ETSI TS 103 410-7 V1.1.1 (2020-07)

TECHNICAL SPECIFICATION

SmartM2M;
Extension to SAREF;
Part 7: Automotive Domain

Reference

DTS/SmartM2M-103410-7-SRF4AUTO

Keywords

IoT, oneM2M, ontology, SAREF, semantic, transport

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Smart Machine-to-Machine communications (SmartM2M) .

The present document is part 7 of a multi-part deliverable covering SmartM2M; Extension to SAREF, as identified below:

Part 1: "Energy Domain";

Part 2: "Environment Domain";

Part 3: "Building Domain";

Part 4: "Smart Cities Domain";

Part 5: "Industry and Manufacturing Domains";

Part 6: "Smart Agriculture and Food Chain Domain";

Part 7: "Automotive Domain";

Part 8: "eHealth/Ageing-well Domain";

Part 9: "Wearables Domain";

Part 10: "Water Domain".

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document presents SAREF4AUTO, an extension of SAREF for the Automotive Domain.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI TS 103 264 (V3.1.1) (02-2020): "SmartM2M; Smart Applications; Reference Ontology and oneM2M Mapping".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI TR 103 508 (V1.1.1) (10-2019): "SmartM2M; SAREF extension investigation; Requirements for Automotive".

[i.2] SENSORIS: "Sensor Interface Specification".

NOTE: Available at https://sensor-is.org/.

[i.3] CEN EN 16157-2:2019: "Intelligent transport systems - DATEX II data exchange specifications for traffic management and information - Part 2: Location referencing".

[i.4] "The DATEX II Parking Publications Extension".

NOTE: Available at https://datex2.eu/implementations/extension_directory/parking-publications-extension-v10a.

[i.5] OGC 11-052r4: "OGC GeoSPARQL - A Geographic Query Language for RDF Data". Version 1.0.

[i.6] ETSI TS 102 894-2: "Intelligent Transport Systems (ITS); Users and applications requirements; Part 2: Applications and facilities layer common data dictionary".

[i.7] SAE J3016™: "Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems".

[i.8] ISO/IEC 11578:1996: "Information technology -- Open Systems Interconnection -- Remote Procedure Call (RPC)".

[i.9] Recommendation ITU-T X.667: "Information technology - Procedures for the operation of object identifier registration authorities: Generation of universally unique identifiers and their use in object identifiers".

[i.10] ISO/IEC 9834-8:2014: "Information technology -- Procedures for the operation of object identifier registration authorities -- Part 8: Generation of universally unique identifiers (UUIDs) and their use in object identifiers".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms apply:

ontology: formal specification of a conceptualization, used to explicit capture the semantics of a certain reality

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AVP	Automated Valet Parking
CAN	Controller Area Network
DATEX	Data Exchange
ECU	Electronic Control Unit
GUID	Globally Unique Identifier
ITS	Intelligent Transport Systems
OBE	On Board Equipment
OWL	Ontology Web Language
OWL-DL	Ontology Web Language - Description Logics
RDF	Resource Description Framework
RDF-S	Resource Description Framework - Schema
RPC	Remote Procedure Call
SAREF	Smart Applications REFerence ontology
SAREF4AUTO	SAREF extension for Automotive
SOSA	Sensor Observation Sampling Actuator
SSN	Semantic Sensor Network
STF	Specialists Task Force
TC	Technical Committee
TR	Technical Report
TS	Technical Specification
UML	Unified Modelling Language
UUID	Universally Unique Identifier
V2V	Vehicle to Vehicle
VMS	Variable Message Systems
VRU	Vulnerable Road Users
W3C	World Wide Web Consortium
WGS 84	World Geodetic System 1984

4 SAREF4AUTO ontology and semantics

4.1 Introduction and overview

The present document has been developed in the context of the STF 566, an ETSI specialists task force that was established with the goal to extend SAREF for the domains of Automotive, eHealth/Ageing-well, Wearables and Water (https://portal.etsi.org/STF/STFs/STF-HomePages/STF566). In particular, the present document is a technical specification of SAREF4AUTO, an OWL-DL ontology that extends SAREF for the Automotive domain.

The intention of SAREF4AUTO is to connect SAREF with existing ontologies (such as W3C SSN, W3C SOSA, GeoSPARQL, etc.) and important standardization initiatives and ontologies in the Automotive domain, as mentioned in the associated SAREF4AUTO requirements document ETSI TR 103 508 [i.1], including:

- ETSI TC ITS for V2V communications.
- The SENSORIS data model [i.2] (https://sensor-is.org/) for exchanging data between vehicles and cloud services.
- The DATEX II standard [i.3] for information exchange between traffic management centres, traffic information centres and service providers.
- The DATEX II Parking Publications [i.4] for specifying information about parking sites and individual parking vehicles (https://datex2.eu/implementations/extension_directory/parking-publications-extension-v10a).

To show the potential of SAREF4AUTO, the present document focuses on three examples, which are the "platooning", "Automated Valet Parking (AVP)" and "Vehicle environment with Vulnerable Road Users (VRU)" use cases. Various other examples exist in the Automotive domain. However, it was necessary to make actionable choices within the STF 566 timeframe and the available resources, thus those have been chosen as the three initial examples to create SAREF4AUTO. As a next step, it is recommended to further refine the proposed examples to add relevant sensors that are not considered yet, and also consider additional use cases to create new releases of SAREF4AUTO, following and extending the examples provided in the present document. As all the SAREF ontologies, SAREF4AUTO is a dynamic semantic model that is meant to evolve over time. Therefore, the stakeholders in the Automotive domain are invited to use, validate and provide feedback on SAREF4AUTO, collaborating with the SAREF ontology experts to improve and evolve SAREF4AUTO in an iterative and interactive manner, so that changes and additions can be incorporated in future releases of the present document.

The prefixes and namespaces used in SAREF4AUTO and in the present document are listed in Table 1.

Table 1: Prefixes and namespaces used within the SAREF4AUTO ontology

Prefix	Namespace
s4auto	https://saref.etsi.org/saref4auto/
s4syst	https://saref.etsi.org/saref4syst/
saref	https://saref.etsi.org/core/
dc	http://purl.org/dc/elements/1.1/
dcterms	http://purl.org/dc/terms/
owl	http://www.w3.org/2002/07/owl#
om	http://www.ontology-of-units-of-measure.org/resource/om-2/
time	http://www.w3.org/2006/time#
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
sf	http://www.opengis.net/ont/sf#
xsd	http://www.w3.org/2001/XMLSchema#
geosp	http://www.opengis.net/ont/geosparql#
wgs84	http://www.w3.org/2003/01/geo/wgs84_pos#

4.2 SAREF4AUTO

4.2.1 General Overview

An overview of the SAREF4AUTO ontology is provided in Figure 1. For all the entities described in the present document, it is indicated whether they are defined in the SAREF4AUTO extension or elsewhere by the prefix included before their identifier, i.e. if the element is defined in SAREF4AUTO, the prefix is `s4auto`, while if the element is reused from another ontology, it is indicated by a prefix according to Table 1.

Arrows are used to represent properties between classes and to represent some RDF, RDF-S and OWL constructs, more precisely:

- Plain arrows with white triangles represent the `rdfs:subClassOf` relation between two classes. The origin of the arrow is the class to be declared as subclass of the class at the destination of the arrow.
- Dashed arrows between two classes indicate a local restriction in the origin class, i.e. that the object property can be instantiated between the classes in the origin and the destination of the arrow. The identifier of the object property is indicated within the arrow.
- Dashed arrows with no identifier are used to represent the `rdf:type` relation, indicating that the element in the origin of the arrow is an instance of the class in the destination of the arrow.

Datatype properties are denoted by rectangles attached to the classes, in an UML-oriented way. Dashed boxes represent local restrictions in the class, i.e. datatype properties that can be applied to the class they are attached to.

Individuals are denoted by rectangles in which the identifier is underlined.

Note that Figure 1 aims at showing a global overview of the main classes of SAREF4AUTO and their mutual relations. More details on the different parts of Figure 1 are provided from clause 4.2.2 to clause 4.2.8 of the present document.

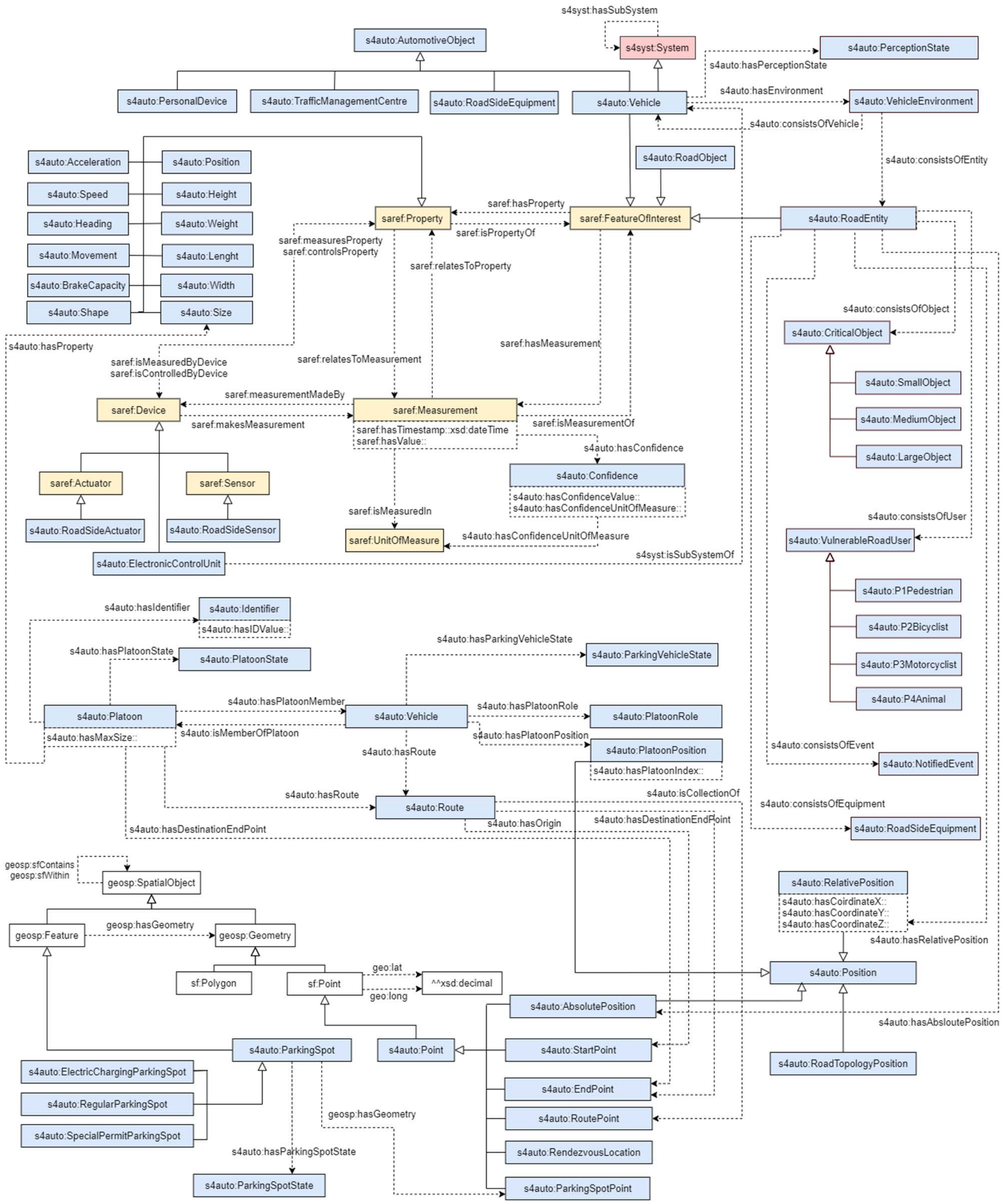

Figure 1: SAREF4AUTO overview

4.2.2 Measurement

Figure 2 shows the modelling of measurements in SAREF4AUTO. This modelling relies on the measurement model proposed in SAREF to express information about a certain property to be measured, its measured value, its measurement unit and the time of the measurement. This modelling includes the `saref:FeatureOfInterest` (whose design pattern has been taken from the W3C SSN ontology) that provides the means to refer to the real world phenomenon that is being observed in a given measurement (e.g. when a camera measures the speed of a car on the road, the car can be defined as the feature of interest of this speed measurement). The reader is referred to the SAREF specification [1] for details about the modelling of measurements, whereas the present document specifies the new concepts created for the SAREF4AUTO extension, namely the `s4auto:Confidence` class and the associated properties and subclasses that are shown in Figure 2. Examples of measurements in SAREF4AUTO are proposed in clause 4.3 of the present document.

As all measurements are subject to uncertainty, a confidence (error) value is always associated with measurement values. Therefore, SAREF4AUTO defines the `s4auto:hasConfidence` property that relates a `saref:Measurement` to its confidence `s4auto:Confidence`.

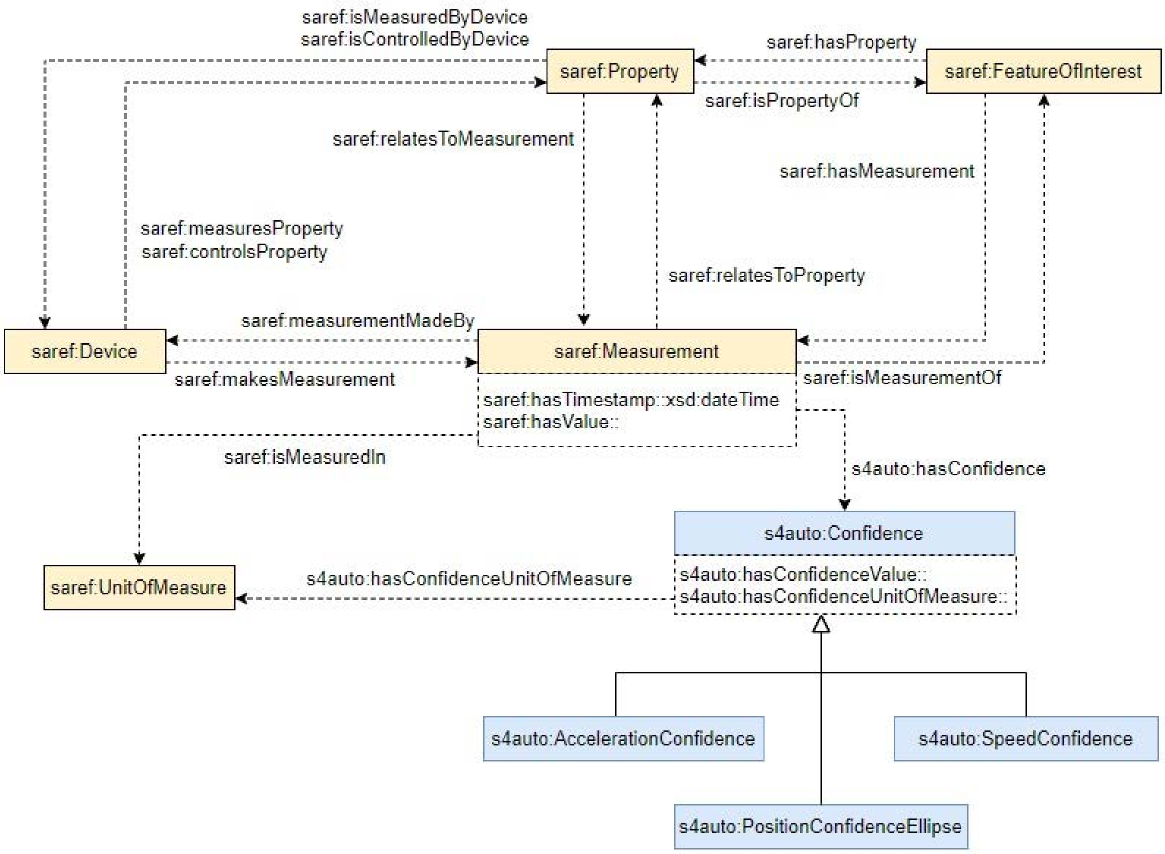

Figure 2: Measurement model

Table 2 and Table 3 summarize the definitions of the classes and properties described above.

Table 2: Measurement: class definitions

Class	Definition
s4auto:Confidence	Every measurement in the automotive domain is an estimated measurement coming from the aggregation of several measurements from different sensors or devices taken at the same time or at consecutive times very close to each other. Therefore, a confidence is always associated to each measurement to indicate that it has not 100 % accuracy.
s4auto:AccelerationConfidence	A type of confidence for acceleration measurements that represents the absolute accuracy of a reported vehicle acceleration value with a predefined confidence level (e.g. 95 %).
s4auto:PositionConfidenceEllipse	A type of confidence that represents horizontal position accuracy in a shape of ellipse with a predefined confidence level (e.g. 95 %). The centre of the ellipse shape corresponds to the reference position point for which the position accuracy is evaluated.
s4auto:SpeedConfidence	A type of confidence for speed measurements that represents absolute accuracy of a speed value in cm/s.

Table 3: Measurement: property definitions

Property	Definition
s4auto:hasConfidence	A relation between an estimated measurement (saref:Measurement class) and its confidence (s4auto:Confidence).
s4auto:hasConfidenceValue	A relation between the confidence (s4auto:Confidence) and its value, which can be a discrete value (±5 cm/s) or a level (e.g. 95 %).
s4auto:hasConfidenceUnitOfMeasure	A relation that allows to specify the unit of measure associated with a certain confidence.

4.2.3 Automotive Object

To define the main objects of interest in the automotive domain, the `s4auto:AutomotiveObject` class has been created. This class includes vehicles (such as cars, trucks, public transport and two-wheeler), personal devices that are used by pedestrians or the riders of a two-wheeler, traffic management centres that provide services to road users, and roadside equipment (i.e. roadside devices used for the communication between connected vehicles and between connected vehicles and the infrastructure). Figure 3 shows the hierarchy of automotive objects in SAREF4AUTO.

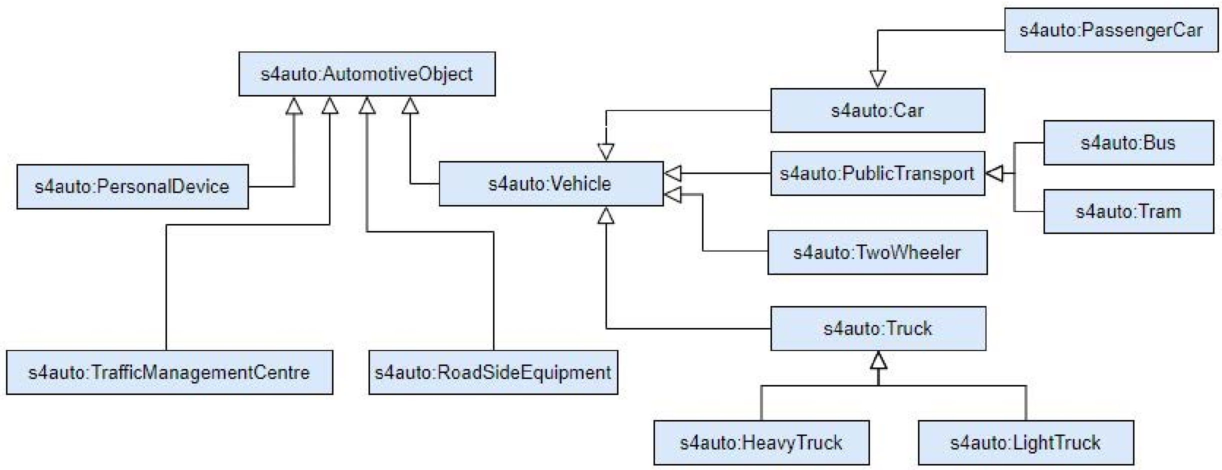

Figure 3: Automotive Object hierarchy

Note that the hierarchy of automotive objects shown in Figure 3 is not exhaustive. It is possible and recommended to further specialize it, if needed, and extend it for new and additional use cases, to be included in future versions of the present document. Table 4 summarizes the definitions of the classes described above and Table 5 focuses on the properties of the Vehicle class.

Table 4: Automotive Objects: class definitions

Class	Definition
s4auto:AutomotiveObject	The main objects of interest in the automotive domain.
s4auto:RoadSideEquipment	A roadside equipment can be either a simple data relay between vehicles and infrastructure or between the vehicles themselves, or it can run applications. It is usually high mounted in existing equipment, such as a street light, city traffic light, VMS or highway sign, or has its own pole.
s4auto:PersonalDevice	A personal device is a nomadic device used by a pedestrian or the rider of a two-wheeler. A personal device in modelled as a different type of automotive object, as it normally does not have access to the vehicle CAN bus.
s4auto:TrafficManagementCentre	A traffic centre provides services to road users: exploiting data for road traffic management, providing information about mobility zones, operational management of the road infrastructure, etc.
s4auto:Vehicle	A vehicle is the main object of interest in the automotive domain. Example of types of vehicles are car, truck, public transport and two-wheeler. See ETSI TS 102 894-2 [i.6] specification for full list of vehicles, which are type of "stations". See also the SENSORIS data model for possible type of vehicles. The main IoT device considered in the vehicle is the ECU (part of the vehicle On Board Equipment, or OBE), see s4auto:ElectronicControlUnit class in clause 4.2.4.
s4auto:Car	A type of vehicle. Example of a car is a passenger car.
s4auto:PublicTransport	A type of vehicle. Example of public transport is a bus or a tram.
s4auto:Truck	A type of vehicle. Example of trucks are heavy trucks or light trucks.
s4auto:TwoWheeler	A type of vehicle. This class involves all sorts of two wheelers, from motorcycles to bicycles and (e-)scooters.

Table 5: Vehicle: property definitions

Property	Definition
s4auto:hasIdentifier	A relation between an entity and its identifier (s4auto:Identifier) which is used to identify: 1) a vehicle (station) during data exchange with other devices/stations; 2) a platoon during data exchange with other devices/stations; 3) a parking spot within a parking area during data exchange with other devices/stations. In all cases, this identifier may be a pseudonym. It may change over space and/or over time.
s4auto:hasHeight	A relation to express the height of an entity, e.g. a vehicle.
s4auto:hasLength	A relation to express the length of an entity, e.g. a vehicle.
s4auto:hasWidth	A relation to express the width of an entity, e.g. a vehicle.
s4auto:hasVehicleRole	A relation between Vehicle and VehicleRole classes, to express the role the vehicle plays in traffic. If not specified, it is assumed that it takes a default value. Otherwise, possible values to be specified are: publicTransport, specialTransport, dangerousGoods, roadWork, rescue, emergency, safetyCar, agriculture, commercial, military, roadOperator, taxi (see ETSI TR 103 508 [i.1]).

Property	Definition
s4syst:hasSubSystem	From SAREF4SYST: Links a system to one of its sub systems. In SAREF4AUTO, a relation to express that the vehicle has a subsystem called 'ElectronicControlUnit' (ECU) that provides measurements about relevant properties, such as speed, acceleration, position, etc. See clause 4.2.4.
s4auto:hasAutomationLevel	A relation to define the level of automation of a singular vehicle. According to SAE J3016 [i.7], there are five levels of automation: • Level 0: No Driving Automation. • Level 1: Driver assistance. • Level 2: Partial driving automation. • Level 3: Conditional driving automation. • Level 4: High driving automation. • Level 5: Full Driving automation.
s4auto:hasAbsolutePosition	A relation between Vehicle and AbsolutePosition to represent the absolute position of a vehicle.
s4auto:hasRelativePosition	A relation between Vehicle and RelativePosition to represent the relative position of a vehicle (e.g. a car is x, y, z (delta-coordinates) in front of another car).
s4auto:hasRoadTopologyPosition	A relation between Vehicle and RoadTopologyPosition to represent the road topology position of a vehicle
s4auto:hasDestinationAddress	A relation to express the destination of a vehicle in terms of an Address.
s4auto:hasDestinationEndPoint	A relation to express the destination of a vehicle, a platoon or a route in terms of an end point with latitude and longitude.
s4auto:hasEnvironment	A relation that allows to describe the environment of a vehicle. See clause 4.2.8.
s4auto:hasPerceptionState	A relation to express the state of a vehicle related to its environment. See clause 4.2.8.
s4auto:hasRoute	A relation to express the route of an entity, e.g. a vehicle or a platoon.
s4auto:isMemberOfVehicleEnvironment	A relation to express that an entity, e.g. a vehicle, can be member of another vehicle's environment.
s4auto:isMemberOfPlatoon	A relation to express that a vehicle can be member of a platoon.
s4auto:hasPlatoonPosition	A relation to express that a vehicle member of a platoon has a platoon position that is defined as the index of the vehicle in the platoon starting from zero (leader) up to N (trailing vehicle).
s4auto:hasPlatoonRole	A relation to specify that in a platoon a vehicle can assume roles such as: unknown, leader, follower, ready-for-leading, trailing, etc. See clause 4.2.6.
s4auto:hasPlatoonVehicleState	A relation to express the state of a vehicle in a platoon. See clause 4.2.6.
s4auto:hasEstimatedRendezvousLocation	A relation to express that, during the forming state, a vehicle member of a platoon is given an estimated rendezvous location for joining the platoon. It is expressed in global coordinates (lat, long, alt), e.g. according to WGS 84.
s4auto:hasEstimatedJoiningTime	A relation to express that, during the forming state, a vehicle member of a platoon is given an estimated time for joining.
s4auto:hasBrakeCapacity	A relation to express that the vehicle has the capacity to reduce its speed (by action on the brake pedal or an autonomous actuator).
s4auto:hasParkingVehicleState	A relation to express the state of a vehicle that is parking.

4.2.4 Device and System

As shown in Figure 4, the `saref:Device` class has been extended in SAREF4AUTO with some specific types of sensors and actuators, i.e. roadside sensors and actuators that can measure and/or control a number of additional properties compared to the SAREF core ontology, such as speed, heading and movement.

As further shown in Figure 4, the main IoT device considered in a vehicle is the Electronic Control Unit (ECU), which is part of the vehicle on board equipment, or OBE. The ECU is defined in SAREF4AUTO as a subsystem of the vehicle by reusing the `s4syst:System` class of the SAREF4SYST extension and its properties `s4syst:hasSystem` and `s4syst:isSubSystemOf`. The ECU uses various and different sensors and devices in the vehicle to provide measurements and control properties, but is out of the scope of SAREF4AUTO to go into the details of all of these devices. It is therefore assumed that the ECU is the device providing measurements and controlling relevant properties in and about the vehicle.

Other relevant types of device are the personal devices that are used by pedestrians or the riders of a two-wheeler. Personal devices may be used for example by passenger car or truck drivers, but as they are considered as devices different from the vehicle ECU.

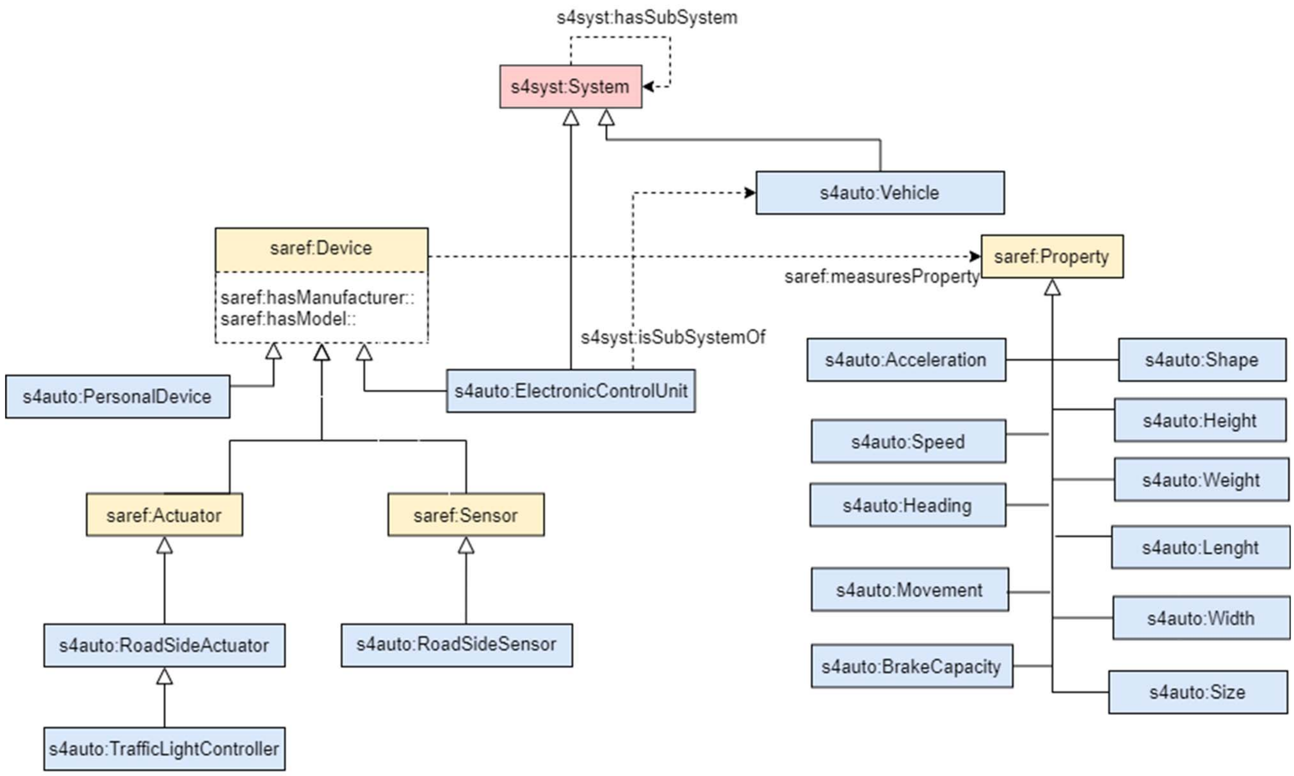

Figure 4: Device model

Table 6 and Table 7 summarize the definitions of the classes and properties described above.

Table 6: Device: class definitions

Class	Definition
s4auto:ElectronicControlUnit	An Electronic Control Unit (ECU) is any embedded system in automotive electronics that monitors and controls one or more of the electrical systems or subsystems in a vehicle. The ECU provides measurements about relevant properties of the vehicle, such as speed, acceleration, position, etc.
s4auto:RoadSideActuator	Devices deployed along the road that produce signals based on roadside sensor data measurements. Types of roadside actuators that are relevant are devices such as traffic light controllers changing from red to green based on detected vehicles.

Class	Definition
s4auto:TrafficLightController	A device that automatically operates coloured lights, typically red, amber, and green, for controlling traffic at road junctions, pedestrian crossings, and roundabouts.
s4auto:RoadSideSensor	Sensors such as cameras, radars that are deployed along the road (e.g. at intersections and at poles along a highway). Roadside sensors perform measurements, such as speed, position (relative or absolute), dimension, direction, etc. of road objects.
s4auto:PersonalDevice	A personal device is a nomadic device used by a pedestrian or the rider of a two-wheeler. A personal device in modelled as a different type of automotive object, as it normally does not have access to the vehicle CAN bus (see clause 4.2.3).

Table 7: Device: property definitions

Property	Definition
s4auto:detectsPosition	A relationship specifying the position (absolute or relative position) that can be detected by a certain device.
s4auto:usesMeasurement	A relation to express that a device can use different types of measurements, such as speed, position of detected road objects (e.g. bicycles, vehicles). This applies in particular to the ElectronicControlUnit and TrafficLightController classes.
s4syst:isSubSystemOf	From SAREF4SYST: Links a system to its super system. Properties of subsystems somehow contribute to the properties of the super system. The exact meaning of "contribute" is defined by sub properties of s4syst:subSystemOf. Property s4syst:subSystemOf is transitive. In SAREF4AUTO, a relation to express that the Electronic Control Unit (ECU) is a subsystem of the vehicle.

4.2.5 Property and Feature of Interest

Figure 5 recalls the device model that was presented in clause 4.2.4 of the present document and provides the details of the `saref:Property` class. It shows the subclasses of `saref:FeatureOfInterest` class, which are `s4auto:Vehicle`, `s4auto:RoadEntity` and `s4auto:RoadObject`. It also depicts the sub-classes of `s4auto:Position` (i.e. `s4auto:AbsolutePosition`, `s4auto:RelativePosition`, `s4auto:PlatoonPosition`, `s4auto:RoadTopologyPosition` and the possible types of position relationships, for example of the `s4auto:Vehicle` feature of interest.

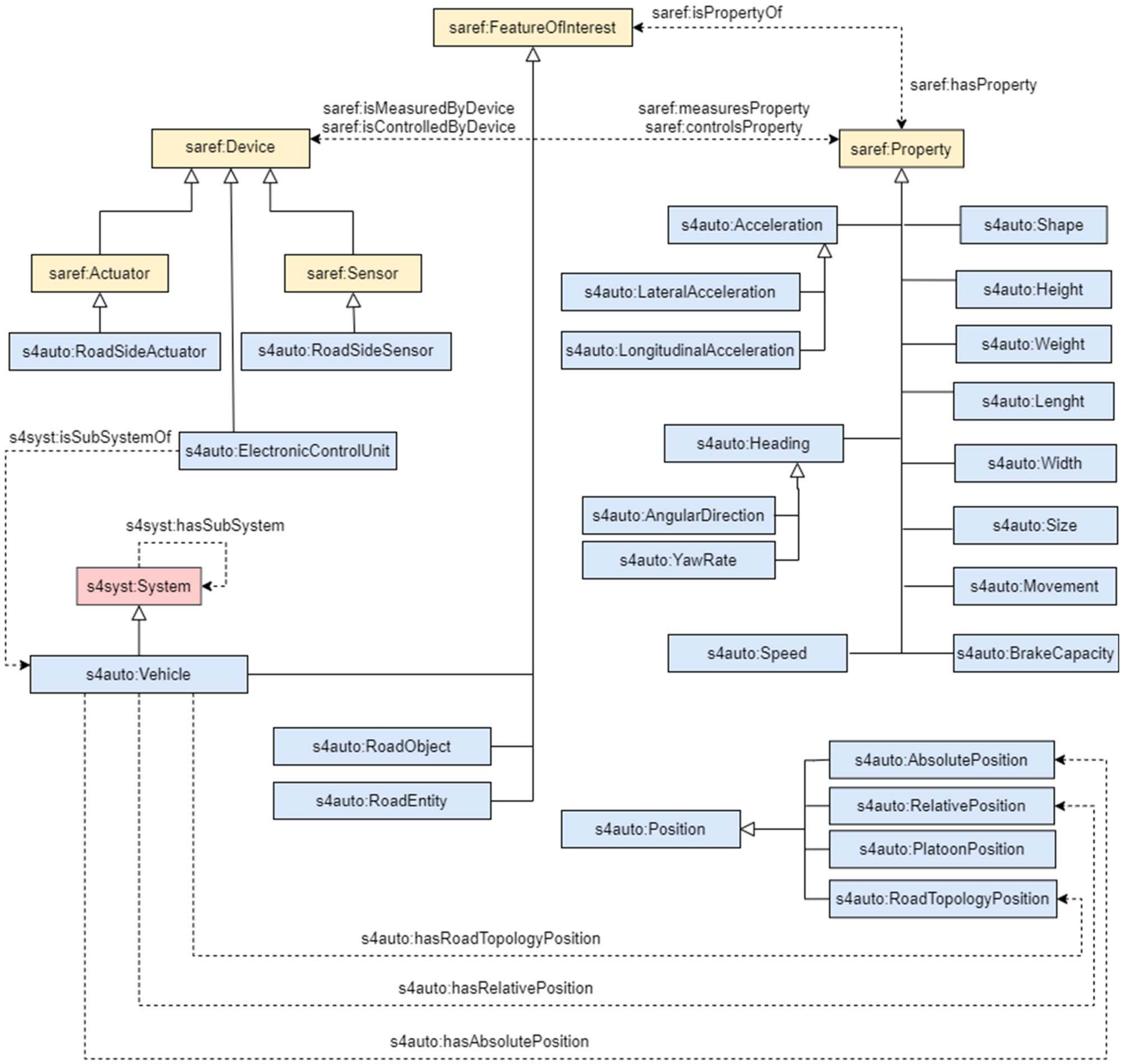

Figure 5: Property and Feature of Interest hierarchy

4.2.6 Platoon

Figure 6 shows the modelling of a platoon, which is composed by some members of type `s4auto:Vehicle` that automatically follow one another at a relatively close distance. Vehicles in the platoon have different roles (`s4auto:hasPlatoonRole` property), such as, for example, `s4auto:leader`, `s4auto:follower` or `s4auto:trailing`. The platoon can be characterized by different states over time (`s4auto:hasPlatoonState` property), such as, for example, `s4auto:assembling` or `s4auto:platooning`. The size of a platoon changes dynamically depending on the composition of the platoon over time (the `s4auto:Size` class). The property (`s4auto:hasMaxSize`) defines a fixed maximum size of vehicles that can compose a platoon at a particular moment in time. Figure 6 further shows that the platoon follows a route with a destination end point (which is usually the same destination of the platoon's leader). An example of a platoon defined according to this model is proposed in clause 4.3.1 of the present document.

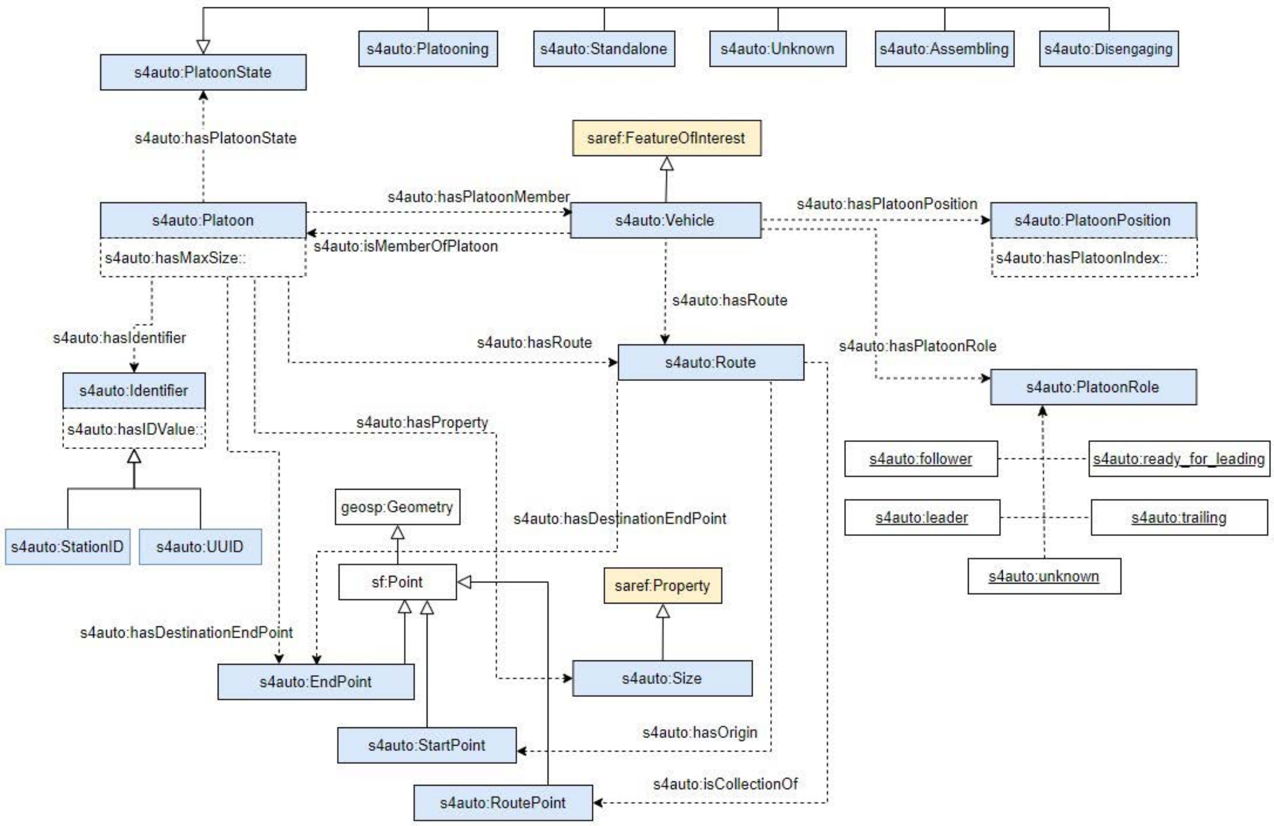

Figure 6: Platoon model

Table 8 and Table 9 summarize the definitions of the classes and properties of platoon that are described above.

Table 8: Platoon: class definitions

Class	Definition
s4auto:Platoon	A group of vehicles automatically following each other at a relatively close distance.
s4auto:Identifier	A class used to identify: 1) a vehicle (station) during data exchange with other devices/stations; 2) a platoon during data exchange with other devices/stations; 3) a parking spot within a parking area during data exchange with other devices/stations. In all cases, this identifier may be a pseudonym. It may change over space and/or over time.
s4auto:StationID	A type of identifier for vehicles and platoons, as defined in ETSI TS 102 894-2 [i.6].
s4auto:UUID	A Universally Unique Identifier (UUID) is a 128-bit number used to identify items and is also known as: Globally Unique Identifier (GUID). In its canonical textual representation, the sixteen octets of a UUID are represented as 32 hexadecimal (base 16) digits, displayed in five groups separated by hyphens, in the form 8-4-4-4-12 for a total of 36 characters (32 alphanumeric characters and four hyphens). UUID are documented in ISO/IEC 11578:1996 [i.8] and in Recommendation ITU-T X.667 [i.9] and ISO/IEC 9834-8:2014 [i.10].
s4auto:PlatoonPosition	A vehicle member of a platoon has a platoon position that is defined as the index of the vehicle in the platoon starting from zero (leader) up to N (trailing vehicle).
s4auto:PlatoonRole	In platoon a vehicle can assume roles such as: unknown, leader, follower, ready-for-leading, trailing, etc.

Class	Definition
s4auto:PlatoonState	The state of a platoon. Examples of states for a platoon are: unknown, standalone, assembling, platooning, disengaging, etc.
s4auto:Size	A Property of interest for the automotive domain. A platoon has a size. This size is defined as: 1) the total number of vehicles currently in the platoon; or 2) length of the platoon (sum of all vehicles' length + inter-vehicle distance) via s4auto: Length class.
s4auto: Length	A Property of interest for the automotive domain. In a platoon, a property to express that a platoon has a size. This size is defined as: 1) the total number of vehicles currently in the platoon (via s4auto:Size class); or 2) length of the platoon (sum of all vehicles' length + inter-vehicle distance).
s4auto:Route	The route of a vehicle or of a platoon. The route of a platoon is defined as the route of the current vehicle leader in the platoon that other follower vehicles (partially) share.
s4auto:StartPoint	The origin of the route.
s4auto:EndPoint	The final destination of a vehicle or a platoon, or the final point of a route.
s4auto:RoutePoint	The intermediate points of a route.

Table 9: Platoon: property definitions

Property	Definition
s4auto:hasIdentifier	A relation between an entity and its identifier (s4auto:Identifier) which is used to identify: 1) a vehicle (station) during data exchange with other devices/stations; 2) a platoon during data exchange with other devices/stations; 3) a parking spot within a parking area during data exchange with other devices/stations. In all cases, this identifier may be a pseudonym. It may change over space and/or over time.
s4auto:hasPlatoonMember	A relation to express that a platoon can have vehicles as its members.
s4auto:hasPlatoonState	A relation to express the state of a platoon.
s4auto:hasSize	A relation to express that a platoon has a size. This size is defined as: 1) the total number of vehicles currently in the platoon; or 2) length of the platoon (sum of all vehicles' length + inter-vehicle distance) via s4auto:hasLength property.
s4auto:hasLength	A relation to express the length of an entity, e.g. a vehicle or a platoon (Length of the platoon is the sum of all vehicles' length + inter-vehicle distance).
s4auto:hasMaxSize	A relation to express the max size of a platoon. Note that it is defined as a datatype property as this is a fixed/static value of the platoon that does not change over time. In contrast, the current size of the platoon is actually a measurement that may change over time and it is therefore defined as an object property (see s4auto:hasSize).
s4auto:hasOrigin	A relation to express the origin of an entity (e.g. a vehicle or a route) in terms of a StartPoint.
s4auto:hasDestinationEndPoint	A relation to express the destination of a vehicle, a platoon or a route in terms of an end point with latitude and longitude.

Property	Definition
s4auto:hasRoute	A relation to express the route of an entity, e.g. a vehicle or a platoon.
s4auto:isCollectionOf	A relation to express the intermediate destination of a route (e.g. of a vehicle or a platoon) in terms of RoutePoint.

4.2.7 Topology

In order to represent the parking topology for the automotive domain, the GeoSPARQL ontology has been reused. As shown in Figure 7, for representing spatial objects the `geosp:SpatialObject` class from GeoSPARQL has been reused along with its subclasses that allow defining spatial features (`geosp:Feature`) and geometries (`geosp:Geometry`). Different properties from GeoSPARQL can be reused to define spatial relations among spatial objects (e.g. `geosp:sfContains`, or `geosp:sfWithin)`or to define the geometry of a feature (`geosp:hasGeometry`). Two types of geometries from the GeoSPARQL Simple Features ontology are proposed to be used: points (`sf:Point`) and polygons (`sf:Polygon`), although others may also be used from that same ontology or from another one.

The topology model refers to the GeoSPARQL standard [i.5] for further details on how to define the parking topology (and possibly also the road topology) for the automotive domain and to the example instantiations in clause 4.3.2.

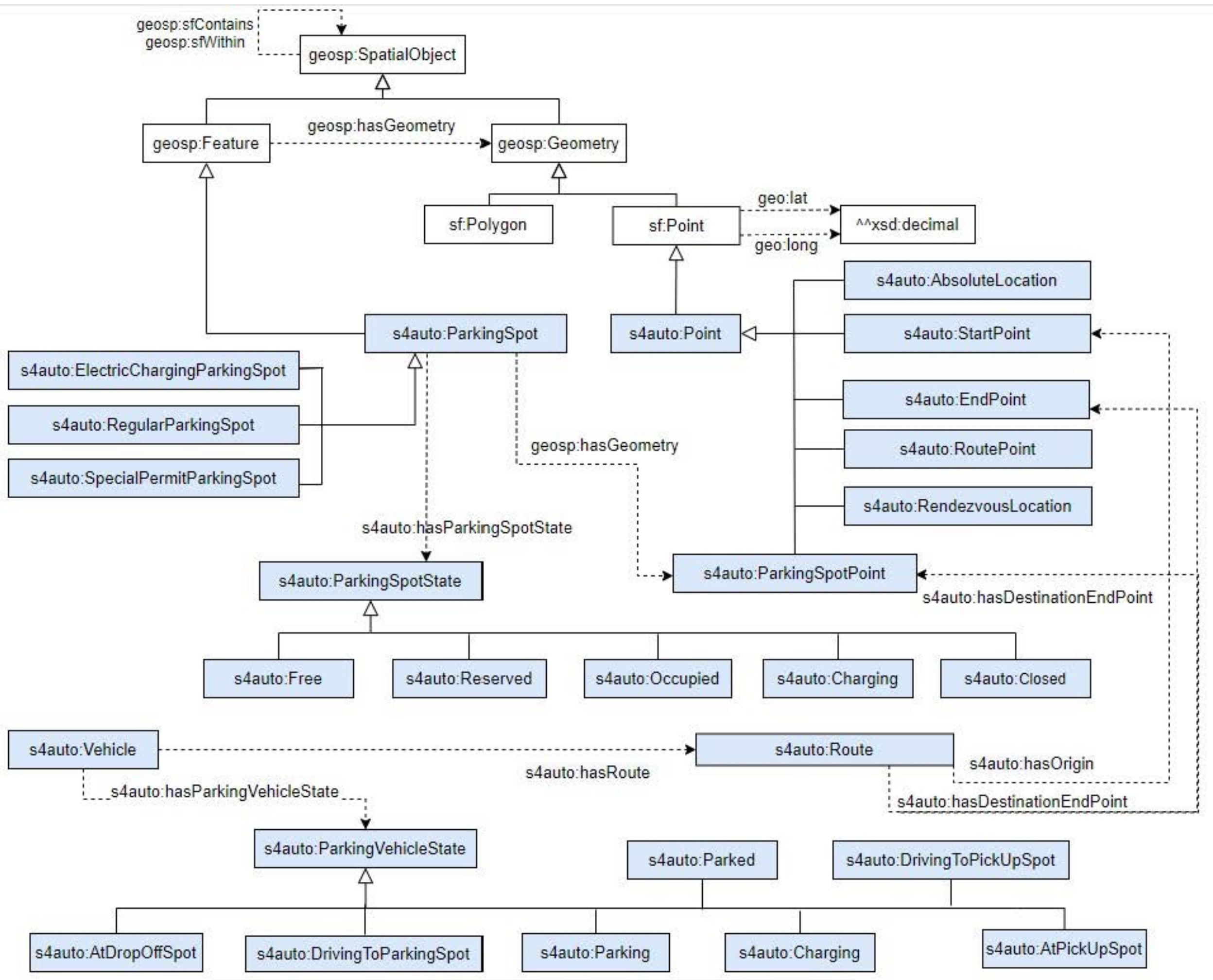

Figure 7: Topology model

4.2.8 Vehicle environment

SAREF4AUTO allows defining the `s4auto:VehicleEnvironment` class which describes the environment of a vehicle on the road. This environment consists of other vehicles (`s4auto:Vehicle`), as well as elements of the `s4auto:RoadEntity` class.

The capacity to perceive the environment is reflected in the `s4auto:PerceptionState` class.

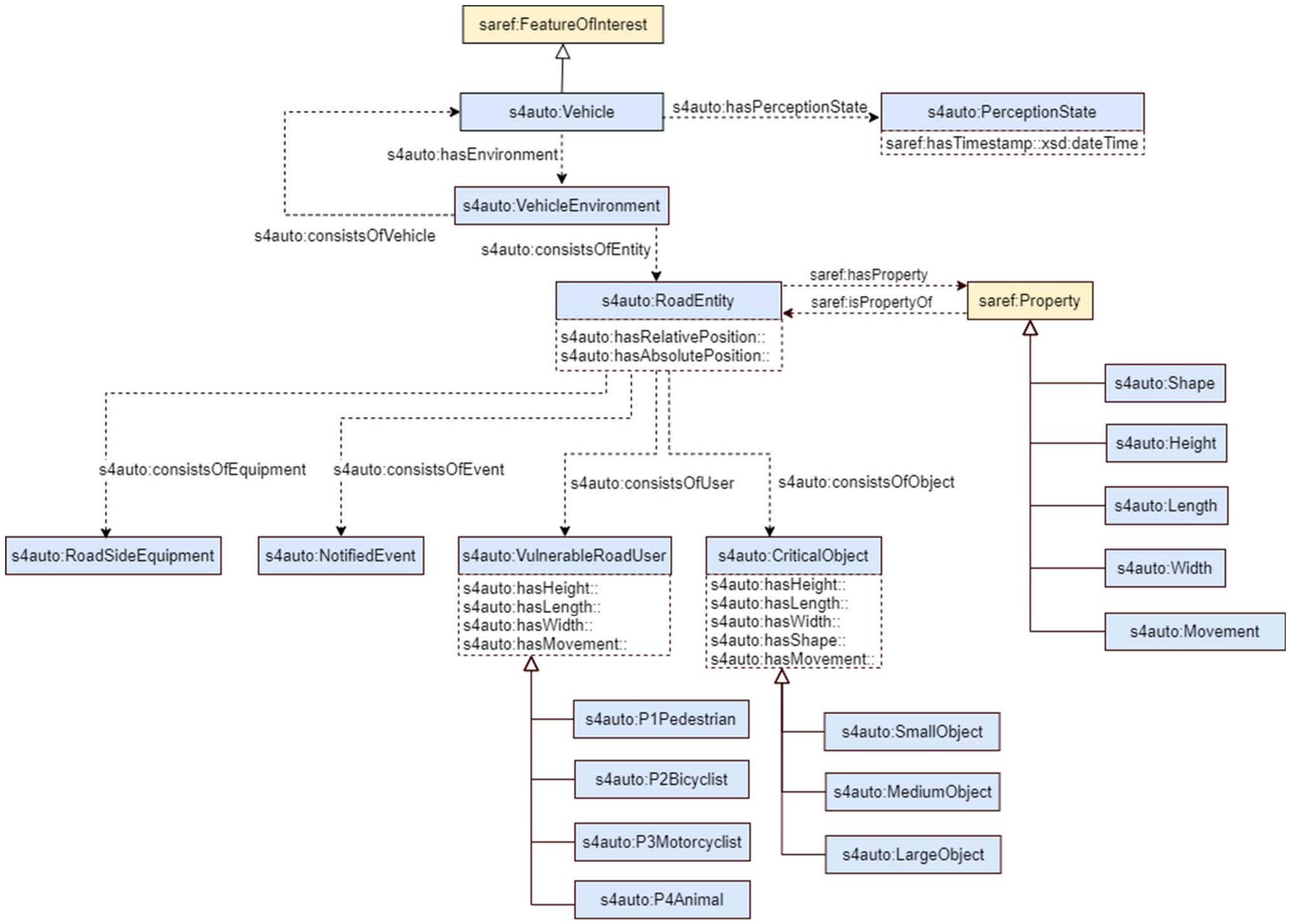

Figure 8: Vehicle environment model

Table 10 and Table 11 summarize the class and properties that characterize the `s4auto:RoadEntity` class. For the `saref:Property` class, see clause 4.2.5 of the present document and Table 12 for the properties specific to the vehicle environment (i.e. shape and movement).

Table 10: Vehicle environment: class definitions

Class	Definition
s4auto:PerceptionState	A state of the vehicle related to its environment.
s4auto:VehicleEnvironment	The vehicle environment is made of entities which are present on the road in the area around the vehicle.
s4auto:RoadEntity	An entity present on the road in the neighbouring area of a vehicle.
s4auto:VulnerableRoadUser	A user more vulnerable to collision, e.g. pedestrian, cyclist or motorcyclist.
s4auto:CriticalObject	A critical object located on the road. Critical objects are small object, medium object, large object.
s4auto:NotifiedEvent	A traffic event such as slippery road notified to the near-by vehicles.
s4auto:RoadSideEquipment	The road-side equipment definition has been provided in clause 4.2.3. It is reused here in the context of a vehicle environment.

Class	Definition
s4auto:P1Pedestrian	VRU Profile 1 users, for example: adult, child, elderly person, pram, animal, blind person guided by a dog, rider off its bike.
s4auto:P2Bicyclist	VRU Profile 2 users, for example: user of bicycle, wheelchair, skater, scooter, Segway, or a mounted horse rider.
s4auto:P3Motorcyclist	VRU Profile 3 users are Motorcyclists, which are equipped with engines that allow them to move on the road. This class includes users (driver and passengers, e.g. children and animals) of Powered Two Wheelers (PTW) such as mopeds (motorized scooters), motorcycles or side-cars.
s4auto:P4Animal	VRU Profile 4 animals presenting a safety risk to other road users, for example : dogs, wild animals, horses, cows, sheep, kangaroos, etc.
s4auto:SmallObject	A type of Critical Object.
s4auto:MediumObject	A type of Critical Object.
s4auto:LargeObject	A type of Critical Object.

Table 11: Vehicle environment: property definitions

Property	Definition
s4auto:hasEnvironment	A relation that allows to describe the environment of a vehicle.
s4auto:hasPerceptionState	A relation to express the state of a vehicle related to its environment.
s4auto:consistsOfEntity	A relation to describe the entities that can compose the vehicle environment.
s4auto:consistsOfVehicle	A relation to describe the vehicles that can compose the vehicle environment.
s4auto:consistsOfEquipment	A relation to describe the roadside equipment that can compose the vehicle environment.
s4auto:consistsOfEvent	A relation to describe the events that can compose the vehicle environment.
s4auto:consistsOfObject	A relation to describe the objects that can compose the vehicle environment.
s4auto:consistsOfUser	A relation to describe the road users that can compose the vehicle environment.
s4auto:hasRelativePosition	A relation between an entity, e.g. a vehicle, or road entity, and RelativePosition to represent its relative position (e.g. a car is x, y, z (delta-coordinates) in front of another car).
s4auto:hasAbsolutePosition	A relation between an entity, e.g. a vehicle or road entity, and AbsolutePosition to represent its absolute position
s4auto:hasHeight	A relation to express the height of an entity, e.g. a vehicle or an entity in the vehicle environment.
s4auto:hasLength	A relation to express the length of an entity, e.g. a vehicle or an entity in the vehicle environment or a platoon (Length of the platoon is the sum of all vehicles' length + inter-vehicle distance).
s4auto:hasWidth	A relation to express the width of an entity, e.g. a vehicle or an entity in the vehicle environment.
s4watr:hasShape	A relation to express the shape of an entity in the vehicle environment.
s4auto:hasMovement	A relation to express the movement of an entity in the vehicle environment.

Table 12: Sub-classes of saref:Property specific to the vehicle environment

Class	Definition
s4auto:Shape	A Property of interest for the automotive domain. It defines the overall shape of a s4auto:CriticalObject entity, i.e.: sphere, torus, cylinder, cone, ellipsoid, cube, cuboid, pyramid, prism and multiple shapes.
s4auto:Movement	A Property of interest for the automotive domain. Relevant types of movement sensed at the reference vehicle are: Static (position only), moving in same direction (speed, acceleration), moving in reverse direction (speed acceleration), crossing (speed, acceleration and direction).

4.3 Instantiating SAREF4AUTO

4.3.1 Platoon

Clause 4.3.1 of the present document shows an example of how to instantiate the SAREF4AUTO extension for the platooning use case. Platooning is a use case where a vehicle is automatically following another vehicle at a relatively close distance. Driving in a platoon requires vehicles to use intervehicle communications to anticipate timely on manoeuvres of other vehicles in the platoon. The example describes a platoon, its properties and states. This platoon consists of three trucks with specific characteristics and roles.

This example instantiation is referred to using the *ex:* prefix. This prefix is different from the `s4auto:` prefix, which indicates the SAREF4AUTO ontology on which the *ex:* example instantiation is built upon.

Figure 9 shows an instance of a platoon, namely *ex:Platoon1*, which is composed by three members, namely *ex:Truck1*, *ex:Truck2* and *ex:Truck3*. These trucks have different roles in the platoon, which are, respectively, `s4auto:leader`, `s4auto:follower` and `s4auto:trailing`. Note that these roles have the `s4auto:` prefix since they are pre-defined in the SAREF4AUTO ontology. In contrast, Platoon1, Truck1, Truck2 and Truck 3 have the *ex:* prefix since they are defined as a separate example instantiation. The platoon can be characterized by different states over time. For example, at *18:48:15* of *2020-02-02* this state is unknown (*ex:Platoon1Unknown* instance), whilst later in the same day at *19:01:00*, *19:04:50* and *19:16:38*, the platoon is in the platooning state (*ex:Platoon1Platooning* instance). The platoon has also a fixed maximum size of vehicles that can compose it (`s4auto:hasMaxSize` property) that in this example has a value of *6*, therefore, not more than 6 vehicles at a time are allowed to join *ex:Platoon1*. The platoon also has a size that changes dynamically, depending on the composition of the platoon over time. This is expressed in Figure 9 by the *ex:Platoon1Size* instance of the `s4auto:Size` class that is associated with the measurement of how many members compose the platoon at a certain moment (*ex:Platoon1Size_Meas1* instance of the `saref:Measurement` class). For example, at *18:45:35* of *2020-02-02*, the size of the platoon has a value of *2*. Figure 9 finally shows that the destination of the platoon (*ex:Platoon1EndPoint* instance) is the same destination of its leader (*ex:Truck1EndPoint* instance), which in this case is Truck1.

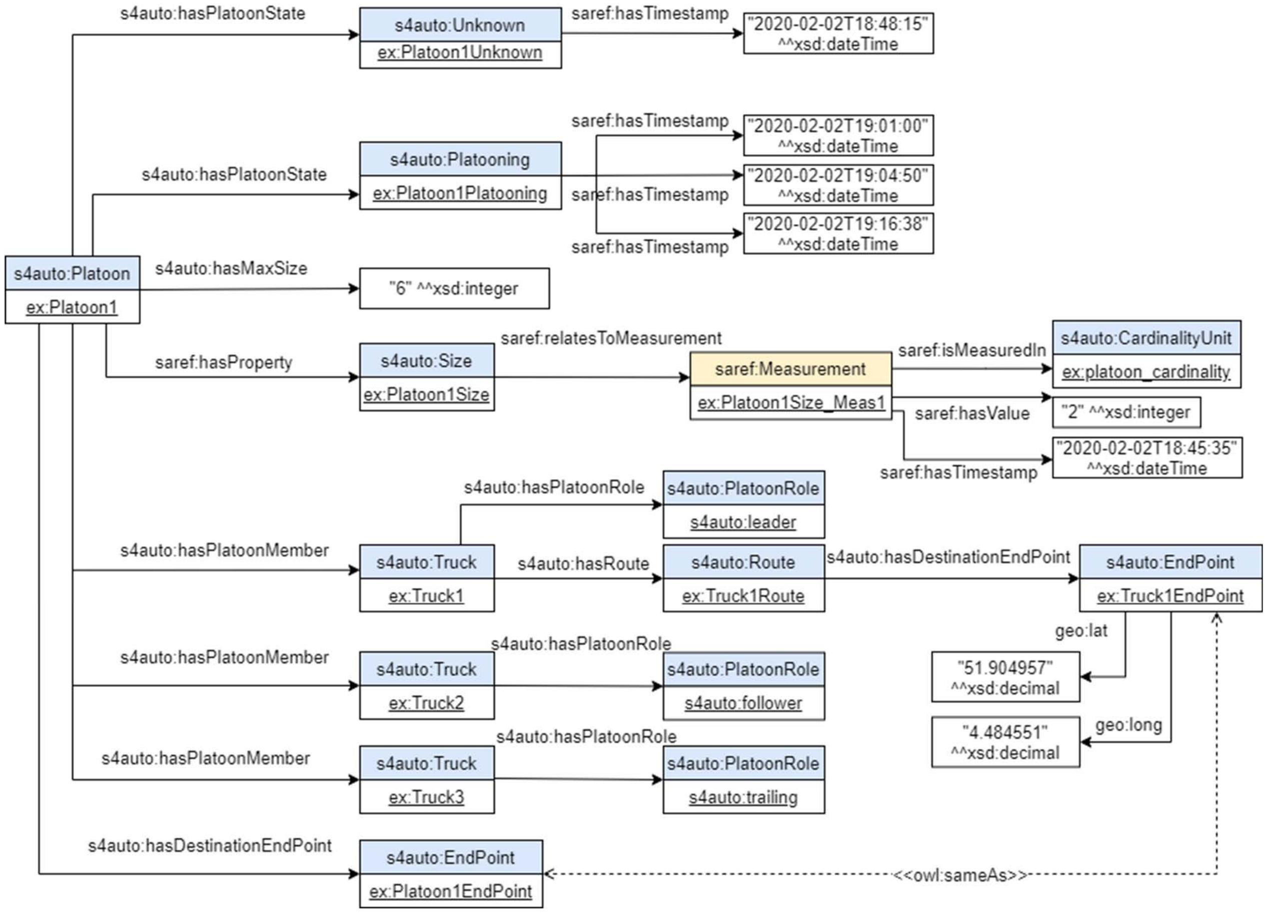

Figure 9: Platoon example

Figure 10 elaborates on the three trucks that are members of the *ex:Platoon1* instance. These trucks are all characterized by an identifier (`s4auto:hasIdentifier` property), which is defined as a Station ID (*ex:StationID4*, *ex:StationID5* and *ex:StationID6* instances). Furthermore, all trucks have a certain role and position in the platoon. For example, Truck1 has the leading role and, consequently, it assumes the first position in the platoon. This is expressed in Figure 10 by the `s4auto:hasPlatoonPosition` property, which relates the *ex:Truck1* instance with the *ex:PlatoonPosition1* instance, which in turn has a `s4auto:hasPlatoonIndex` property value of *1*. The *ex:Truck2* instance has a follower role and consequently relates to the *ex:PlatoonPosition2* instance with a `s4auto:hasPlatoonIndex` property value of *2*. Since the *ex:Truck3* instance has a trailing role, it occupies the last position in the platoon (*ex:PlatoonPosition3* instance) with a `s4auto:hasPlatoonIndex` property value of *3*.

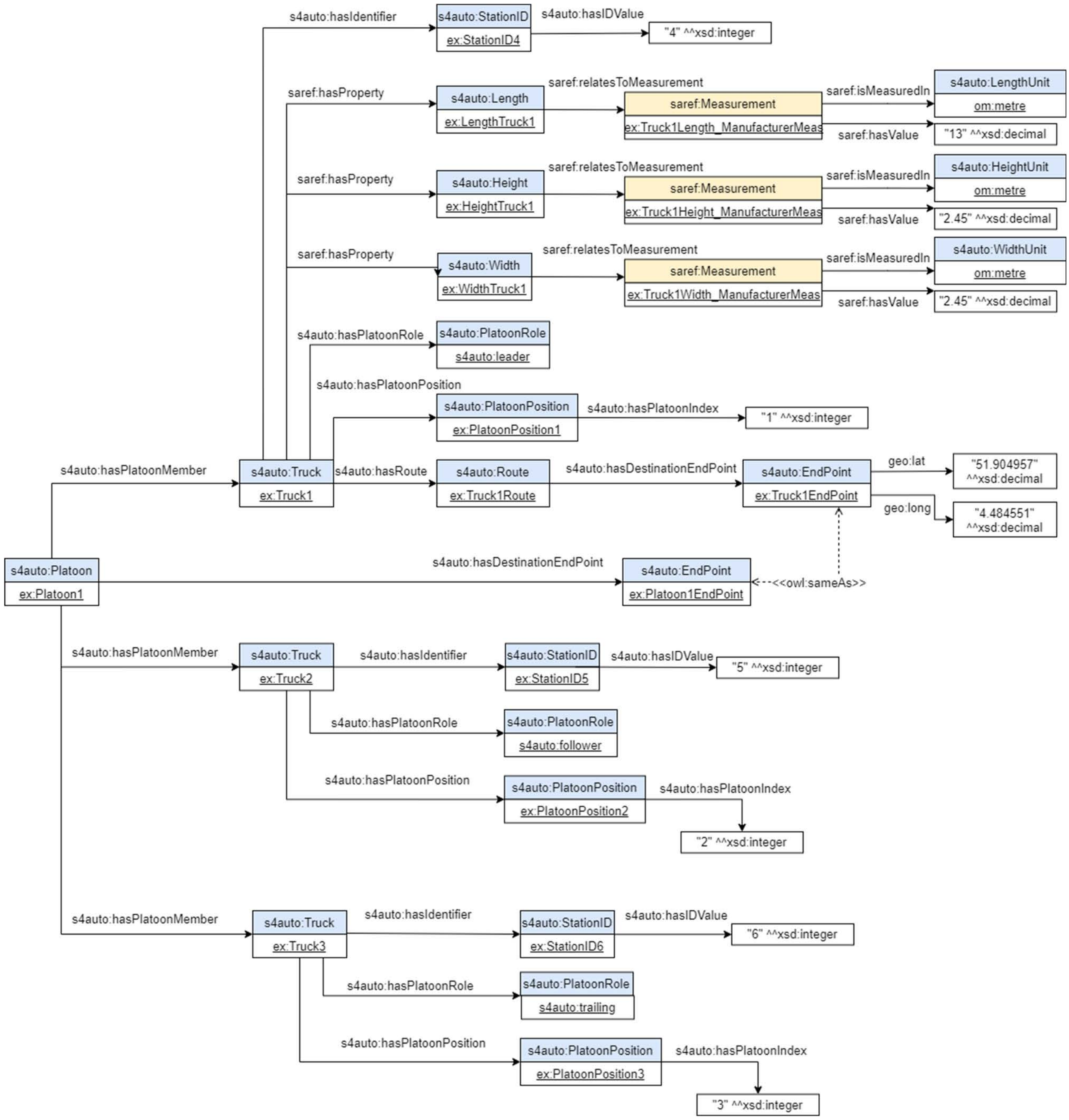

Figure 10: Platoon members example

Figure 10 further shows the fixed dimensions defined by a manufacturer for its vehicle. For example, Truck1 is characterized by a length of 13 metres, an height of 2,45 metres and a width of 2,45 metres. This is expressed, respectively, by the *ex:Truck11Lenght_ManufacturerMeas*, *ex:Truck11Height_ManufacturerMeas* and *ex:Truck11Width_ManufacturerMeas* instances of the `saref:Measurement` class. Note that these are fixed dimensions defined by a manufacturer and do not change over time. Therefore, they do not have an associated timestamp. In cases when a trailer is present in the vehicle (e.g. in a truck), the company operating the vehicle may set this value at the beginning of its journey. In contrast, the dimensions of the same vehicle can be dynamically measured over time on the road, for example, by another vehicle in the surrounding environment for the purpose of automated driving or the assembling of a platoon. In this case, a timestamp shall be included to indicate at which moment in time the measurement has been taken.

4.3.2 Automated Valet Parking

Clause 4.3.2 of the present document shows an example of how to instantiate the SAREF4AUTO extension for the Automated Valet Parking (AVP) use case. The AVP is a use case where an automated vehicle uses driving functions based on knowledge about the surrounding environment to park, such as navigation functionality based on a digital map, positions of the automated vehicle and vacant parking spots. The vehicle can use its own functions and sensors to observe the immediate environment, but it can also access IoT platforms that can provide data and functions based on IoT-enabled sensors, like parking cameras, as well as position information from other vehicles driving (or being parked) at parking. Furthermore, IoT platforms may provide information to support services for booking a parking place and arranging (automated) payment. The example instantiation in this clause describes a car, its properties and its possible parking states in relation to a parking spot with its topology and states.

This example instantiation is referred to using the *ex:* prefix. This prefix is different from the `s4auto:` prefix, which indicates the SAREF4AUTO ontology on which the *ex:* example instantiation is built upon.

Figure 11 shows an instance of a vehicle, *ex:Car1*, which is about to automatically drive towards a nearby parking spot, which is shown in Figure 12. This car is characterized by an identifier (`s4auto:hasIdentifier` property), which is defined as a Station ID (*ex:StationID1* instance with value "1"). Figure 11 further shows the fixed dimensions defined by the manufacturer for this car, which is characterized by a length of 3 metres, an height of 1,5 metres and a width of 1,45 metres. This is expressed, respectively, by the *ex:Car1Lenght_ManufacturerMeas*, *ex:Car1Height_ManufacturerMeas* and *ex:Car1Width_ManufacturerMeas* instances of the `saref:Measurement` class. As mentioned for the trucks in the platoon example, these are static dimensions defined by the manufacturer that do not change over time and consequently do not have an associated timestamp. In cases when a trailer is present in the vehicle (e.g. in a truck), the company operating the vehicle may set this value at the beginning of its journey. In contrast, the dimensions of the same car can be dynamically measured over time on the road, for example, by another vehicle in the surrounding environment for the purpose of automated driving or by sensors that support the car for parking. In this case, a timestamp shall be included to indicate at which moment in time the measurement has been taken.

Figure 11 further shows that the car is provided with a `s4auto:ElectronicControlUnit` (*ex:Car1ECU* instance) that can make measurements about relevant properties, such as speed, acceleration, position, etc. In the example, the electronic control unit provides a speed measurement (*ex:Car1Speed_Meas1* instance) with a value of 100 km per hour with an associated timestamp. As all measurements are subject to uncertainty, a confidence is always associated with measurement values. Therefore, a confidence of the *ex:Car1Speed_Meas1* measurement is also provided and show in Figure 11 as the *ex:Confidence1* instance with a value of 3 km per hour.

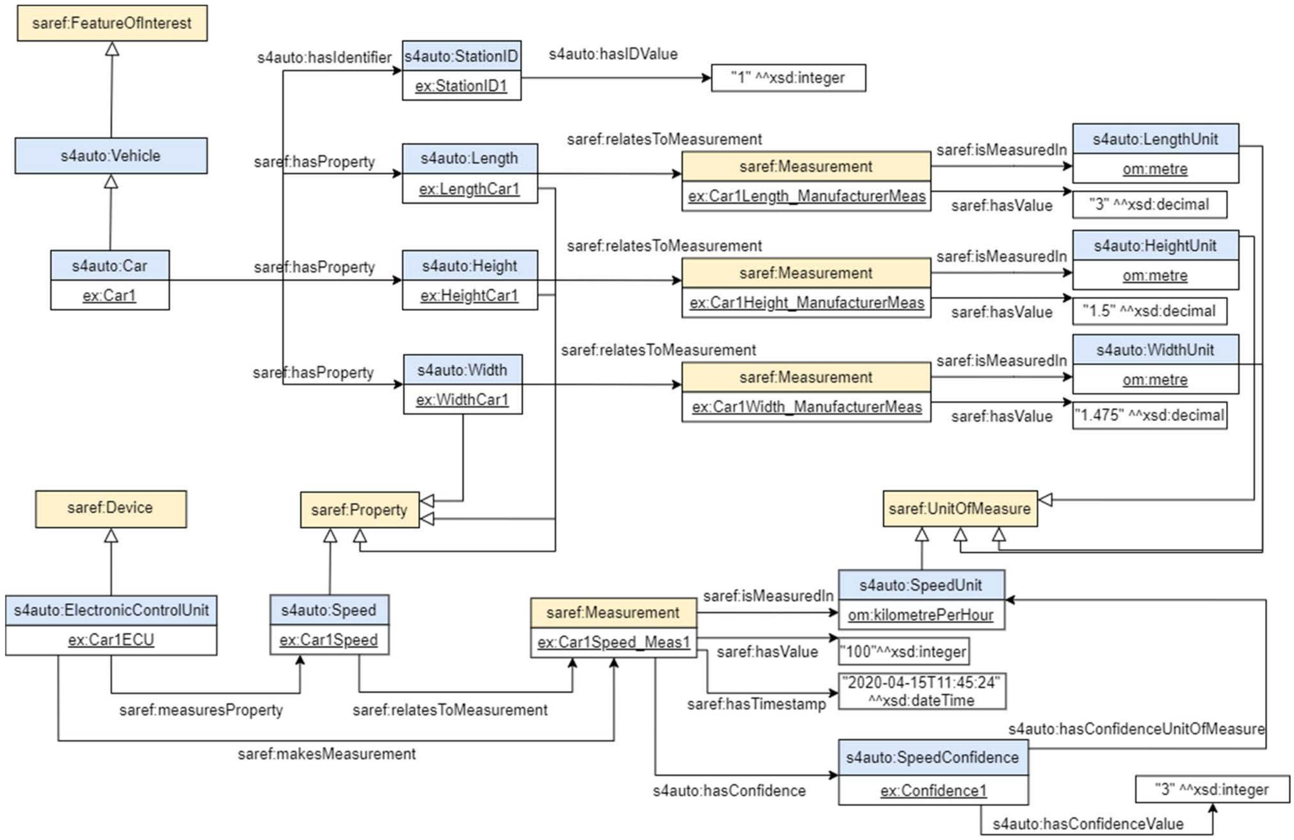

Figure 11: Car example

Figure 12 shows how the GeoSPARQL ontology has been reused in SAREF4AUTO to represent the parking spot topology. A parking spot is defined as the *ex:ParkingSpot1* instance of the `s4auto:RegularParkingSpot` class, which in turn is a subclass of the more general `geosp:Feature` class that allows to define spatial features. The geometry of a parking spot is usually defined by several points describing the shape of the spot, e.g. 4 points describing a rectangle shape and then a central point. For the sake of simplifying the example, the geometry of this parking spot is simplified defining only its central point, namely the *ex:ParkingSpot1* instance of the `s4auto:RegularParkingSpot` class, with a latitude and longitude that have values 52.081927 and 4.325145, respectively. Note that the `sf:Polygon` class described in clause 4.2.7 can be used to further define the rectangle shape of the parking spot. The parking spot is characterized by different states over time (`s4auto:hasParkingSpotState` property), depending on the states of the car that is driving towards the spot (which are shown in Figure 13). For example, at *14:00:00* of *2020-04-08* the parking spot is free (*ex:ParkingSpot1Free* instance), whilst a few minutes later at *14:03:00* the parking spot becomes reserved (*ex:ParkingSpot1Reserved* instance), since Car1 arrived at the drop off spot (shown in Figure 13) and reserved a nearby spot available, which happens to be ParkingSpot1. At *14:07:00*, after the car drives to the parking spot and parks, the parking spot becomes occupied (*ex:ParkingSpot1OCcupied* instance).

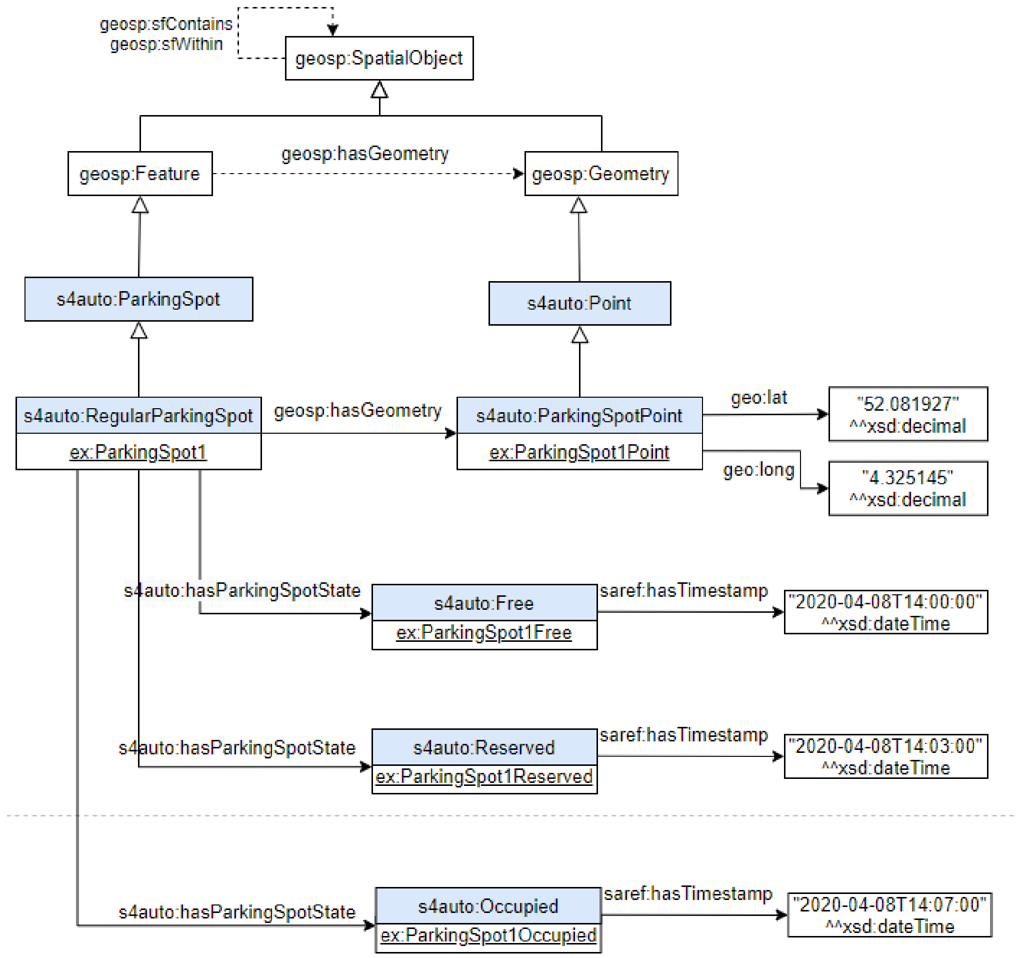

Figure 12: Parking Spot example

Figure 13 elaborates on the concepts needed for the car to automatically drive towards the parking spot described above. First, the car needs a route towards the parking spot (*ex:Car1ParkingSpot1RoutePoint* instance) with an origin (`s4auto:hasOrigin` property) and a destination (`s4auto:hasDestinationEndPoint` property). The origin is the chosen point where the car is dropped off for the automated valet parking (*ex:Car1DropOffPointParkingSpot1* instance), which has latitude and longitude of 52.082028 and 4.330057, respectively. The destination of the car's route is set as the same as (`owl:sameAs` property) the point where the parking spot is located, namely the *ex:ParkingSpot1Point* instance, with a latitude and longitude of 52.081927 and 4.325145, respectively.

Finally, Figure 13 shows the different states of the car during the AVP. At *14:02:10* of *2020-04-08* the car arrives at the drop off spot (*ex:Car1AtDropOffSpot1* instance) and looks for an available nearby parking spot (provided by a third party service in the IoT platform). The *ex:ParkingSpot1* is subsequently reserved and its state becomes reserved at *14:03:00* (*ex:ParkingSpot1Reserved* instance shown in Figure 12). The car drives to the reserved parking spot and at *14:04:00* has state *ex:Car1DrivingToParkingSpot1*, which then turns into the *ex:Car1ParkinginSpot1* state at *14:07:00*, once the car has arrived at the parking spot and it is parking. Note that at the same moment, the state of the parking spot becomes occupied (*ex:ParkingSpot1Reserved* instance shown in Figure 12). At *14:10:00*, once the car is parked, its state turns to parked (*ex:Car1ParkedInSpot1* instance).

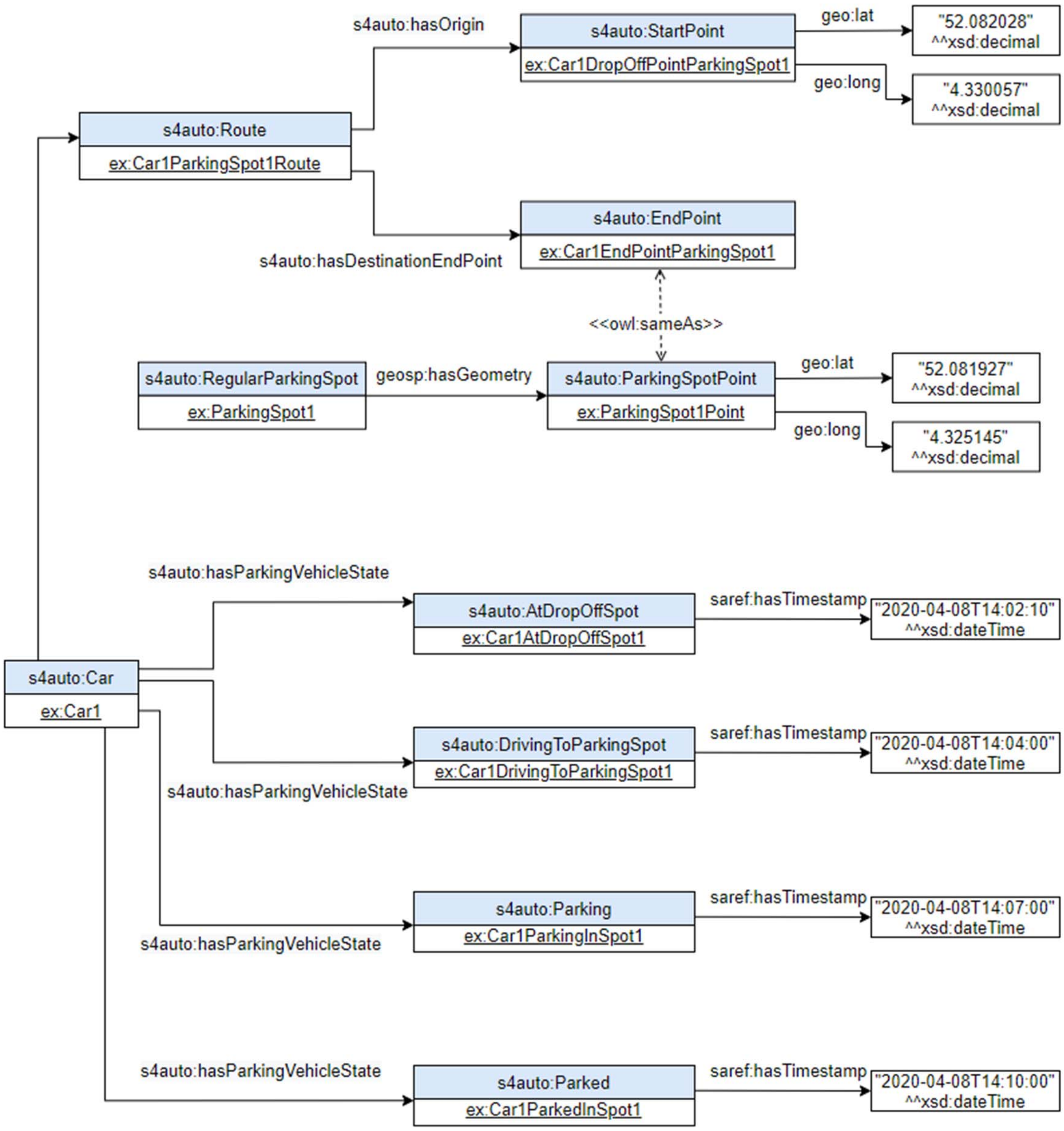

Figure 13: Automated Valet Parking example

4.3.3 Vehicle environment at a crossroad

This clause shows an example instance of a vehicle at a crossroad which is able to recognize the different road entities in its environment. For each entity, the information related to its properties is included. The example shows in the environment a vehicle coming in the opposite direction and another one coming from the right. There are two pedestrians, one waiting to cross a road (thus not moving) and a young child (smaller height) crossing while it is not allowed to, thus generating an alert from a smart traffic light. A motorcycle is also crossing from the left of the instantiated vehicle. Figure 14 helps visualizing this example.

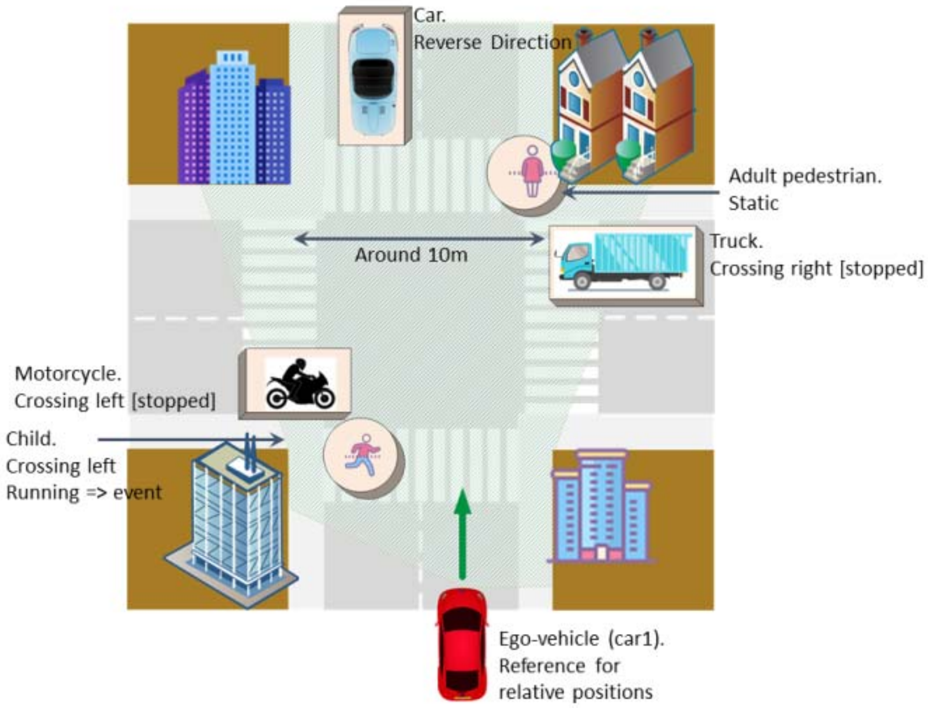

Figure 14: Example scenario for vehicle environment with VRU

Figure 15 contains an example of instantiation of SAREF4AUTO for the example featured in Figure 14. The main feature of interest in the example is Car1 (`ex:Car1`) which indicates that it is able to sense its environment (`ex:EnvironmentDetected`). At the time of detection, Car1 has a position (`ex:Car1AbsolutePosition1`) and a speed (`ex:Car1Speed`) which is measured as 10 km/h (Car1Speed_Meas1). Figure 15 also illustrates the different entities that compose the environment (`ex:Car1VehicleEnv`) of Car1: two vehicles, i.e. a car (`ex:Car4`) and a truck (`ex:Truck4`), as well as four other road entities: a child pedestrian (`ex:RoadEntityChild1`), an adult pedestrian (`ex:RoadEntityAdult1`), a motorcycle (`ex:RoadEntityMoto1`) and an event (`ex:RoadEntityEvent1`) which is received because the child is crossing the road in front of Car1, while it is not allowed to do so.

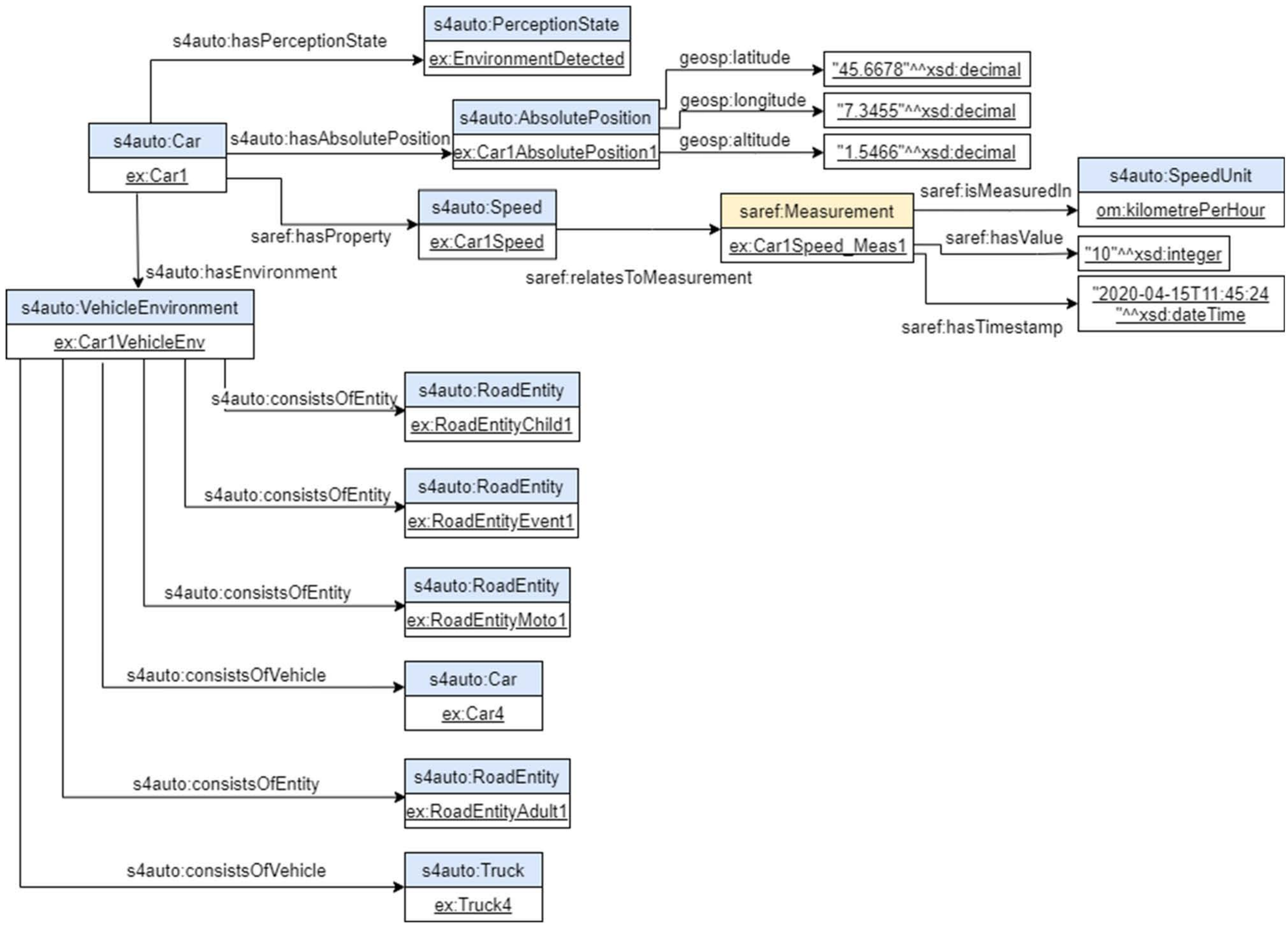

Figure 15: Example of Vehicle environment at a crossroad

Figure 16 provides examples of the instantiation of the different road entities when sensed as part of the vehicle environment.

As for the sensed vehicles, the example of Truck4 (`ex:Truck4`) indicates the main parameters of the sensing. First, its position (`ex:Truck4RelativePosition`) relative to Car1 (`ex:Car1`) with coordinates in the reference system made of a reference point, the absolute position of Car1 (`ex:Car1AbsolutePosition1` in Figure 15), and three dimensions, x is orthogonal to Car1 direction, y is in front of Car1 and z is vertical compared to Car1 trajectory. The truck has a speed (`ex:Truck4Speed`) which as measured (`ex:Truck4Speed_Meas1`) as a value of 0 km/h (`ex:km_per_hour`), because the truck is stopped waiting for the passing of Car1.

The example of Car4 (`ex:Car4`) is similar, with Relative position (`ex:Car4RelativePosition`) and speed (`ex:Car4Speed`), but it is less developed in Figure 16 to maintain readability.

Regarding the other road entities, the example of the adult (`ex:RoadEntityAdult1`) shows how vulnerable road users are instantiated. The adult consists of a user of profile P1 (`ex:adult1`) which has a relative position (`ex:Adult1RelativePosition`) to Car1. The relative position is described in the same manner as for Truck4 (`ex:Truck4`). The adult pedestrian is characterized by two properties: her height (`ex:adult1Height`) and her movement (`ex:static`). Her height is measured as 1,80 metre (`ex:Adult1Height_Meas`). Her movement is considered as static because the adult is waiting on the sidewalk for the passing of Car1.

The examples of the child (`ex:RoadEntityChild1`) and motorcycle (`ex:RoadEntityMoto1`) are instantiated in a similar manner.

The last road entity shown in Figure 16 is the event (`ex:RoadEntityEvent1`) that is disseminated to indicate to vehicles that the child (`ex:RoadEntityChild1`) is crossing the road while it is not allowed to do so. This entity consists of a notified event (`ex:humanPresenceOnTheRoad`) and has the same values for its relative position (`ex:Event1RelativePosition`) as the child (`ex:RoadEntityChild1`).

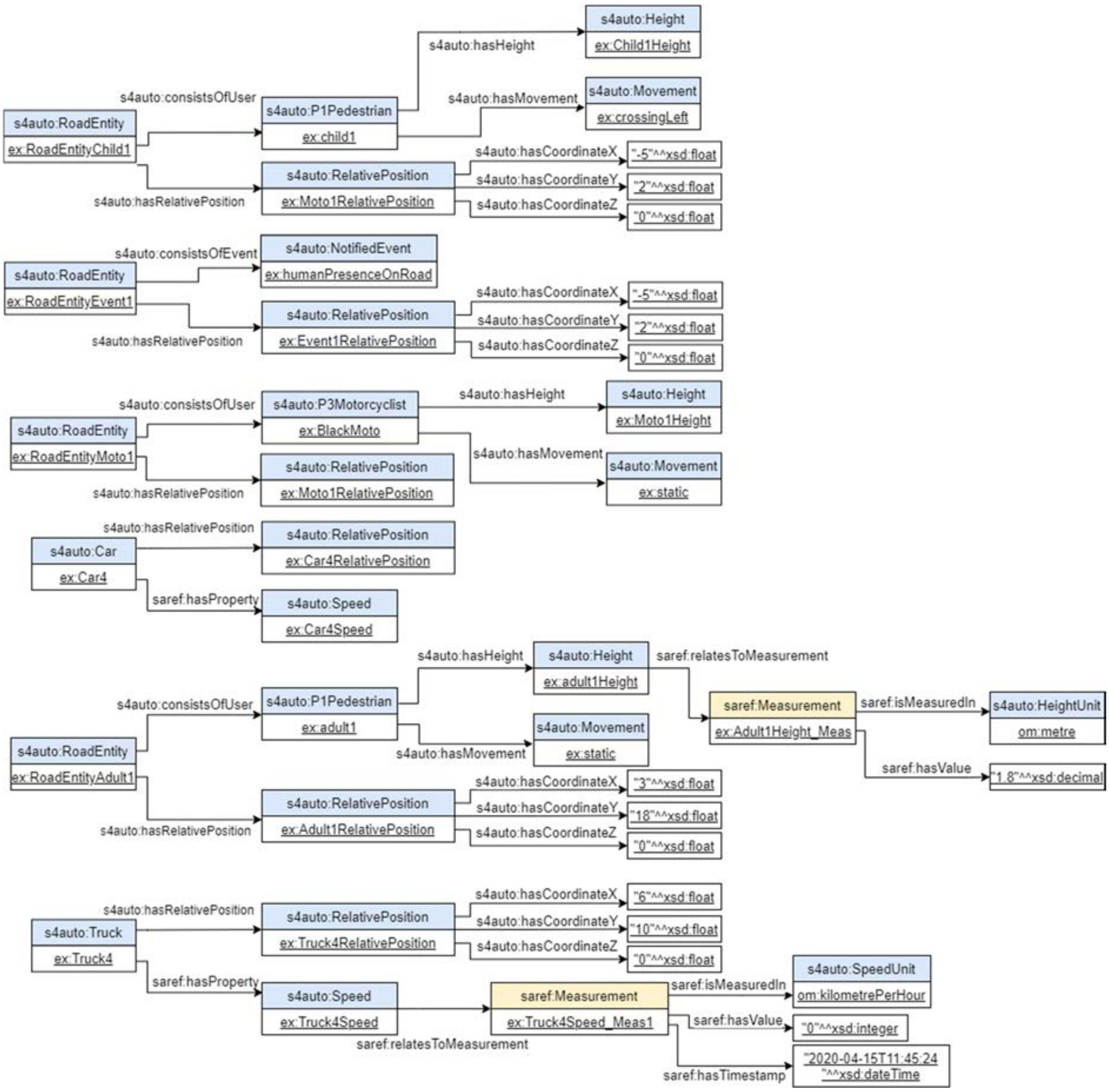

Figure 16: Example of sensed entities with Vulnerable Road Users

4.4 Observations

In the following, several observations about the SAREF4AUTO ontology and its usage are mentioned.

The hierarchies and individuals defined in the extension should not be considered exhaustive. It is possible and recommended to extend the hierarchies and lists of individuals for new and additional use cases, as well as to further specialize the classes defined in this extension if needed, for example, the hierarchy of type vehicles and their subclasses specified in clause 4.2.3 of the present document.

SAREF4AUTO extends the SAREF core ontology with the `s4auto:State` class and the `s4auto:hasState` property to define the various states in which vehicles, platoons and parking spots can assume over time. The following hierarchy of states has been defined in SAREF4AUTO:

- `saref:State`
  - o `s4auto:State`
    - ▪ `s4auto:ParkingSpotState`
      - `s4auto:Charging`
      - `s4auto:Closed`
      - `s4auto:Free`

        - s4auto:Occupied
        - s4auto:Reserved
    - s4auto:ParkingVehicleState
        - s4auto:AtDropOffSpot
        - s4auto:AtPickUpSpot
        - s4auto:Charging
        - s4auto:DrivingToParkingSpot
        - s4auto:DrivingToPickUpSpot
        - s4auto:Parked
        - s4auto:Parking
    - s4auto:PerceptionState
    - s4auto:PlatoonState
        - s4auto:Assembling
        - s4auto:Disengaging
        - s4auto:Platooning
        - s4auto:Standalone
        - s4auto:Unknown
    - s4auto:PlatoonVehicleState
        - s4auto:Disengaging
        - s4auto:Engaging
        - s4auto:Forming
        - s4auto:Platooning
        - s4auto:Searching
        - s4auto:Standalone
        - s4auto:Unknown

Since the states above can change over time, the `saref:hasTimestamp` property has been reused from SAREF and associated to the `s4auto:State` class and it is, consequently, inherited by all its subclasses. In this way, it is possible to associate to a certain state the moment in time it occurs. The following hierarchy of properties has also been defined in SAREF4AUTO to be used with the `s4auto:State` class as a range:

- s4auto:hasState
    - s4auto:hasParkingSpotState
    - s4auto:hasParkingVehicleState
    - s4auto:hasPerceptionState
    - s4auto:hasPlatoonState
    - s4auto:hasPlatoonVehicleState

The properties above are used with the following classes as a domain:

- geosp:Feature
    - s4auto:ParkingSpot
- s4auto:Platoon
- saref:FeatureOfInterest
    - s4auto:Vehicle

The main observation regarding states is that the `saref:hasState` property could not be reused as super-property of the newly defined `s4auto:hasState` property, since the `saref:hasState` property is restricted to have only `saref:Device` as a domain, whereas the `s4auto:ParkingSpot`, `s4auto:Platoon` and `s4auto:Vehicle` classes in SAREF4AUTO are not type of devices. Therefore, it is recommended to remove this restriction in the SAREF core ontology, as not only devices are characterized by states. It is furthermore recommended to "promote" the `saref:hasTimestamp` property associated in SAREF4AUTO to the `s4auto:State` class, a level up to the `saref:State` class, as it important to associate to any state, also the state of devices, the moment in time when they occur in order to enable dynamic reasoning about them, which is an essential requirement of the practical applications that will use SAREF and its extensions.

The route of a vehicle has a destination that can be expressed as an end point in terms of longitude and latitude, and/or in the form of an address, for which a `s4auto:Address` class has been defined. It is out of scope of SAREF4AUTO to define the details of such an address. It is instead recommended to reuse existing ontologies that specify how to model an address in terms of Street, Postal Code, City, Region, Country, etc.

When presenting SAREF4AUTO to the stakeholders, in several occasions the question has been asked on the relationship with the W3C automotive model (http://www.automotive-ontology.org/). It appears that the W3C mainly standardizes vehicle configuration and characteristics (colour, number of doors, engine power) while SAREF4AUTO standardizes the vehicle behaviour in traffic situations. Therefore, the two models rather than alternatives to each other, can be used to complement each other.

Privacy and data ownership are important topics in the automotive domain, as they are in all domains, although there is a special focus on these topics in the automotive domain. Therefore, related information would then need to be taken into account, for example, as additional meta data of measurements, together with the unit of measurement. As the same observation applies to most of the SAREF extensions, privacy and data ownership concerns could also be addressed at the core SAREF level. Existing ontologies addressing these concerns could be linked to SAREF for this purpose.

History

Document history		
V1.1.1	July 2020	Publication