling Language

In the field of software engineering, the use of models is a well-established and commonly used method to abstract and transfer information, manage inherent complexity, and improve development capabilities. According to de Gelder et al. (2022), a model-based scenario description has several advantages:

- Having a clear and formal description of the scenarios is essential for conducting standardized, repeatable, and reproducible tests.
- Automated comparison and classification become more feasible with standardized scenario descriptions (de Gelder et al., 2020).
- Models foster a qualitative description that is also readable and understandable to human experts.

Following the domain ontology defined in the previous

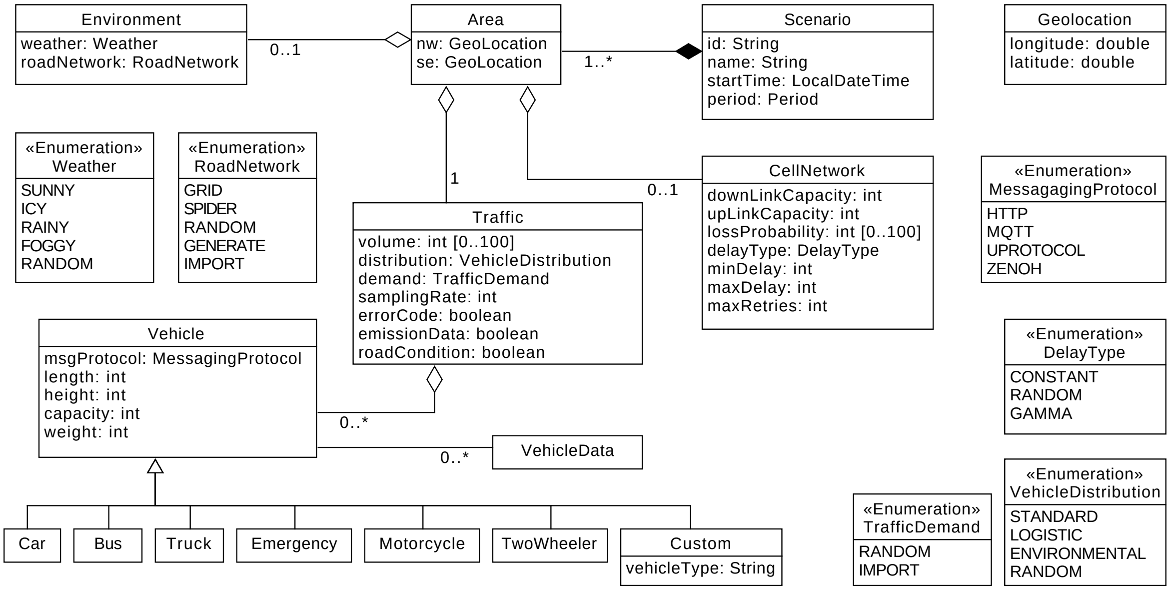


Figure 1. UML class diagram of the scenario metamodel

section, this section proposes a DSL based on a metamodel to formally describe CAV scenarios. In general, a DSL is made up of three fundamental elements: abstract syntax, concrete syntax, and semantics.

5.1. Scenario Metamodel

As a first step, we introduce a metamodel to specify the abstract syntax and semantics of our scenario modeling approach. Metamodels function as an explicit description of how a domain-specific model can be created. Domain models, in our case, represent concrete scenarios for cloud-based mobility services and will be referred to as *scenario model* in the following. Every *scenario model* is a formalized instance of the metamodel and therefore conforms to its defined structure, vocabulary, and concepts to promote uniformity among models and enable automatic processing by software tools. The main purpose of a *scenario model* is to automatically derive co-simulation environments for the generation of synthetic data specific to the described scenario. Aspects regarding Traffic, Network & Communication, and Environment describe how the scenario should be simulated and will be used as input for the simulator configurations as described in our previous work (Heisig and Flick, 2021). As scenarios can consist of thousands of vehicles, it would require considerable effort to model them manually. Thus, individual vehicles including their properties will be automatically generated by the according traffic simulator. In addition, we treat CAVs as black boxes that send and receive data to and from cloud computing, but do not consider In-vehicle data processing and also abstract the built-in communication module of single vehicles as data transmission is simulated in a generic way based on the modeled cellular network.

While V&V activities for an ADS require high-fidelity scenarios, we strive to model low-fidelity scenarios at scale to test cloud-based mobility services. For the design of the metamodel and the parameters it should cover, we investigated existing scenarios, standards, and parameterization options of different simulators, namely Eclipse SUMO, VISSIM, Eclipse MOSAIC (Network settings), OMNeT++, and ns-3. Unified Modeling Language (UML) diagrams have been predominantly used by the software engineering community as a standardized visual language to illustrate the architecture of a software system, along with its actors, concepts, and their interrelationships (Yazdizadeh and Farooq, 2020). As shown in Figure 1, we use an UML class diagram to represent our metamodel and the different entities including their properties and relationships among each other as defined in the previous section.

A `Scenario` can be uniquely identified using an ID and has a name, a starting time, and a period of time for which the scenario should be simulated. A `Scenario` is composed of one or more instances of an `Area`. In this way, the user has the option to define regions with individual settings, e. g. different weather conditions, network performance, crowded areas, etc. An `Area` has the form of a rectangle defined by its northwestern and southeastern `Geolocation`, which represent longitude and latitude values.

An `Area` contains a description of its `Traffic` and optionally about its `Environment` and `CellNetwork`. `Traffic` within an `Area` can be specified by its volume, i. e. the number of vehicles on the road, and the distribution of vehicle types. `STANDARD` represents a common distribution with mostly cars on the road, whereas `LOGISTIC` put a focus on the transportation domain with a relatively large volume of trucks and `ENVIRONMENTAL` simulates a large share of electric vehicles and two-wheelers. In addition, traffic demand can either be generated randomly by the respective simulator or an existing traffic demand model can be imported. The sampling rate on how often vehicles send their data to the cloud can also be adjusted. To restrict the amount of vehicle data needed for the scenario, users can also model if vehicles should generate error codes, emission data, and weather-related data, e. g. to determine road conditions.

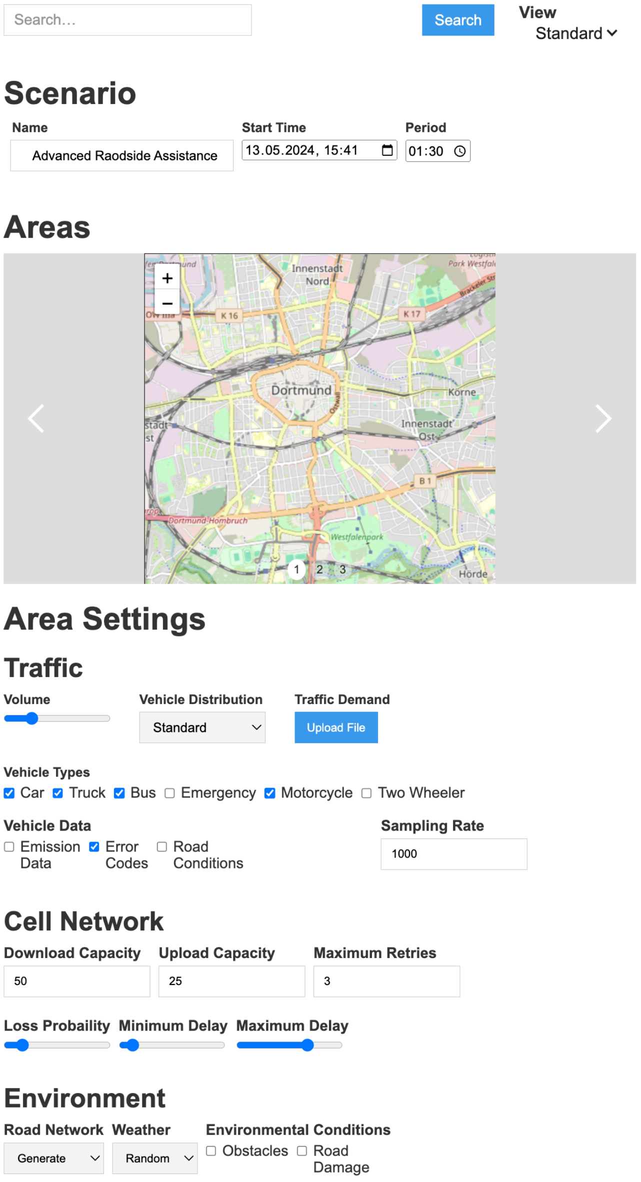
Traffic is made up of an arbitrary number of `Vehicle` that consist of a communication module to send and receive data to and from the cloud. A vehicle employ a specific `MessagingProtocol` that is `HTTP`, `MQTT`, `UPROTOCOL`, or `ZENOH`. Vehicles are also distinguished by their type, which can be `Car`, `Bus`, `Truck`, `Motorcycle`, `TwoWheeler`, `Emergency`, or a `Custom` type defined by a string value.

The `Environment` encompasses settings for the weather conditions in a certain `Area` (`SUNNY`, `RAINY`, `ICY`, or `FOGGY`) and with respect to the topology of the road network. Typically, the road network will be generated from real-world maps like OSM or an existing one can be imported. As an alternative, synthetic road networks can be generated in the form of a grid, a spider, or completely random.

With `CellNetwork`, rudimentary network performance settings for an `Area` can be specified. This includes the maximum uplink and downlink capacity of the network, the probability that messages get lost, the maximum retries to deliver a message, and settings to simulate network delays including a minimum and maximum delay as well as the model of the (randomized) delay distribution.

5.2. Scenario Editor

After defining the abstract syntax and semantic via the metamodel, the next step is to specify the metamodel-specific concrete syntax, which acts as user interface (UI) to create formalized domain models that conform to the metamodel. Although textual DSLs are a great way to formally describe domain concepts, we found that their usability is often not evaluated (Barisic, 2017) and may be limited to certain notation. Depending on the complexity of the DSL and the applied framework, maintenance and evolution may also be aggravated. Instead of creating a textual DSL, we therefore propose the usage of a graphical DSL by means of a web-based UI composed of different HTML5 elements, i. e. text input fields, checkboxes, etc. Having a web-based interface, the usage of advanced UI elements, such as the integration of OSM for the selection of an area, is also possible. Type checking ensures that user input will be provided in the right data types and further validation is conducted regarding the range of data, e. g. longitude values are only valid from -180 to 180. In this way, *scenario models* are syntactically and semantically validated.



Scenario

Name: Advanced Raodside Assistance
Start Time: 13.05.2024, 15:41
Period: 01:30

Areas

Area Settings

Traffic

Volume
Vehicle Distribution: Standard
Traffic Demand: Upload File

Vehicle Types: ☑ Car ☑ Truck ☑ Bus ☐ Emergency ☑ Motorcycle ☐ Two Wheeler

Vehicle Data: ☐ Emission Data ☑ Error Codes ☐ Road Conditions
Sampling Rate: 1000

Cell Network

Download Capacity: 50
Upload Capacity: 25
Maximum Retries: 3
Loss Probaility
Minimum Delay
Maximum Delay

Environment

Road Network: Generate
Weather: Random
Environmental Conditions: ☐ Obstacles ☐ Road Damage

Figure 2. Web-based UI for modeling general and area-specific properties for a CAV scenario involving cloud computing

Figure 2 shows the UI prototype screens for our scenario modeling approach with the example of an Advanced Roadside Assistance use case. The upper screen depicts general scenario properties for a *scenario model* and provides the possibility of specifying different areas that should be simulated. For convenience, users can select areas directly via OSM without providing geolocations. Within the lower screen, properties regarding the simulation of traffic, cellular network, and environment can be specified for each area. To further support the modeling process, we introduce three different views for modeling a scenario:

- The *Simple* view abstract the domain concepts as much as possible and provide preconfigured settings, e. g. a 5G configuration for the cell network that automatically set all values regarding capacity, delay, etc.
- *Standard* provide most of the configuration possibilities depicted in the scenario metamodel, but neglects detailed settings such as the delay type model or custom vehicle types. This view will cover most scenarios and is the means of choice.

- With the *Expert* view, users need to model all attributes from the scenario metamodel, which requires the most effort but also allows us to describe scenarios most detailed and specific to the scenario requirements.

5.3. Data Serialization

While the previous section introduced a web-based UI to model cloud-based mobility services scenarios, the next step is to serialize the configuration of the scenario through a standardized data format to exchange scenarios with other stakeholders and automatically process the configuration data with software tools, e. g. to generate simulator configurations. For that purpose, we use YAML (Evans et al., 2017), which is a human-readable data serialization language that is popular for writing configuration files. YAML does not use any format symbols, such as closing tags, and is therefore easy to read and understand compared to, for example, JSON or XMI. Furthermore, it provides flexibility and good integration with other languages such as Python. Listing 1 shows an example of how the user input from the web-based UI is mapped to a YAML file. Using the metaclasses and attributes shown in Figure 1, we ensure that such a *scenario model* is in accordance with the concepts defined in the metamodel.

6. Discussion

Developers have to make many design decisions during the development of cloud-based mobility services. Such design decisions require contentious and early feedback to the development team during the development process to ensure that the software architecture design is suitable for the according scenario and meets all quality-of-service requirements. Especially in the early stages of a development process, feedback based on synthetic data provides crucial insights into potential problems with the defined software architecture or the technology used that needs to be refactored. Often testing approaches are designed for domain experts and require specific knowledge. However, future mobility will go beyond the automotive domain and involve various IoT domains with various stakeholders and different levels of expertise. As our targeted stakeholder include SMEs, cities and municipalities, or third-party service provider yet outside of the automotive domain, we want to provide an easy-to-use tooling with a lot of automation internally. By having a web-based UI with different views and predefined strategies, we can greatly abstract the complexity of the domain and provide a seamless user experience. Furthermore, it gives us flexibility to integrate different types of modern UI elements and reuse existing software tooling, e. g. OSMWebWizzard (Deepika et al., 2022). As a modeling language can comprise various concrete syntaxes, we have the flexibility to also add a textual concrete syntax in the future.

The biggest challenge for scenario modeling is to define the expressiveness of the model. On the one hand, we need to abstract the complex reality of mobility scenarios and on the other hand provide the right level of detail for an appropriate simulation setup and testing environment. Such a trade-off is important as the simulation should provide only the data that are required to test and validate a distinct scenario. For example, simulation of detailed vehicle dynamics may be irrelevant for most scenarios, but would cost valuable computation power if simulated. We are aware that our modeling approach neglects, respectively, abstract particular scenario details, especially concerning the environment, and thus cannot cover all CAV scenarios. However, our approach supports all the basic aspects that are relevant for CAVs and thus should be suitable for most CAV scenarios. Furthermore, the approach provides flexibility and will be extended in the future with additional aspects for CAV scenarios, e. g. the integration of POI similar to Reichsöllner et al. (2022).

```yaml
scenario:
    id: f8c3de3d-1fea-4d7c-a8b0-29f63c4c3454
    name: Advanced Roadside Assistance
    startTime: 2024-05-13T15:41:00
    period: 5400
area:
    nw:
        longitude: 51.5251
        latitude: 7.4391
    se:
        longitude: 51.4939
        latitude: 7.4922
    traffic:
        volume: 35
        distribution: standard
        demand: random
        samplingrate: 1000
        errorCode: true
        emissionData: false
        roadCondition: false
        vehicleTypes:
            - car
            - bus
            - truck
            - motorcycle
    environment:
        weather: random
        roadnetwork: generate
    cellnetwork:
        downlinkcapacity: 50
        uplinkcapacity: 25
        lossprobability: 10
        delaytype: random
        mindelay: 100
        maxdelay: 5000
        maxretries: 3
```

Listing 1. YAML-based data serialization and exchange format

With the scenario modeling approach proposed here, we provided the formal basis for the subsequent V&V activities described in our previous work (Heisig and

Flick, 2021). This includes the semiautomatic setup of co-simulation environments to generate all necessary test data for a specific scenario as well as testing (i) how well the system performs under different conditions, such as load, stress, and volume (performance testing); (ii) whether the system meets the specified functional requirements (functional testing); (iii) how the different system components work together (integration testing); and (iv) to which degree stakeholders' expectations are satisfied (acceptance testing). For the setup of an appropriate co-simulation environment, config generators can be used that generate a specific simulator configuration for each simulator part of the co-simulation environment. To achieve maximum flexibility for the support of various simulators, aspects of the *scenario model* should be mapped to existing standards, e. g. ASAM OpenSCENARIO (ASAM, 2022), as much as possible.

A formal scenario description helps to provide the right test data in the right granularity and omit data that are not needed. Although we discussed vehicle data within the domain ontology, they are not specified further in the metamodel. This is because our focus in this work is on formally describing the different building blocks of a CAV scenario (*scenario model*) and not on the co-simulation output, which are basically large sets of vehicle-specific data. However, the structure and attributes of vehicle data also need to be specified through a metamodel and mapped to an appropriate standard, such as VSS, to foster uniformity and reuseability throughout the testing process.

7. Conclusion

The overall objective of our work is to provide a simple and straightforward approach to evaluate cloud-based mobility services in a virtual way. As a first step towards this, we proposed a scenario modeling approach for CAVs with a particular focus on the cloud dimension. At first, we provided a domain ontology with aspects regarding traffic, vehicles, cellular networks, and environment. Based on the ontology, we specified a DSL to formally describe domain concepts, their characteristics, and interrelationships. The DSL is composed of a metamodel depicted via a UML class diagram and a web-based UI with different views that abstract domain complexity and complement our modeling approach. Using YAML as the data serialization format, scenarios can be exchanged with other stakeholders and automatically processed by software tools. In the long term, the concept proposed here helps to integrate CAVs in IoT ecosystems by allowing a proof-of-concept design of cloud-based software architectures to foster robust, accurate, and scalable applications.

For the future, we plan to evaluate our approach with different mobility scenarios and stakeholders, which may reveal additional scenario properties that we need to cover. In addition, we need to provide simulator-specific config generators that allow one to generate appropriate co-simulation environments out of a *scenario model* to produce large sets of synthetic test data from different simulated vehicles. In this regard, vehicle data will also be specified by a metamodel and mapped to VSS, if applicable, as an emerging standard for vehicle signals.

References

Ali, G. M. N., Sadat, M. N., Miah, M. S., Sharief, S. A., and Wang, Y. (2024). A comprehensive study and analysis of the third generation partnership project's 5g new radio for vehicle-to-everything communication. *Future Internet*, 16(1):21.

Alvarez-Coello, D., Wilms, D., Bekan, A., and Gómez, J. M. (2021). Towards a data-centric architecture in the automotive industry. *Procedia Computer Science*, 181:658–663.

ASAM (2022). Asam openscenario. Standard, Association for Standardization of Automation and Measuring Systems.

Bach, J., Otten, S., and Sax, E. (2016). Model based scenario specification for development and test of automated driving functions. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 1149–1155. IEEE.

Barisic, A. (2017). *Usability Evaluation of Domain-Specific Languages*. PhD thesis, Universidade NOVA de Lisboa (Portugal).

Batsch, F., Kanarachos, S., Cheah, M., Ponticelli, R., and Blundell, M. (2021). A taxonomy of validation strategies to ensure the safe operation of highly automated vehicles. *Journal of Intelligent Transportation Systems*, 26(1):14–33.

Battle, R. and Kolas, D. (2011). Geosparql: enabling a geospatial semantic web. *Semantic Web Journal*, 3(4):355–370.

Bellini, P., Benigni, M., Billero, R., Nesi, P., and Rauch, N. (2014). Km4city ontology building vs data harvesting and cleaning for smart-city services. *Journal of Visual Languages & Computing*, 25(6):827–839.

Bieker, L., Krajzewicz, D., Morra, A., Michelacci, C., and Cartolano, F. (2015). Traffic simulation for all: A real world traffic scenario from the city of bologna. In Behrisch, M. and Weber, M., editors, *Modeling Mobility with Open Data*, pages 47–60. Springer International Publishing.

Chen, W. and Kloul, L. (2018). An ontology-based approach to generate the advanced driver assistance use cases of highway traffic. In *10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management.*

Codecá, L., Frank, R., Faye, S., and Engel, T. (2017). Luxembourg sumo traffic (lust) scenario: Traffic demand evaluation. *IEEE Intelligent Transportation Systems Magazine*, 9(2):52–63.

Codecá, L. and Härri, J. (2017). Towards multimodal mobility simulation of c-its: The monaco sumo traffic scenario. In *2017 IEEE Vehicular Networking Conference (VNC)*, pages 97–100.

de Gelder, E., den Camp, O. O., and de Boer, N. (2020). Scenario categories for the assessment of automated vehicles. *CETRAN, Singapore, Version*, 1.

de Gelder, E., Paardekooper, J.-P., Saberi, A. K., Elrofai, H., den Camp, O. O., Kraines, S., Ploeg, J., and De Schutter, B. (2022). Towards an ontology for scenario definition for the assessment of automated vehicles: An object-oriented framework. *IEEE Transactions on Intelligent Vehicles*, 7(2):300–314.

Deepika, M., Shenoy, P. D., and Venugopal, K. (2022). Sumo-network: Generating and remodeling real world map using osmwebwizard and netedit. In *2022 IEEE North Karnataka Subsection Flagship International Conference (NKCon)*, pages 1–6. IEEE.

Evans, C., Ben-Kiki, O., and döt Net, I. (2017). Yaml ain't markup language (yaml™) version 1.2.

Genesereth, M. R. and Nilsson, N. J. (2012). *Logical foundations of artificial intelligence*. Morgan Kaufmann.

Grimm, D., Schindewolf, M., and Sax, E. (2023). Fleet in the loop: An open source approach for design and test of resilient vehicle architectures. In *2023 IEEE 15th International Symposium on Autonomous Decentralized System (ISADS)*, pages 1–8. IEEE.

Hallerbach, S. (2020). *Simulation-based testing of cooperative and automated vehicles*. PhD thesis, Universität Oldenburg.

Heisig, P. and Flick, C. (2021). Towards a domain-specific language for the virtual validation of cloud-native mobility services.

Irvine, P., Baker, P., Mo, Y. K., Da Costa, A. B., Zhang, X., Khastgir, S., and Jennings, P. (2022). Vehicle-to-everything (v2x) in scenarios: Extending scenario description language for connected vehicle scenario descriptions. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, pages 548–555. IEEE.

Jiang, Q., Nian, D., Guo, Y., Ahmed, M., Yang, G., and Ma, J. (2023). Evaluating connected vehicle-based weather responsive management strategies using weather-sensitive microscopic simulation. *Journal of Intelligent Transportation Systems*, 27(1):92–110.

Junietz, P., Wachenfeld, W., Klonecki, K., and Winner, H. (2018). Evaluation of different approaches to address safety validation of automated driving. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 491–496. IEEE.

Katsumi, M. and Fox, M. (2018). Ontologies for transportation research: A survey. *Transportation Research Part C: Emerging Technologies*, 89:53–82.

Klotz, B., Troncy, R., Wilms, D., and Bonnet, C. (2018). VSSo - A vehicle signal and attribute ontology. In *9th International Semantic Sensor Networks Workshop (SSN)*.

Lobo, S. C., Neumeier, S., Fernandez, E. M., and Facchi, C. (2020). Intas–the ingolstadt traffic scenario for sumo. *arXiv preprint arXiv:2011.11995*.

López Díaz, M. and Tundis, A. (2023). Combining tapas and sumo towards crises management based on traffic data. In *Proceedings of the 22nd International Conference on Modeling Applied Simulation (MAS 2023)*.

MacHardy, Z., Khan, A., Obana, K., and Iwashina, S. (2018). V2x access technologies: Regulation, research, and remaining challenges. *IEEE Communications Surveys & Tutorials*, 20(3):1858–1877.

Marosi, A. C., Lovas, R., Kisari, Á., and Simonyi, E. (2018). A novel iot platform for the era of connected cars. In *2018 IEEE international conference on future IoT technologies (Future IoT)*, pages 1–11. IEEE.

Mostefaoui, A., Merzoug, M. A., Haroun, A., Nassar, A., and Dessables, F. (2022). Big data architecture for connected vehicles: Feedback and application examples from an automotive group. *Future Generation Computer Systems*, 134:374–387.

Rahman, M. S., Abdel-Aty, M., Wang, L., and Lee, J. (2018). Understanding the highway safety benefits of different approaches of connected vehicles in reduced visibility conditions. *Transportation research record*, 2672(19):91–101.

Rapelli, M., Casetti, C., and Gagliardi, G. (2021). Vehicular traffic simulation in the city of turin from raw data. *IEEE Transactions on Mobile Computing*, 21(12):4656–4666.

Reichsöllner, E., Freymann, A., Sonntag, M., and Trautwein, I. (2022). Sumo4av: An environment to simulate scenarios for shared autonomous vehicle fleets with sumo based on openstreetmap data. In *SUMO Conference Proceedings*, volume 3, pages 83–94.

Schrab, K., Protzmann, R., and Radusch, I. (2022). A large-scale traffic scenario of berlin for evaluating smart mobility applications. In *Conference on Sustainable Urban Mobility*, pages 276–287. Springer.

Stadler, C., Lehmann, J., Höffner, K., and Auer, S. (2012). Linkedgeodata: A core for a web of spatial open data. *Semantic Web*, 3(4):333–354.

Studer, R., Benjamins, V. R., and Fensel, D. (1998). Knowledge engineering: Principles and methods. *Data & knowledge engineering*, 25(1-2):161–197.

Viktorović, M., Yang, D., and Vries, B. d. (2020). Connected traffic data ontology (ctdo) for intelligent urban traffic systems focused on connected (semi) autonomous vehicles. *Sensors*, 20(10):2961.

Yamazaki, Y., Tamura, Y., Défago, X., Javanmardi, E., and Tsukada, M. (2023). Tost: Tokyo sumo traffic scenario. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 3597–3604. IEEE.

Yazdizadeh, A. and Farooq, B. (2020). Smart mobility ontology: Current trends and future directions. *Handbook of smart cities*, pages 1–36.

Zipfl, M., Koch, N., and Zöllner, J. M. (2023). A comprehensive review on ontologies for scenario-based testing in the context of autonomous driving. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–7. IEEE.