TECHNICAL SPECIFICATION

ISO/TS 5206-1

First edition
2023-04

Intelligent transport systems — Parking —

Part 1: Core data model

Systèmes intelligents de transport - Stationnement —

Partie 1: Modèle de données de base

Reference number
ISO/TS 5206-1:2023(E)

Documents
Document Sharing Hub
© ISO 2023

Full document details: ISO TS 05206-1-2023

COPYRIGHT PROTECTED DOCUMENT

© ISO 2023

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

© ISO 2023 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO TS 05206-1-2023

Contents

Page

Foreword ... **v**
Introduction ... **vi**
1 Scope ... **1**
2 Normative references ... **1**
3 Terms, definitions and abbreviated terms ... **1**
3.1 Terms and definitions ... 1
3.2 Abbreviated terms ... 2
4 Conformance ... **2**
5 UML notation ... **2**
6 Data model ... **3**
6.1 Data model overview ... 3
6.2 Place data concepts ... 4
6.2.1 Place hierarchy ... 4
6.2.2 IdentifiedArea sub-types ... 12
6.2.3 Common components ... 16
6.2.4 Payment method ... 17
6.2.5 Basic elements (in place) ... 18
6.3 Occupancy ... 19
6.3.1 Introduction to occupancy ... 19
6.3.2 Supply ... 20
6.3.3 Demand ... 21
6.3.4 Enumerations (for occupancy) ... 21
6.3.5 External codelists (for occupancy) ... 22
6.4 Rates ... 22
6.4.1 Introduction to rates ... 22
6.4.2 Rate tables ... 22
6.4.3 Eligibility ... 24
6.4.4 Enumerations (for rates) ... 25
6.4.5 External codelists (for rates) ... 26
6.5 Right ... 26
6.5.1 Enumerations (for right) ... 37
6.5.2 External codelists (for right) ... 38
6.6 Session ... 38
6.6.1 General ... 38
6.6.2 Enumerations (for session) ... 40
6.6.3 External codelists (for session) ... 40
6.7 Observation ... 40
6.7.1 Enumerations (for observation) ... 42
6.7.2 External codelists (for observation) ... 42
6.8 Quote ... 42
6.8.1 Introduction to quote ... 42
6.8.2 QuoteRightRequest — Request for a new transaction ... 43
6.8.3 QuoteRightResponse — Response to a QuoteRightRequest for a new transaction ... 44
6.8.4 QuoteSessionExtensionRequest — Request for an extension of an existing session ... 45
6.8.5 QuoteSessionExtensionResponse — Response to a QuoteSessionExtensionRequest for an extension of an existing session ... 46
6.8.6 Enumerations (for quote) ... 48
6.8.7 External codelists (for quote) ... 48
6.9 Common elements ... 48
6.9.1 Introduction to common elements ... 48

© ISO 2023 – All rights reserved

AVA Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

6.9.2 Data types (general) 50
6.9.3 Enumerations (general) 50
6.9.4 Common classes 51
6.9.5 External codelists 52
6.9.6 Organisation, contacts and address 53
6.9.7 Describing Locations 56
6.9.8 Data types (for location) 64
6.9.9 Enumerations (for location) 64
6.9.10 External codelists (for location) 66
6.9.11 Times 66
6.10 Facilities 69
6.10.1 Overview 69
6.10.2 Data types (for facilities) 70
6.10.3 Enumerations (for facilities) 70
6.10.4 External codelists (for facilities) 70
6.11 Energy infrastructure 71
6.11.1 Overview 71
6.11.2 Data types (for energy infrastructure) 72
6.11.3 Enumerations (for energy infrastructure) 73
6.11.4 External codelists (for energy infrastructure) 74

Annex A (normative) **Data dictionary** **75**

Annex B (informative) **Relationship to CEN 16157 "DATEX II" Data Modelling Concept and Framework** **196**

Annex C (informative) **Use case examples** **197**

Annex D (informative) **Example user-defined codelists** **246**

Bibliography **247**

© ISO 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 204, *Intelligent transport systems*.

A list of all parts in the ISO 5206 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

© ISO 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

Introduction

Around the globe, new services and technologies are intersecting to create value-added convenience to customers and business owners while better utilizing available parking, mobility and transport infrastructure. Whether supporting car sharing, ride sharing, micro mobility services, prepaid parking, dynamic pricing in parking structures, remote management of operations, and/or improved reporting, the sharing of data is key to accelerating the adoption of these services.

To enable the sharing of data, the global community has collaborated to create consensus-built, international parking and mobility data specifications to establish a common language for data concepts and definitions in the parking, transport and mobility sectors. These “data specifications” define a common language composed of a set of data concepts and definitions that public and private property owners, operators and service providers can follow to facilitate communication between themselves and with the other industries. These specifications facilitate seamless integration, compatibility and communication between parking entities, mobility operators, the automotive industry, IT developers, ITS operators, services, and map and app providers, as well as other stakeholders.

These specifications, as defined in this document, seek to reduce the effort required to connect technology solutions to one another and to allow companies to refocus their resources on innovating new services and operations.

This document defines the structure, definition and relationships of data constructs relevant to parking and mobility data. This document provides a description of the data constructs in the context of the data model in Clause 6.

Annex A provides a data dictionary of definitions for attributes, classes and relationships.

Annex B provides a description of the data modelling approach defined in this document and how this relates to the CEN 16157 series "DATEX II" modelling approach, defined in EN 16157-1.

NOTE By choice, the Alliance for Parking Data Standards has substantively adopted the DATEX II modelling approach, as this can potentially facilitate simpler integration, at a later date, to data concepts and standards defining road traffic management and information concepts.

Annex C provides use cases and examples for the use of the data defined.

© ISO 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

Intelligent transport systems — Parking —

Part 1:
Core data model

1 Scope

This document defines terms, characterization and the relationship of concepts, defined using model-driven architecture methods, for parking and parking-related activities (both on-street and off-street) covering common data supporting business to business exchanges and end user services.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 639-1, *Codes for the representation of names of languages — Part 1: Alpha-2 code*

ISO 3166-1, *Codes for the representation of names of countries and their subdivisions — Part 1: Country code*

ISO 4217, *Codes for the representation of currencies*

ISO 8601-1, *Date and time — Representations for information interchange — Part 1: Basic rules*

ISO/IEC 10646, *Information technology — Universal coded character set (UCS)*

ISO/IEC 19505-1, *Information technology — Object Management Group Unified Modeling Language (OMG UML) — Part 1: Infrastructure*

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1.1
data receiving party
entity that is requesting or receiving data using the specification from a reliable data source

3.1.2
data distributing party
entity holding data with permission to distribute, that is issuing data using the specification to other entities

© ISO 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

3.2 Abbreviated terms

ANPR automatic number plate recognition

RFID radio frequency identification

UML unified modeling language

4 Conformance

A specification, system or exchange of data (hereafter called "specification") claiming conformance to this document shall include a comparison of the specification of data against the data model defined in this document. This comparison shall be defined at two conformance levels:

— level 1 (data domain) which identifies which data domains within the specification correspond to the data domains specified in this document, and which are not;

— level 2 (detailed level) which compares the data model within the specification against the packages, classes, attributes and relationships specified in this document.

The level 1 conformance statement should be presented as a table based on the data domains defined in the normative part of this document. This includes place, occupancy, session, right, rates, quote, observation, energy infrastructure and parking common (PkCommon); or

The level 2 conformance statement should be presented as a table in which the data concepts used in the specification are described as follows.

— “Unmodified”: concepts in the specification which have the same definition, properties and relationships as in the corresponding data domain specified in this document.

— “Modified”: concepts in the specification which are similar to the data concept defined in this document, but which differ in the details of certain attributes and/or relationships (e.g. attributes added).

— “Alternative”: concepts or groups of concepts in the specification intended to model the same concepts as defined in this document but in a significantly different way.

— “Additional”: concepts in the specification which are additional to the concepts defined in this document and therefore are not covered by and drawn from the data concepts specified in this document.

— “Omitted”: data concepts specified in this document which are not used in the specification.

Data concepts shall include packages, classes, attributes, relationships, definitions and properties, and comparison shall present conformance with the following requirements as expressed in this document:

— conformance to all stipulated minimum and maximum multiplicity requirements for UML elements and relationships;

— conformance to all definitions, types and ordering;

— employment of optional elements as specified;

— conformance to all expressed constraints.

5 UML notation

The UML notation used in this document shall be as described in ISO/IEC 19505-1.

NOTE The use of UML notation follows the DATEX II methodology which is specified in EN 16157-1.

© ISO 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

6 Data model

6.1 Data model overview

This specification facilitates the sharing of basic information between entities and systems. This includes map services, online marketing and aggregator services, event ticketing platforms, transit and transportation agencies, and other firms, organizations or individuals that have a need to know the location of parking and other mobility-related services and general information about the operation.

Figure 1 shows the data domains in this document. Each data domain defines a specific set of data concepts that logically can be grouped.

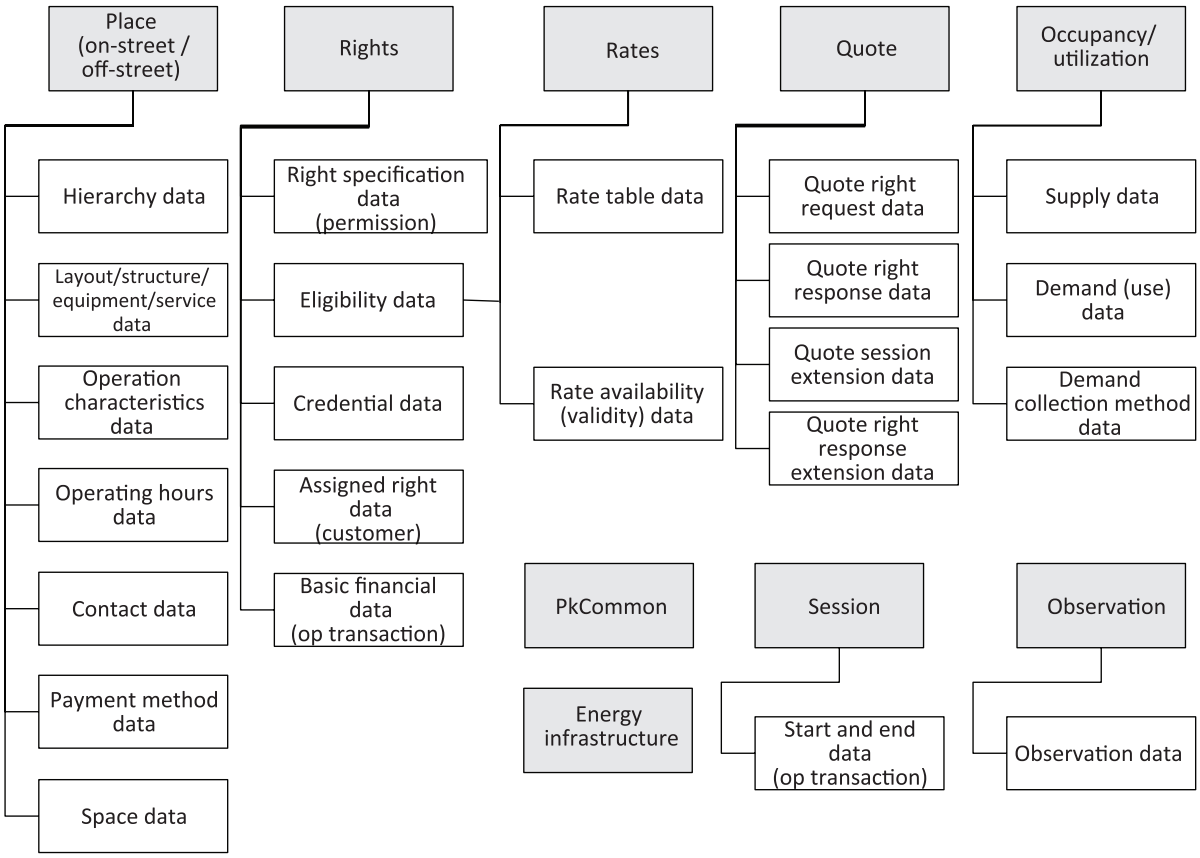


Figure 1 — Data domains

To assist navigation and understanding of the specification in this document, the overall model is separated into data domains, which can be seen in UML packages in Figure 2. The domains focus on functional groups of data concepts and relationships. Each of the domains is specified in its own namespace. In addition, the structure of the data model supports the definition of common concepts that are referenced and reused across several domains. These common concepts are specified in the "PkCommon" domain.

© ISO 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

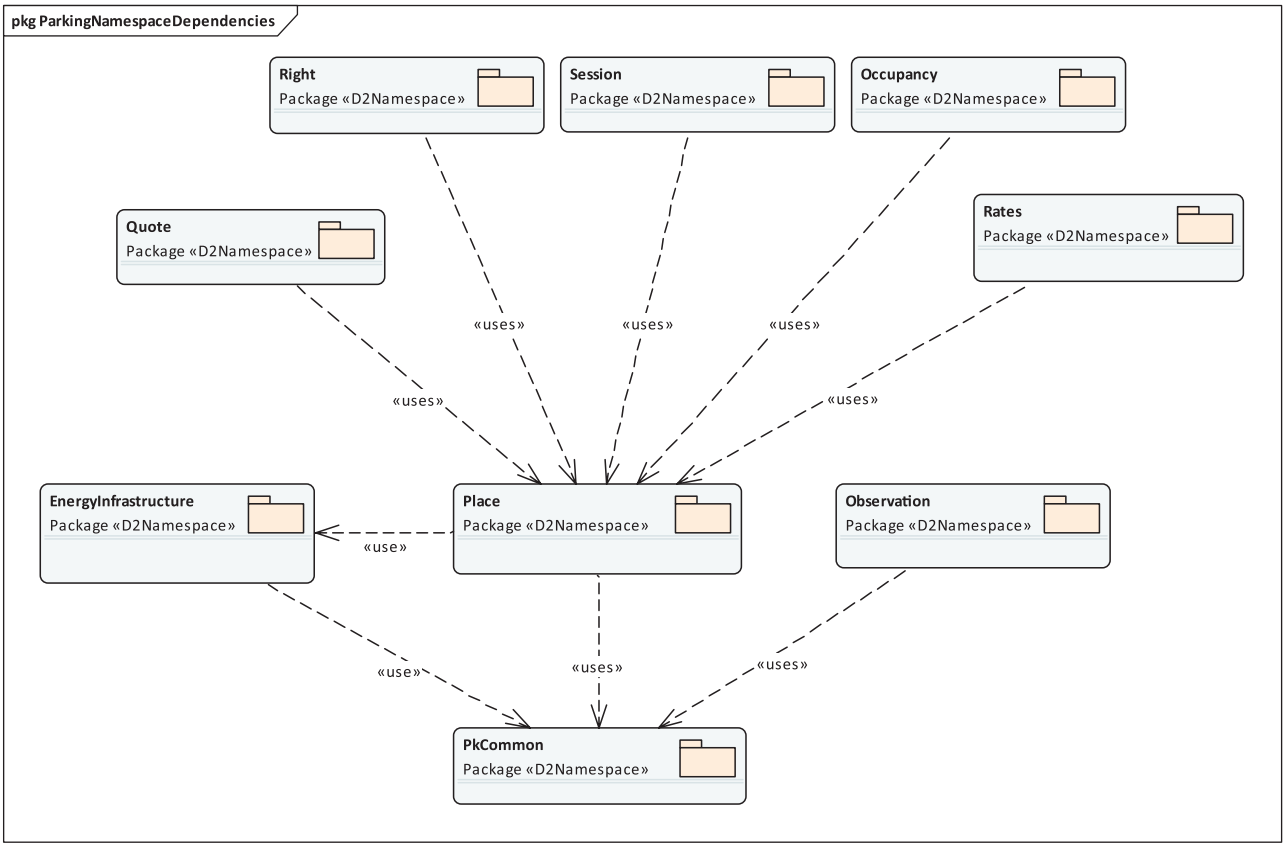


Figure 2 — Namespace dependencies

Figure 2 illustrates the dependency between namespaces.

These data domains, as characterized by the namespaces, are combined in various manners to support specific use cases that exist in parking and mobility operations. For example:

— the operations and systems to support automated valet parking rely on this document's data domains of place, rate, quote, session, right and occupancy;

— the operation and systems to support curbside management rely on all data domains in this document to support data sharing.

6.2 Place data concepts

6.2.1 Place hierarchy

6.2.1.1 Overview

This subclause describes concepts used in the place hierarchy, which are defined in the <Place> namespace of the data model.

NOTE Location, time and contact/organization concepts are defined in the <PkCommon> namespace of this data model.

6.2.1.2 Place hierarchy

The specification defines a method for building a hierarchy of place records in on-street, off-street and zone environments. Each instance of a place record corresponds to an instance of a class within the place hierarchy. This enables a parking or other type of operation to identify a location or zone in a

© ISO 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

simple manner with a single hierarchy element or to break down a place into a multi-layered hierarchy that identifies discrete parking enclosures or defined areas to communicate operating hours, space counts, operating restrictions, location, rights and associated pricing and occupancy.

The hierarchy supports the ability of lower-level place hierarchy class instances (child records) to inherit specific instance data from a higher-level place hierarchy class instance (a parent record) in order to simplify the amount of data shared. The place hierarchy also enables lower-level place hierarchy class instances to document variations in specific attributes from their parent, higher-level place hierarchy class instance.

Additionally, the specification enables other operation types, not directly related to parking, to define a place in discrete operating enclosures. This may include a defined on-street area for managing delivery services or a sidewalk area enabled for bike or e-scooter placement.

The hierarchy allows a data distributing party to decide the appropriate level of detail to send to a data receiving party via messaging protocols. In addition, it is not necessary to build multi-layer hierarchies of Place data. Simple data needs can be represented without using the multiple layers available in the data specification.

Each of the component Place data concepts is mentioned in this subclause, but further details and descriptions follow in subsequent subclauses. These component data concepts are:

- `Place` (6.2.1.4),
- `SubplaceElement` (6.2.1.5),
- `IdentifiedArea` (6.2.1.6),
- `Space` (6.2.1.7),
- `Campus`,
- `PedestrianAccess` (6.2.2.1),
- `VehicularAccess` (6.2.2.2),
- `SpecificArea` (6.2.2.3), and
- `SupplementalFacility` (6.2.2.4).

Place is a term introduced in the specification to define where a vehicle may park, stand, rest or briefly transit to allow a person to change modes of transport (i.e. taxi drop-off/pickup, ride share drop-off/pickup, valet stand, etc.). Place is instantiated via the `Place` class. Place can also be used to define entry and exit roadways, driveways, and acceleration/deceleration zones for vehicles as well as pedestrian access points. Place supports both on-street and off-street operating environments. Place also defines specific areas to be defined for mobility or other related uses such as bike storage, e-scooter enclosures, etc. where it is useful to share operating parameters or assign RightSpecifications, Rates or other data domains.

A `Place` is an aggregation of instances of `SubplaceElement`, `IdentifiedArea` and `Space`. In this specification, the lowest mandatory class instance to define a Place is the `IdentifiedArea`.

NOTE `Space` is a lower data concept instance that can exist below `IdentifiedArea` data concept instance, but it is not required.

An aggregation of `IdentifiedAreas` can create an instance of a `SubplaceElement` or a `Place`. Specific attributes are associated to each of the four specialization types of `IdentifiedAreas`.

An aggregation of instances of `SubplaceElements` can be linked to an instance of a higher level `SubplaceElement` or a `Place`. This allows a data provider to share as much or as little detail as necessary to support their operation.

© ISO 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

At the highest level of the place hierarchy, an aggregation of instances of `Place` can be linked to an instance of a `Campus`. An instance of `Campus` is not a required within an implemented place hierarchy.

Figure 3 illustrates how the instances of the various place hierarchy data concepts are combined to construct instances of nested parking hierarchy data concepts within one campus as a use case example.

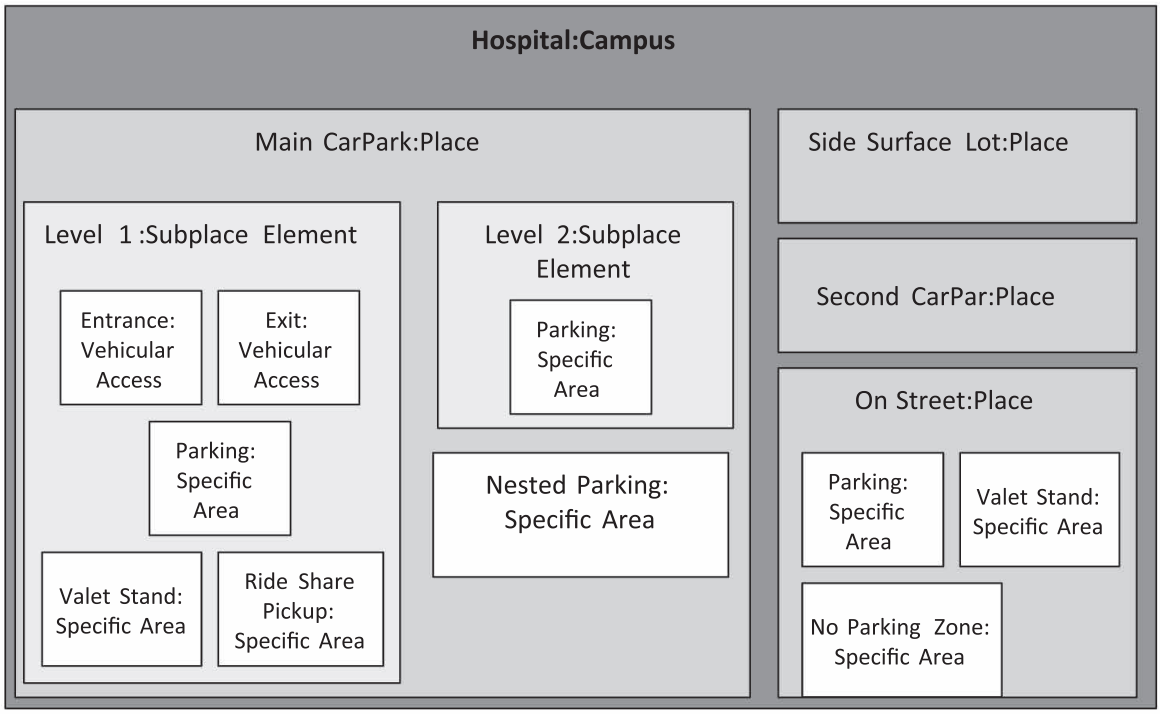


Figure 3 — Place hierarchy example – user-oriented

To support parking, mobility or curbside management operations, the `IdentifiedArea` class can be refined, through specialization, as one of four specialization types: `PedestrianAccess`, `VehicularAccess`, `SpecificArea`, and `SupplementalFacility`.

`PedestrianAccess` class is a specialization of `IdentifiedArea` class that enables a place to share coherent, relevant information about pedestrian entry and exit access points to a `Place`. This allows an entity to identify operating hours for the pedestrian access points and access requirements such as credentials or restrictions for each access point.

`VehicularAccess` class is a specialization of `IdentifiedArea` class that enables a `Place` to share coherent, relevant information about entry lane, exit lane and similar types of vehicular access points to a `Place`. The `VehicularAccess` class is used to define entry and exit to a specific facility or is used to provide a more macro description of an entry to a facility by reference to `Road` and `RoadNodes`.

> `Road`: allows an organizational entity to identify the primary road on which a place exists and to describe routes to that `Place`.
>
> `RoadNode`: allows an organizational entity to identify nearby major intersections of roads.

`SpecificArea` class is a specialization of the `IdentifiedArea` class that denotes a specific (generally geographically-bounded) area in a place that has a common operating purpose and common characteristics. Examples of common operating purpose include reserved parking area, electric vehicle parking, bike parking, delivery zone, loading zone, etc. The `SpecificArea` class describes the physical components of a `Place`.

For clarity, when defining the use of a `SpecificArea`, authorized activities to perform in an `IdentifiedArea` are defined by the assignment of a `RightSpecification` and include activities such as valet parking, disabled parking, delivery zone, etc. The assignment and authorization to perform a use

© ISO 2023 – All rights reserved
AVA Documents Document Sharing Hub
Full document details: ISO TS 05206-1-2023

in a `SpecificArea` occurs via the assignment of a `RightSpecification`. As an example, valet parking is associated to an `IdentifiedArea` with a `SpecificArea` class via the `RightSpecification` which assigns valet parking to a specific person and place.

`SupplementalFacility` is a specialization of `IdentifiedArea` class that enables a `Place` to identify the type and location of equipment and services available within the `Place`. The `SupplementalFacility` class is used to define restroom/toilet, shower, food concession services, etc. It is also used to identify specific equipment available in a `Place`, such as bike storage, electric chargers, payment stations, etc. This `IdentifiedArea` specialization defines physical structures and equipment in the place as well as services available.

The `IdentifiedArea` data concept collects general operating information such as operating hours, operating restrictions, space information and payment information. If the data is absent at the specific instance of an `IdentifiedArea`, it is assumed the data for this information exists at a higher level in the hierarchy, encapsulated in either instances of `Place` or any level instance of `SubplaceElement` above the instance of `IdentifiedArea`. This allows for customization of operations at lower levels while relying on default data from higher levels in the `Place` hierarchy.

A `Place` is synonymous with the structure or area with which a consumer associates a vehicle being parked or a mobility service being delivered. It can be an entire parking structure or an aggregation of streets supporting on-street parking (also sometimes called a zone). General operating information such as operating hours, operating restrictions, space information, payment information, etc. is associated to a `Place` and any parts of the hierarchy underneath it as appropriate.

`Place` is a core component of this model as it defines a hierarchy that supports the identification of portions of locations that may be related to parking and other types of operations. Use of this hierarchy enables data suppliers to provide a structured mechanism to refer to related zonal and place concepts with an ordered hierarchy.

Figure 4 shows concepts within the place hierarchy.

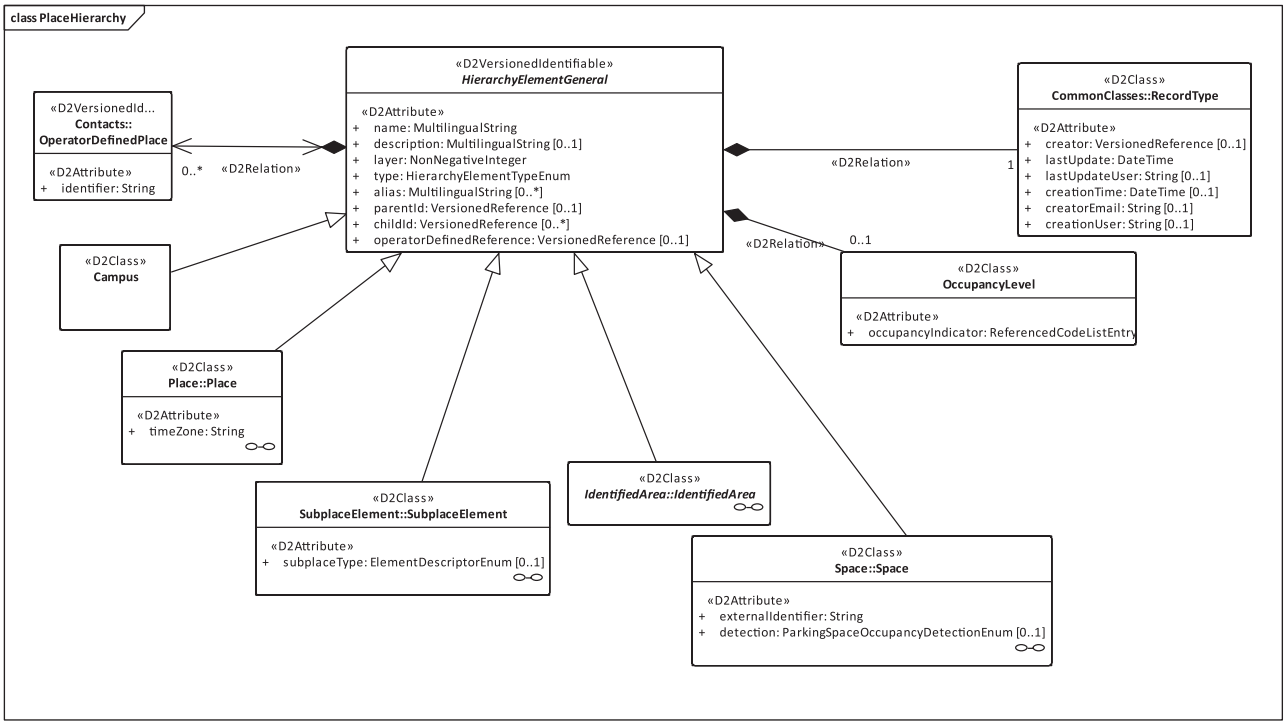


Figure 4 — Place hierarchy

© ISO 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

The concepts shown in Figure 4 are:

— `HierarchyElementGeneral` – a generalized component of a `Place` hierarchy that forms one element in the tree-like hierarchy. This forms a reusable block of the hierarchy, with relations to its parent element (if one exists) and any child elements. Each `Place` in the hierarchy shall have a name and may support a free-text description and an operator/property-owner-defined reference (e.g. location number/identifier).

— There are five specializations of the `Place` hierarchy (`HierarchyElementGeneral` class), which conceptually defines different scales of hierarchy elements. From largest to smallest, they are:

  — `Campus` – the highest level in the hierarchy, it typically defines a large facility (such as a university campus, or an airport), or a large geographic zone (such as a city or a town), which may contain numerous places that can be logically reported together. A campus combines and encompasses a number of `Places` that can be logically reported together. Different entities sharing data may create their own aggregation of `Places`. Thus, a `Place` may appear in different campuses if a receiving party receives data from different sending parties. For example, a parking operator may group five (5) places together in which it operates and call it a campus to reflect the five (5) operations in a city. Three of the same places may be associated with a property management firm that defined a separate campus with the three places.

  — `Place` – a place or location used for parking, loading, unloading, standing, or another mobility or transport related activity. Place typically identifies a parking structure, surface lot or on-street parking zone.

  — `SubplaceElement` – a sub-division of a `Place` for the convenience of the operator to segment the `Place` and identify varied uses for parking and mobility-related operations or other purposes.

  — `IdentifiedArea` – an identifiable discrete bounded geographic zone that shares common characteristics and is used for parking and mobility-related operations or other purposes. IdentifiedAreas are segmented into four specializations: `PedestrianAccess` class, `VehicularAccess` class, `SpecificArea` class and `SupplementalFacilityArea` class.

  — `Space` – a single space for parking or other mobility-related purposes, usually designed for one vehicle, which may, but not necessarily, be denoted by painted marker or another road surface marker.

These specializations of place hierarchy (`HierarchyElementGeneral`) class (i.e. `Campus`, `Place`, `SubplaceElement`, `IdentifiedArea` and `Space`) support both the notion of different scales within the hierarchy as well as different forms of attribution, which are described later in this document.

The `Place` hierarchy shall be defined “top-down” with the highest layer being numbered 0 (zero) and each subsequent layer being numbered by incremental increasing integers.

The `Place` hierarchy can best be considered as a tree-like structure which starts from one common root element which will either be an instance of `Place` or, if used, `Campus`. Navigating down through the layers of the hierarchy, irrespective of which branch is taken, each subsequent layer sub-divides the previous. Each branch of hierarchy "tree" shall terminate in either an instance of `IdentifiedArea` or `Space`. Each branch of the hierarchy shall contain one instance of `Place`, and no more than one instance of `IdentifiedArea`.

A single instance of `Place` may support multiple branches, thus an instance of `Place` can have multiple instances of `IdentifiedArea` associated to it. If an instance of `Campus` is present, it shall only occur once in each branch of the hierarchy. See Figure 5, which provides an illustrative example of elements of a place hierarchy combined.

© ISO 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

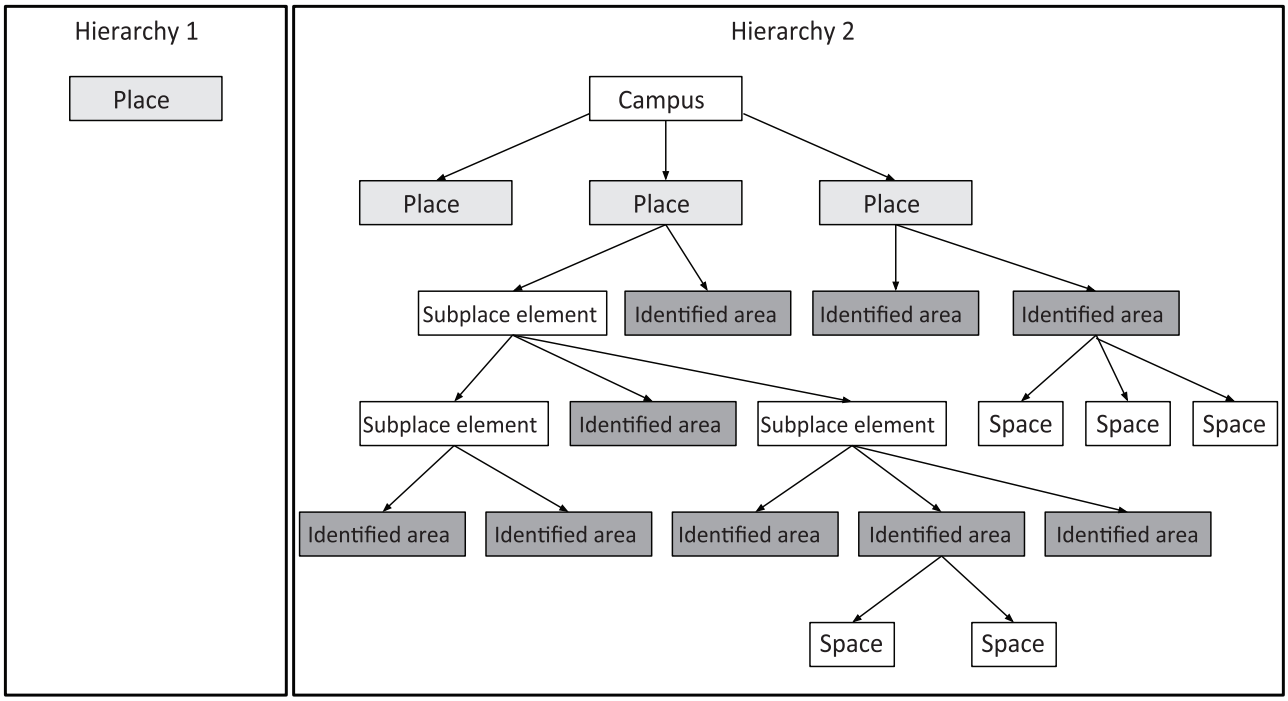


Figure 5 — Examples of place hierarchy

When describing the homogeneous characteristic for an entire place, the tree structure may terminate with the instance of `Place`, (see Hierarchy 1 in Figure 5) with no further child hierarchy elements (including `IdentifiedArea`).

If the place is sub-divided, each "branch" of the hierarchy tree shall terminate in one instance of `IdentifiedArea`, and the `IdentifiedArea` class shall appear only once in each branch of the hierarchy tree. Only instances of the `Space` class may exist at the levels of a hierarchy below the occurrence of `IdentifiedArea` in each branch of the hierarchy tree. Every branch of the hierarchy tree may use instances of the data concepts `Campus`, `Place`, `SubplaceElement`, `IdentifiedArea` and `Space` in the order stated. Instances of `SubplaceElement` may be repeated in any branch of the hierarchy, so long as this conforms to the order in the stated sequence.

6.2.1.3 Location hierarchy example

Top down:

Airport example

— Layer 0 – Campus (Airport)

— Layer 1 – Place (Short Stay Car Park) {others might be "long stay", "valet meet and greet", etc.}

— Layer 2 – SubplaceElement (Orange Level or Third Floor)

— Layer 3 – IdentifiedArea (Row J) - Type SpecificArea, with additional attributes

— Layer 4 – Space (74)

It is not necessary to build multi-layer hierarchies of `Place` data. Simple `Place` data needs can be represented without using the multiple layers available in the data specification.

6.2.1.4 Place

Figure 6 provides a UML class diagram for Place.

© ISO 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

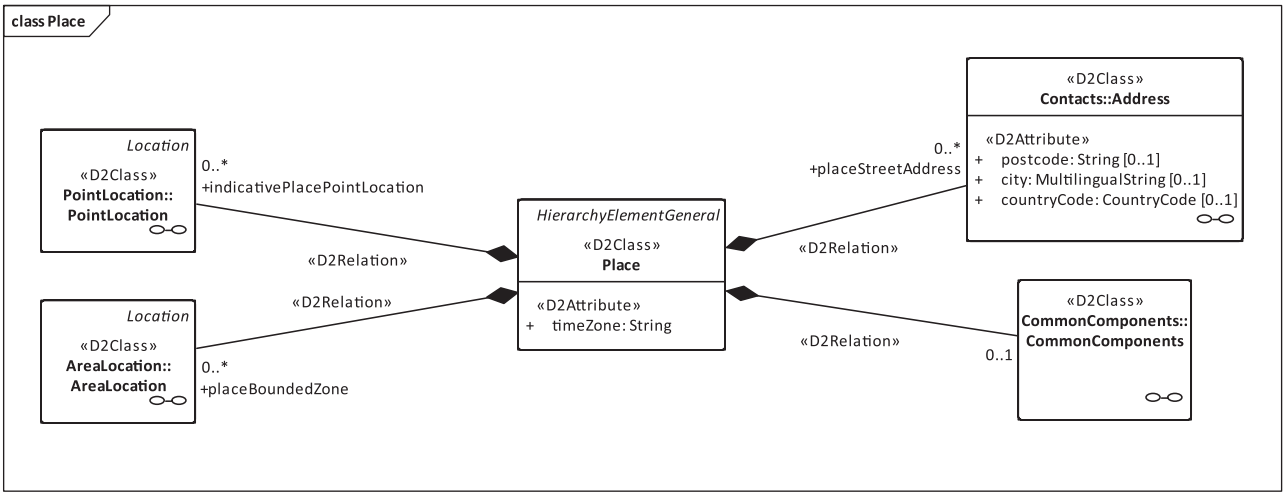


Figure 6 — Place

An instance of `Place` can have zero, one or several point locations that represent the notional point spatial location of the place. Additionally, an instance of `Place` can have zero, one or several area locations that represent the notional bounding polygon or zone for the place. An instance of `Place` can have zero, one or several street addresses. An instance of `Place` can access CommonComponents classes which support a wide range of optional attribution for contacts (which includes addresses), times (operating times, entrance opening times, exit opening times), location data, marketing content references, payment methods, operating restrictions, colour association (where parking facility elements are colour coded) and other characteristics (such as operating mode, type of the parking structure, access control, etc.)

Instances of `IdentifiedArea` and `Space` enable additional classes beyond the CommonComponents to be associated. `IdentifiedArea` is an abstract class, see Figure 8, which can be realized in four implementable specializations: `PedestrianAccess`, `VehicularAccess`, `SpecificArea, and SupplementalFacility`.

`SubplaceElement` and `Space` have a user-defined codelist related to occupancy level. This user-defined codelist enables a `Place` to define codes for various occupancy levels and the user-defined definition of each occupancy level. As an example, a place or space may identify occupancy levels using user-defined colour codes and define each colour code as follows:

User-defined Code 1: Green

User-defined Code 1 Definition: Occupancy utilization is less than 80 % of Supply

User-defined Code 2: Yellow

User-defined Code 2 Definition: Occupancy utilization is less than 85 % of Supply

User-defined Code 3: Red

User-defined Code 3: Definition: Occupancy utilization is equal to or greater than 95 % of Supply

6.2.1.5 SubplaceElement

Figure 7 provides a UML class diagram for `SubplaceElement`.

© ISO 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

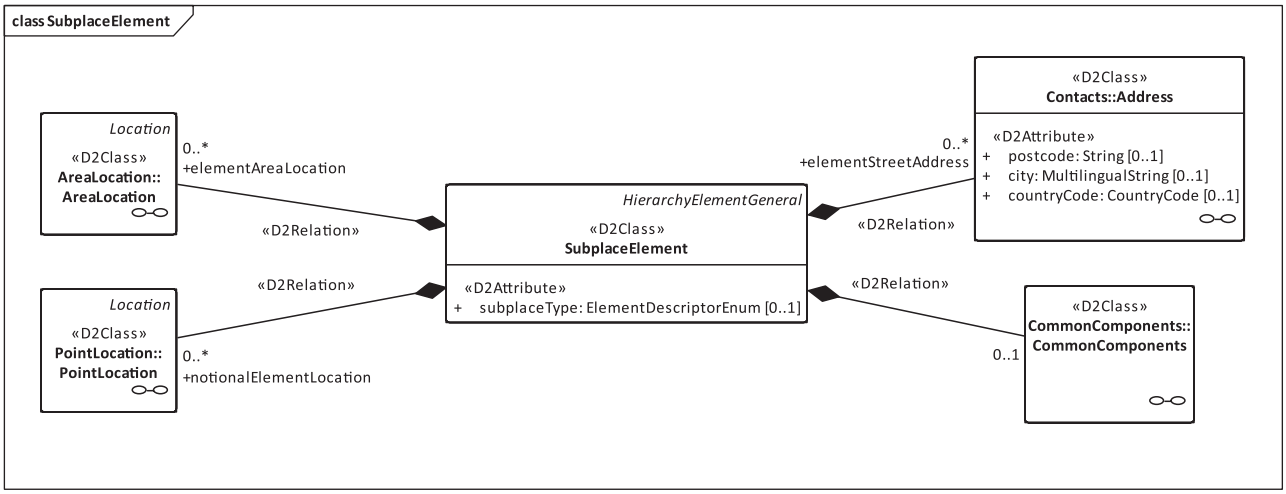


Figure 7 — SubplaceElement

6.2.1.6 IdentifiedArea

Figure 8 provides a UML class diagram for `IdentifiedArea`.

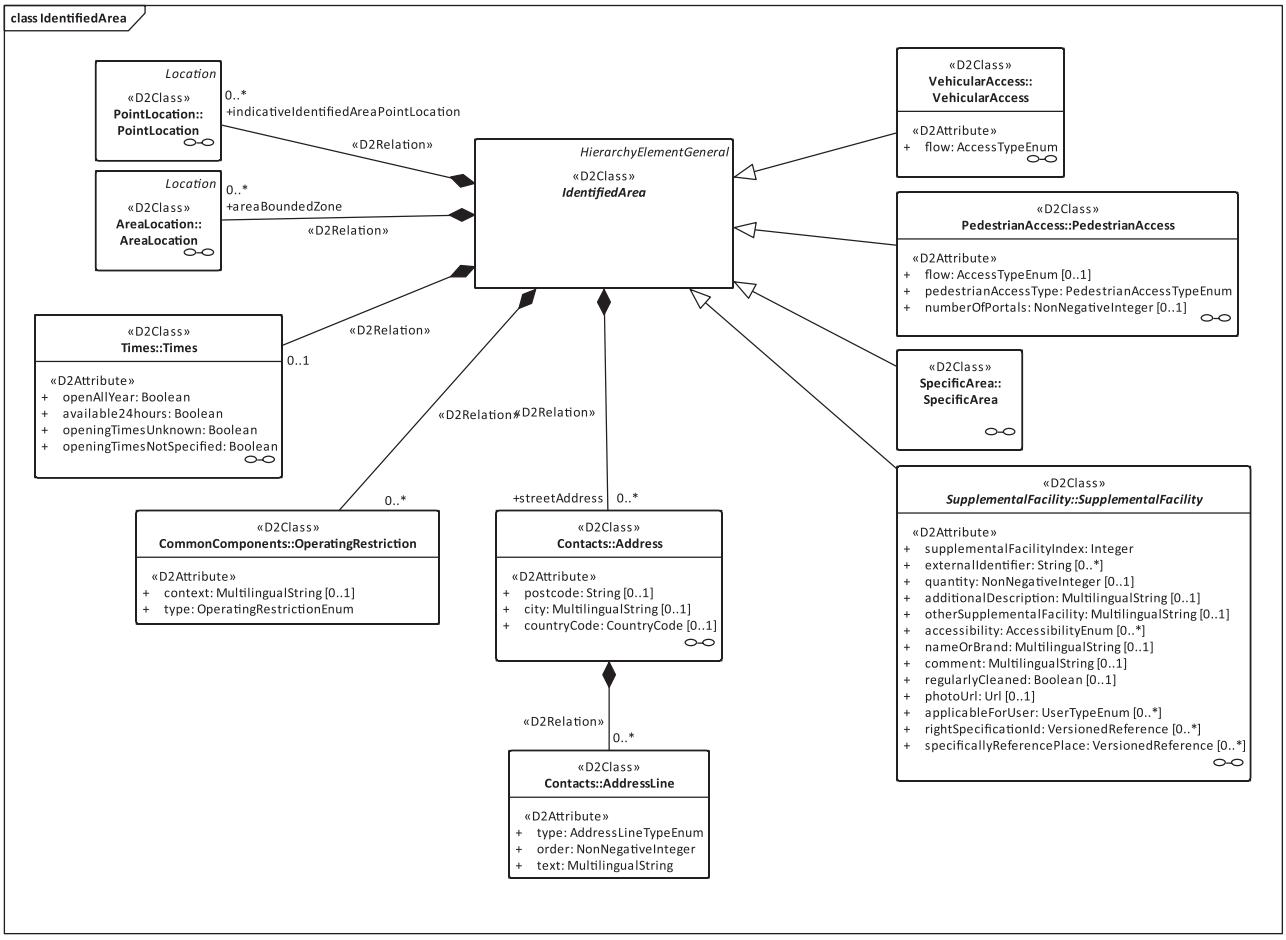


Figure 8 — IdentifiedArea

© ISO 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

6.2.1.7 Space

Figure 9 provides a UML class diagram for `Space`.

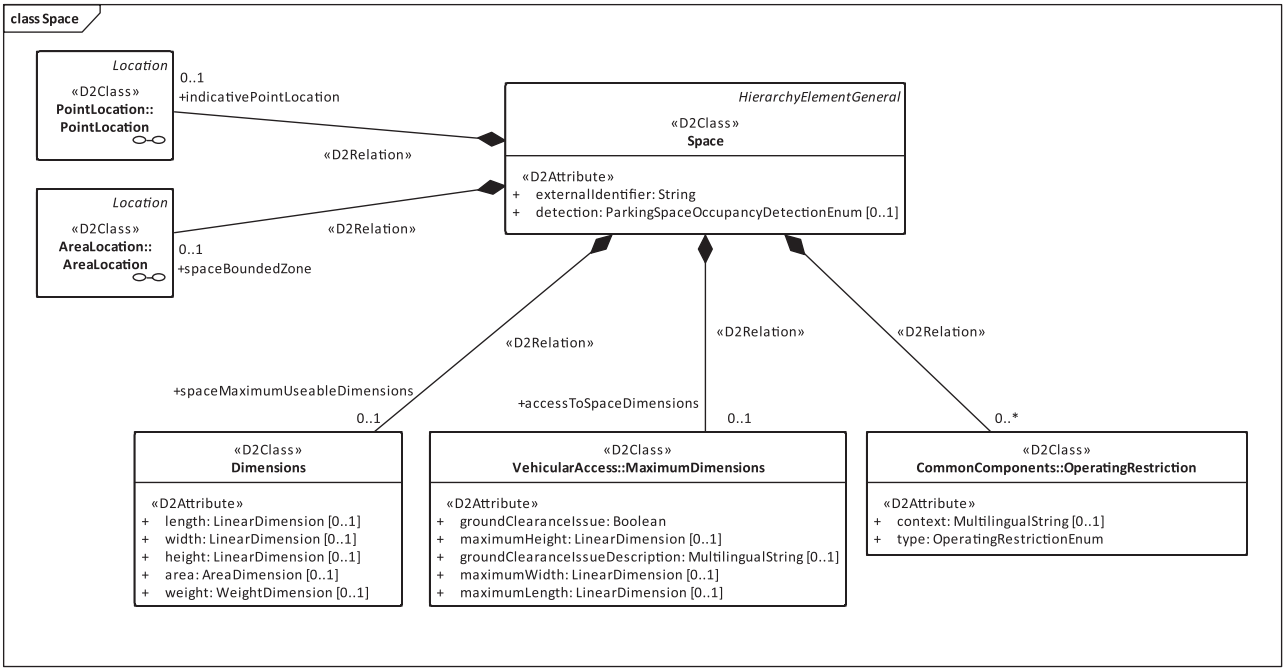


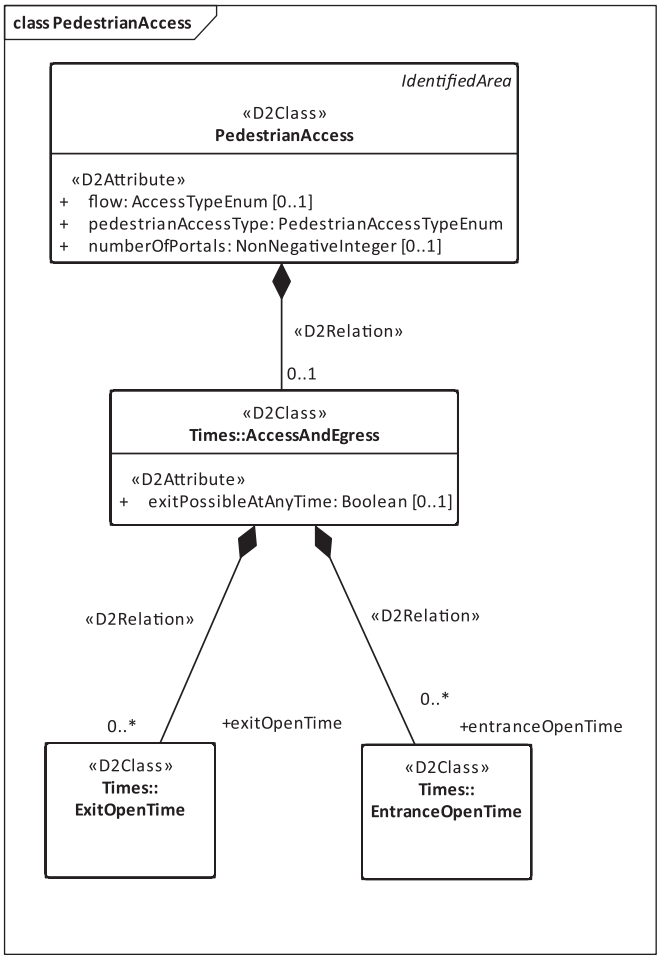
Figure 9 — Space

6.2.2 IdentifiedArea sub-types

6.2.2.1 Pedestrian access

Figure 10 provides a UML class diagram for `PedestrianAccess`.

© ISO 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023



NOTE This model only supports the modelling of vehicular and pedestrian accesses at the present time. Other forms of access can be introduced during a later revision if a stakeholder requirement is identified.

Figure 10 — PedestrianAccess class diagram

PedestrianAccessType enumerations (Pedestrian`AccessTypeEnum`) for `PedestrianAccess` support the definition of characteristics for access in and out from a place or part thereof respectively (denoted by the pedestrianAccessType attribute, using the PedestrianAccessType enumerations). `PedestrianAccess` enables an entity to identify the quanity of each access type as well as the operating times the `PedestrianAccess` is available for use.

6.2.2.2 Vehicular access

Figure 11 provides a UML class diagram for `VehicularAccess`.

© ISO 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

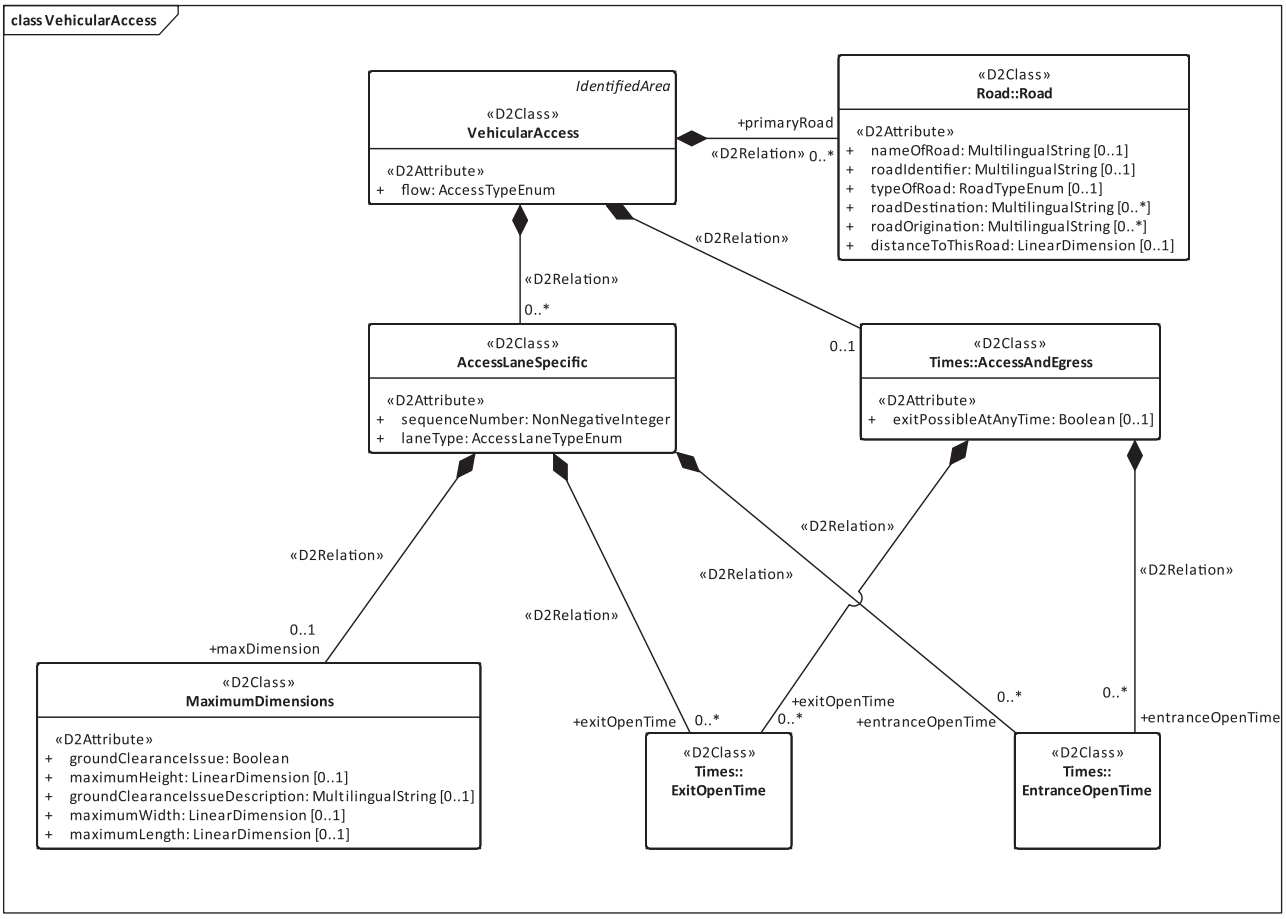


Figure 11 — VehicularAccess class diagram

AccessType enumerations (`AccessTypeEnum`) for `VehicularAccess` support the definition of characteristics for the how vehicles flow to, from and both to and from a place or part thereof respectively (denoted by the flow attribute, using the AccessType enumerations). Each of the AccessLaneType enumerations (`AccessLaneTypeEnum`) may support the definition of characteristics for each lane within the instance of VehicularAccess. Preferred practice gives a sequence number to each lane left to right when entering the facility.

`VehicularAccess` also supports a macro view of access by enabling an entity to reference access or approaches to a facility by defining primary roads near the place and/or identifying primary routes from a highway and/or nearby significant crossroads to define the approximate location of the vehicular access point. This is accomplished via instances of `Road` and `RodeNode` classes, see Figure 12.

© ISO 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

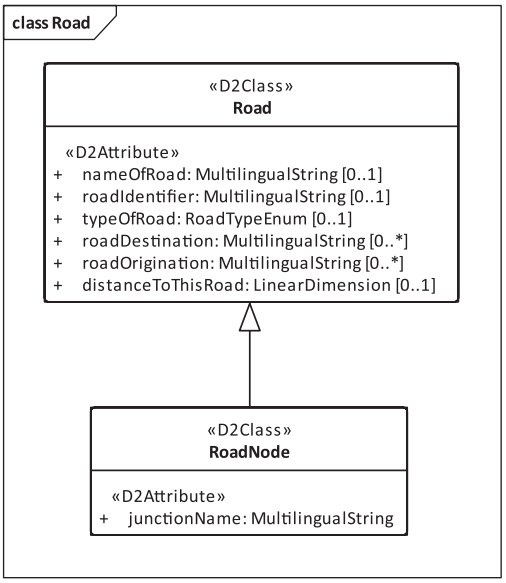


Figure 12 — Road class diagram

6.2.2.3 Specific area

Figure 13 provides a UML class diagram for `SpecificArea`.

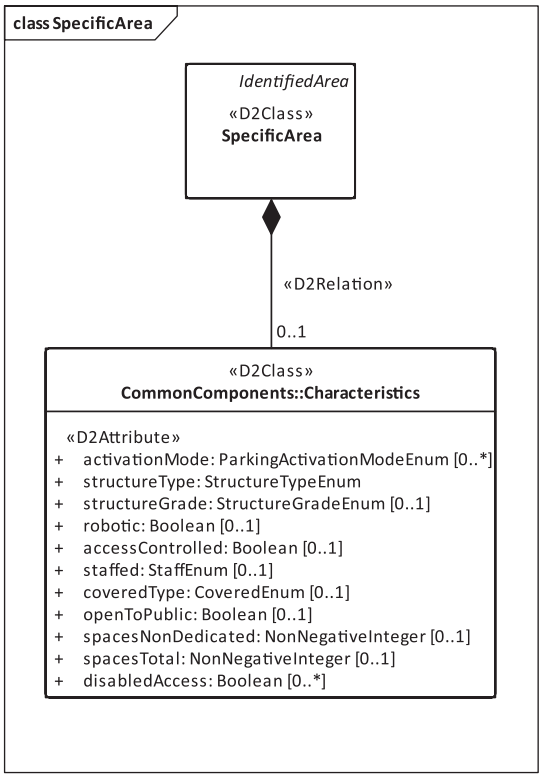


Figure 13 — SpecificArea class diagram

`SpecificArea` class is a specialization of `IdentifiedArea` that denotes a specific geographic area in a place that has a common physical purpose of use and common characteristics. Examples of common physical purpose of use include parking area, delivery zone, ride share pickup zone, and valet. `SpecificArea` data concept describes the physical components of a `Place`. `SpecificArea` specialization of `IdentifiedArea` is used to assign `Rightspecifications`.

© ISO 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

6.2.2.4 Supplemental facility

Figure 14 provides a UML class diagram for `SupplementalFacility`.

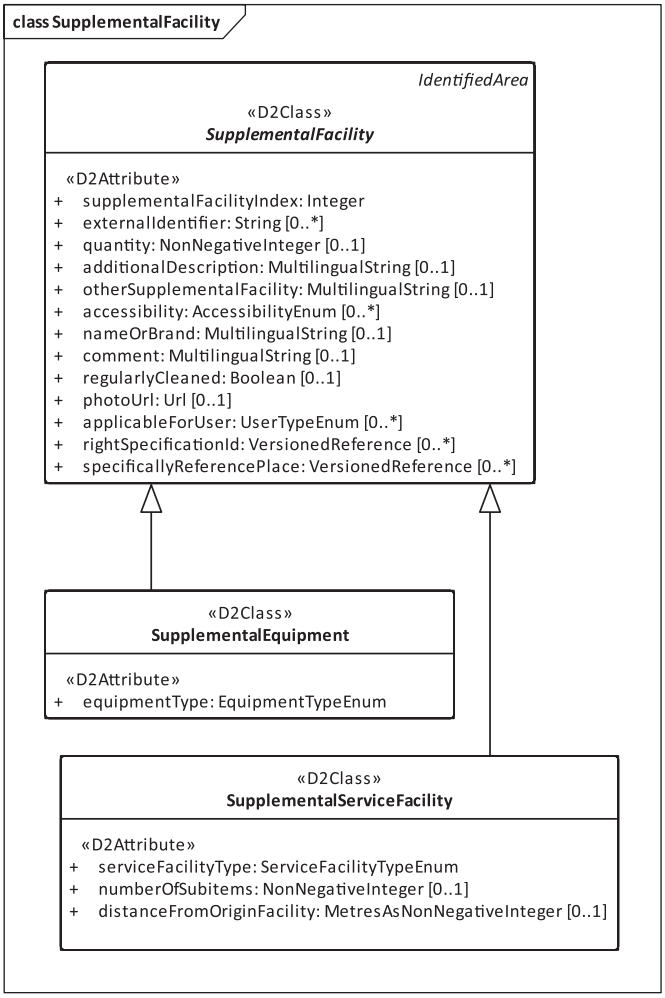


Figure 14 — SupplementalFacility class diagram

`SupplementalFacility` is a specialization of `IdentifiedArea` data concept that denotes a specific geographic area in a `Place` that has specific services and equipment available. This may include showers, meeting rooms, restrooms/toilets, lockers and other services.

The `SupplementalFacility` specialization relies on the `RightSpecification` to define eligible users or vehicles of the defined equipment and services. Within the `SupplementalFacility` specialization, further classes are available to detail Equipment (`SupplementalEquipment`) and Services (`SupplementalServcesFacility`). In the `SupplementalEquipment` class, an entity can identify a specific type of equipment and quantity available; this can include electric charging stations, payment stations, bike storage, etc. Additional details for specific equipment can be further defined in future revisions of the specification. The data model currently includes a data class for ElectricChargingPoints which allows for more detail to be provided for one-to-many charging stations. The `SupplementalServiceFacility` class similarly allows a `Place` to identify various services available in the `IdentifiedArea`. Services include restroom, shower, food store and many other services.

6.2.3 Common components

`CommonComponents` represent a collection of common attributions that may be applied at several points in the `Place` hierarchy.

© ISO 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

`CommonComponents` provides optional support for a range of attribution including times (operating times, entrance opening times, exit opening times) [defined in the <PkCommon> namespace], marketing content references, payment methods, operating restrictions, colour association (where parking facility elements are colour coded) and other characteristics (such as operating mode, type of the parking structure, disabled parking characteristics, access control, etc.).

Figure 15 provides a UML class diagram for `CommonComponents`.

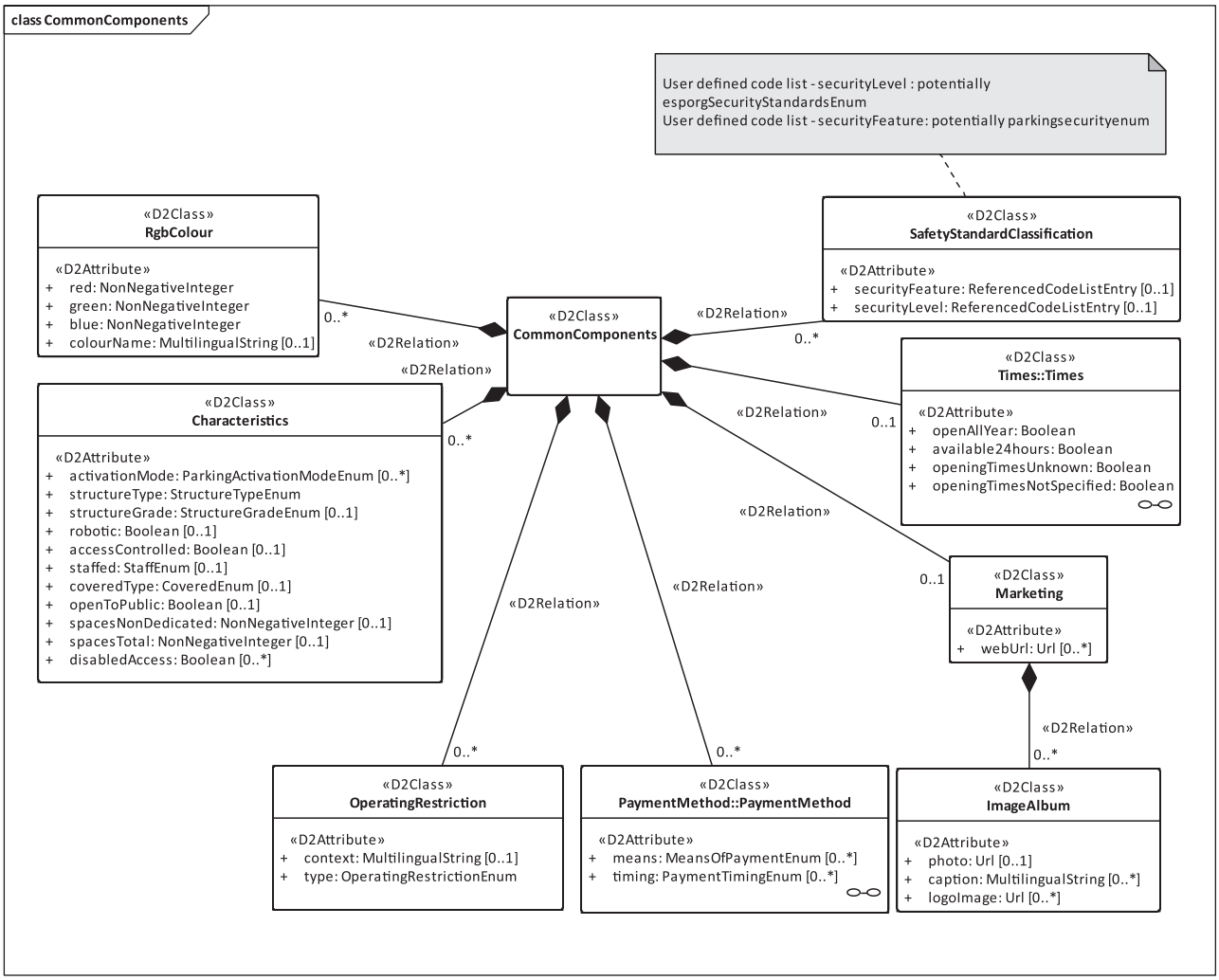


Figure 15 — CommonComponents class diagram

6.2.4 Payment method

Figure 16 provides a UML class diagram for `PaymentMethod`.

© ISO 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

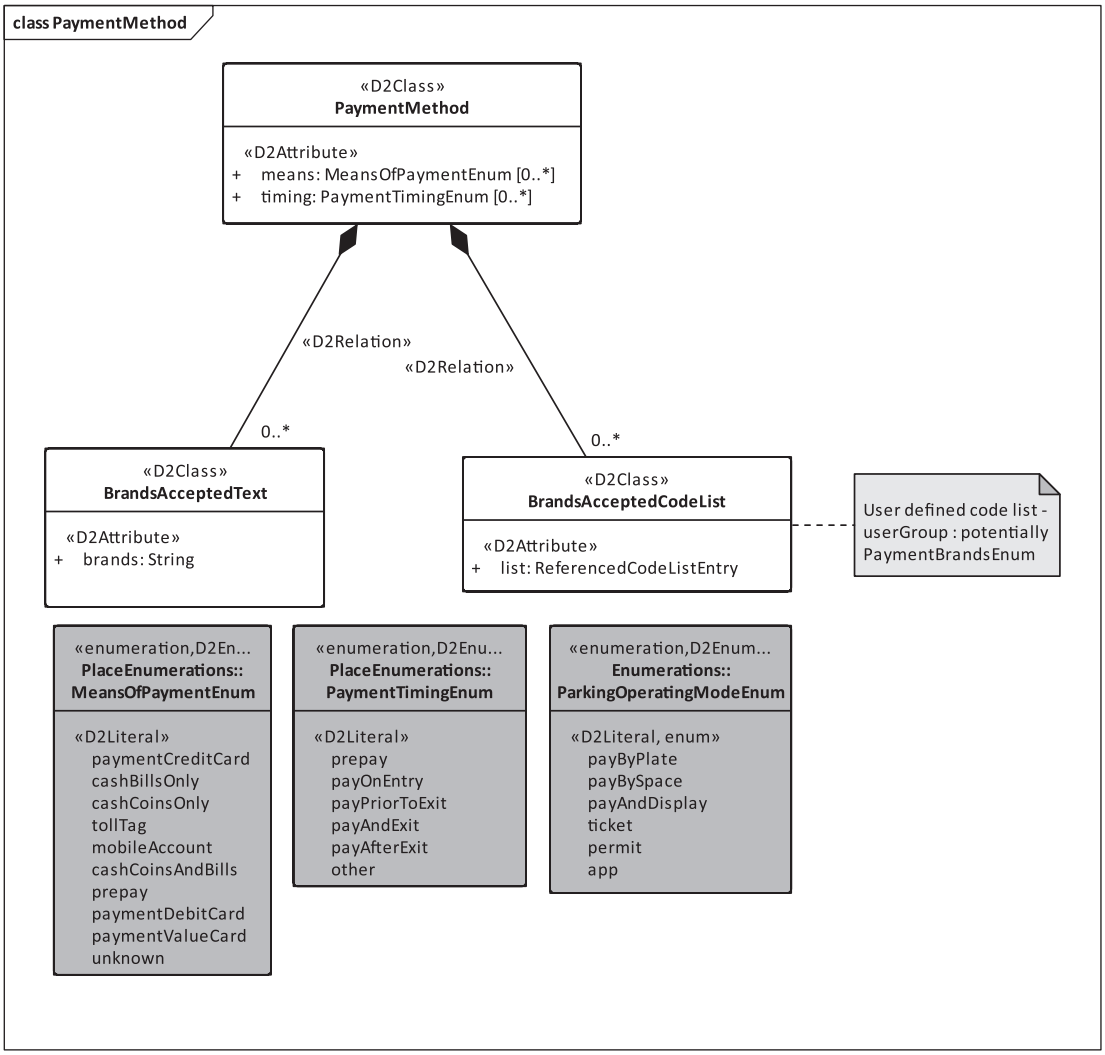


Figure 16 — PaymentMethod class diagram

6.2.5 Basic elements (in place)

6.2.5.1 Overview

The following concepts define data types, enumerations and common classes defined and used specifically in the <Place> namespace.

More widely used data concepts, data types, enumerations and common classes are defined in 6.9.

6.2.5.2 Data types (for place)

No specific data types are defined in <Place> namespace. For general data types, see 6.9.2.

6.2.5.3 Enumerations (for place)

The following enumerations are defined in the <Place> namespace, see Figure 17.

© ISO 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

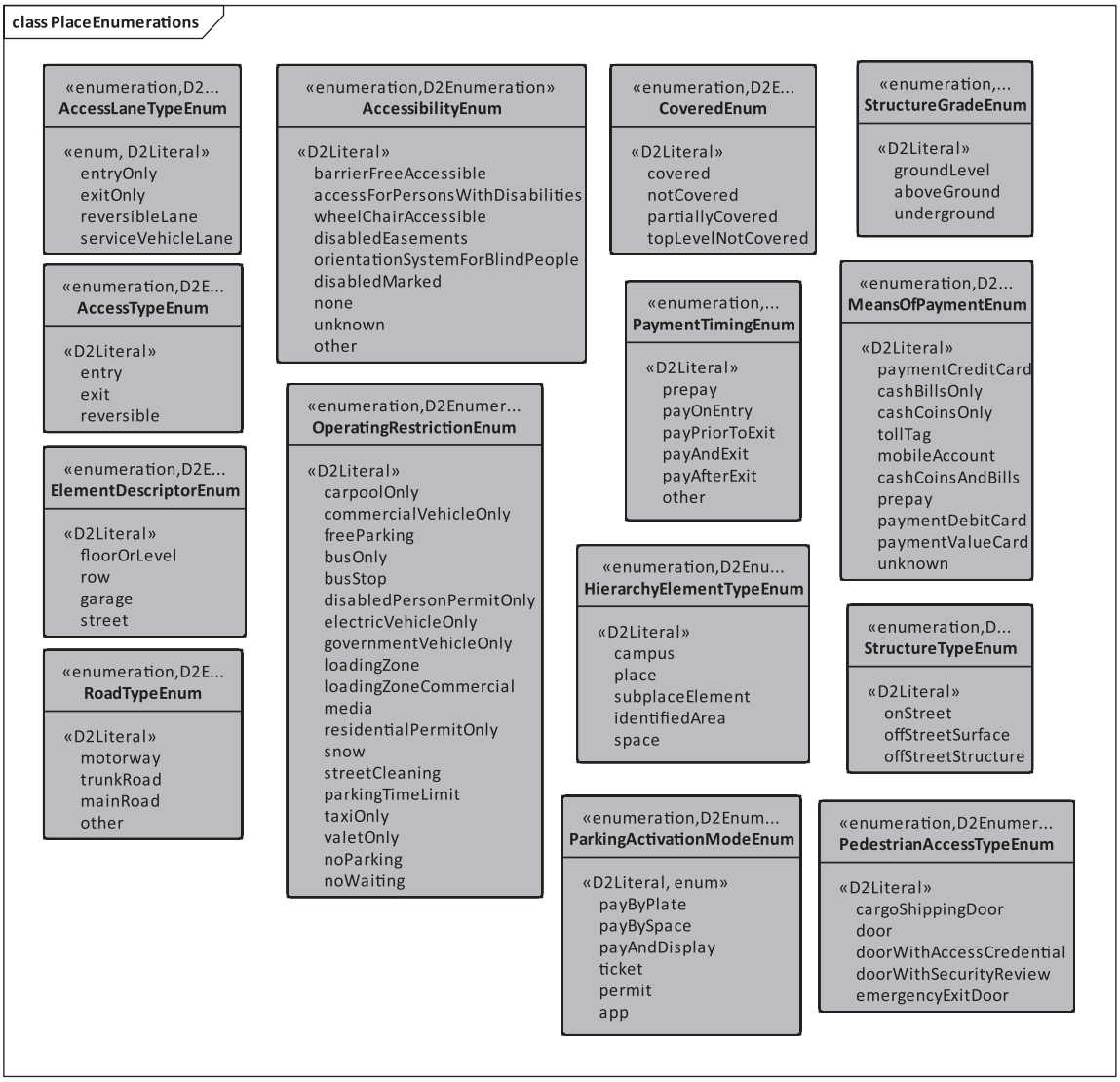


Figure 17 — Enumerations in the place domain

6.2.5.4 External codelists (for place)

An external codelist exists for <Place> which enables a place owner to define a custom list of occupancy levels as a code with accompanying definitions for each occupancy level code. For general external codelists, see 6.9.5.

No specific external codelists are defined in <Place>. For general external codelists, see 6.9.5.

6.3 Occupancy

6.3.1 Introduction to occupancy

This section describes concepts used to define occupancy supply and demand for a specified place.

— `Supply` shows space inventory allocated for use (i.e. for parking, delivery, etc.).

— `Demand` shows the usage of space inventory.

These are defined in the <Occupancy> namespace of the data model.

© ISO 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

6.3.2 Supply

Figure 18 shows the supply side of the <Occupancy> namespace. Through the `HierarchyElementReference` class, the `Supply` data class are linked to an identifiable element within the `Place` hierarchy.

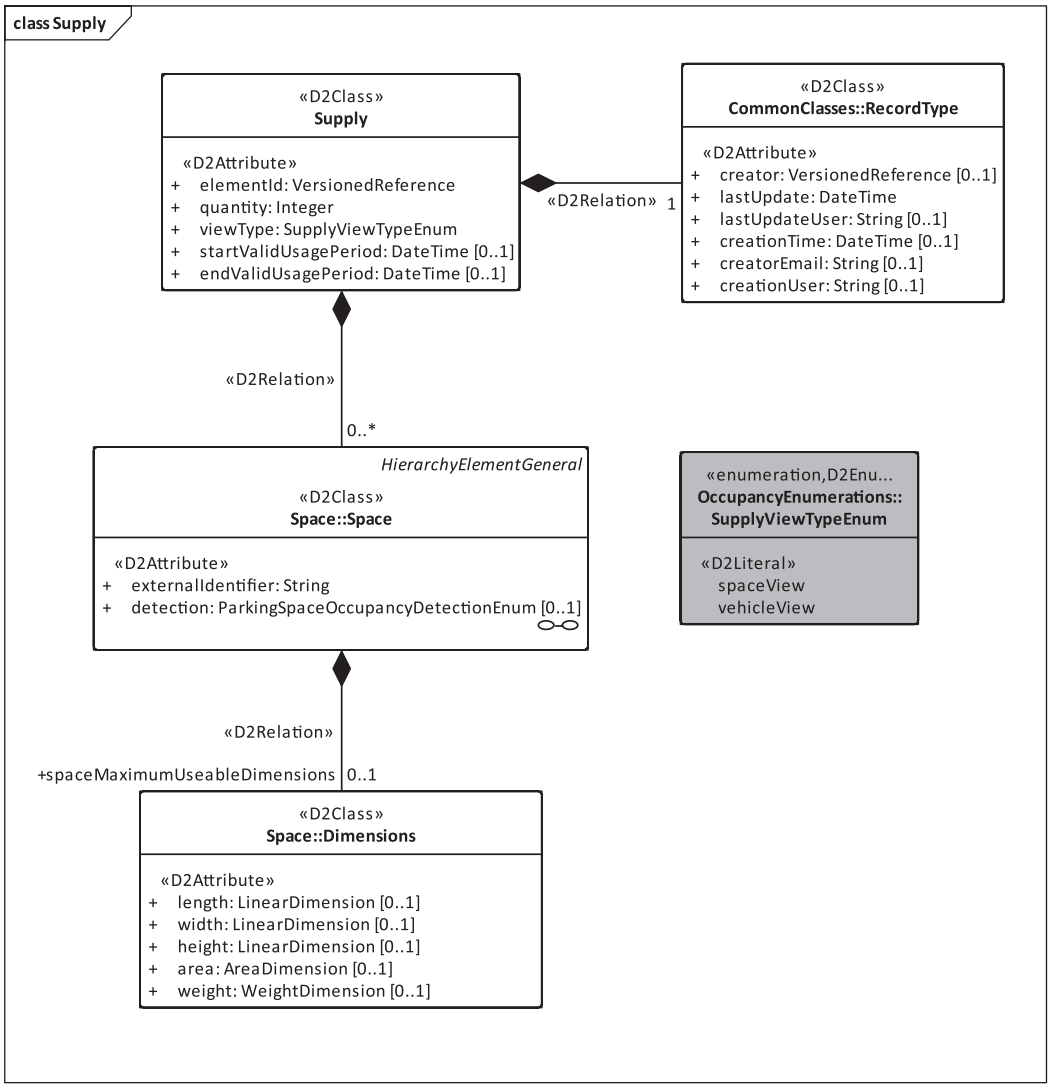


Figure 18 — Supply class model

The `Supply` class defines the supply quantity as an integer, relevant for a defined period. If the Supply.startValidUsagePeriod attribute is not set, the SupplyQuantity is considered to be relevant from the present time until the Supply.endValidUsagePeriod date/time. The SupplyViewType attribute enables identification of different perspectives of specifying supply:

— spaceView - means that each space is physically marked and identifiable.

— vehicleView - means that vehicle capacity in an area is estimated based on calculating a capacity via a defined method (e.g. curb length divided by a length of vehicle, or the holding capacity of a specific parking area). The distributing party decides on the appropriate means to calculate the vehicleView (e.g. length of vehicle to use).

The `Supply` class can be linked to one-to-many identified `Spaces` whose characteristics can be defined through the `Dimensions` class which provides maximum useable dimensions for a space.

© ISO 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

6.3.3 Demand

Figure 19 shows the demand side of the <Occupancy> namespace. Through the `HierarchyElementReference` class, the `DemandTable` class are linked to an identifiable element of the `Place` hierarchy. Demand being the quantity of available capacity in use.

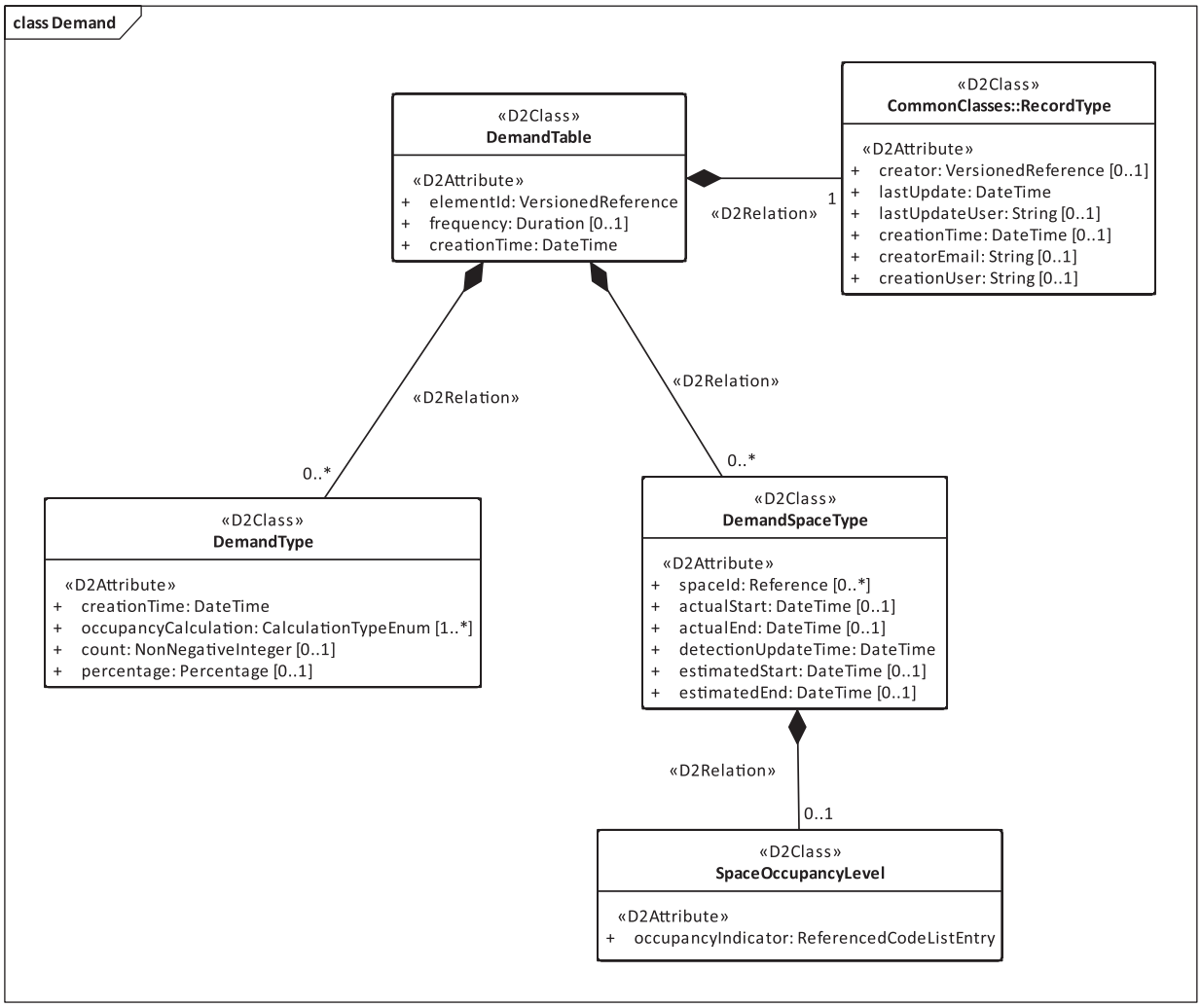


Figure 19 — Demand class model

The `DemandType` class indicates the timestamp of the associated demand characteristics, and the frequency at which the demand characteristics are updated.

The `DemandType` class provides the numeric value or the percentage of this demand compared to the total supply and the specific date/time stamp of when the value or percentage is determined. Only one numeric value or percentage shall be provided. The `DemandType` class also supports identification of how the demand is calculated through the occupancyCalculation attribute.

The `DemandSpaceType` class enables the demand to be defined relative to a specific identified `Space`. This can be used to indicate historical demand with the actualStart and actualEnd being used to provide actual times of occupancy. Current demand uses either the actualStart or estimatedStart to define the known or estimated start of the `Session` respectively. Future demand can be defined by use of the estimatedStart and estimatedEnd to define the estimated `Session`.

6.3.4 Enumerations (for occupancy)

The following enumerations are defined in the <Occupancy> namespace. See Figure 20.

AVA Documents
Document Sharing Hub

© ISO 2023 – All rights reserved

Full document details: ISO TS 05206-1-2023

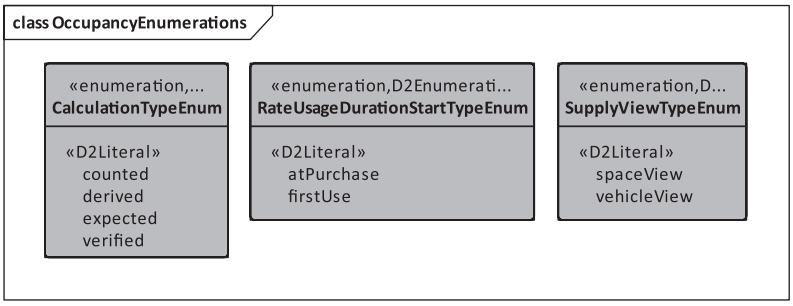


Figure 20 — Enumerations in the occupancy domain

6.3.5 External codelists (for occupancy)

No specific external codelists are defined in <Occupancy>. For general external codelists, see 6.9.5.

6.4 Rates

6.4.1 Introduction to rates

This subclause describes concepts used to define rates and tariffs for a specified `Place` element. Rates can be associated to a variety of activities including parking, delivery, loading, unloading, crossing an access point, etc.

These are defined in the <Rates> namespace of the data model.

6.4.2 Rate tables

Figure 21 details the data structure related to <Rates>, which are used to convey information concerning tariffs and fee schedules relating to an identifiable element of the `Place` hierarchy through use of the `HierarchyElementReference`.

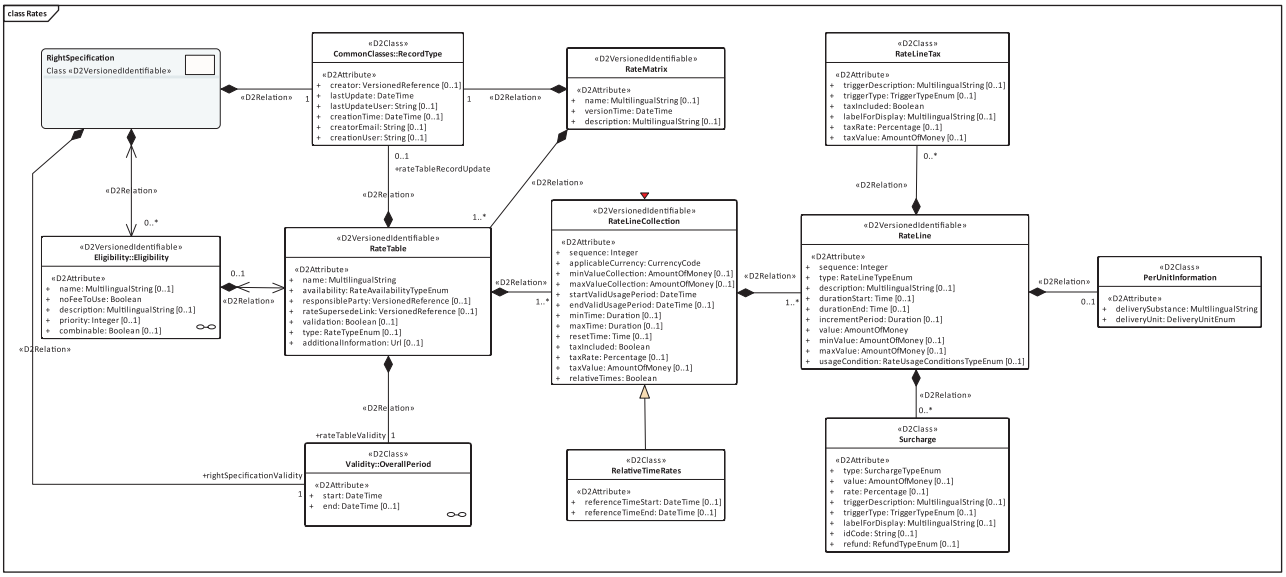


Figure 21 — Rates domain class diagram

A `RateTable` represents a set of charges that are applied to a single `RightSpecification` for parking or other operating activities (e.g. delivery permits, rideshare access, etc.) at the `Place`. Examples of

© ISO 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

a `RateTable` are a weekday charge rate scale in a public, multi-storey car park or evening rates and weekend rates.

The `Validity` class associated to the Rate table defines two important time periods related to a Rate table:

— `overallPeriod.start` and `overallPeriod.end` identifies when the Rate table and associated attributes are valid for usage. This enables an entity to schedule the release of new ratetables and communicate when current ratetables will expire. If the `overallPeriod.end` is not set, the `RateTable` is considered to be valid until it is replaced

— `period.startOfPeriod` and `period.endOfPeriod` define when the fees and tariffs defined in the Rate table are available during a specific period to be applied to a `Session`.

To support the transmission of a `RateTable` that may contain multiple charging elements; a `RateTable` contains one to many `RateLineCollection`(s). An example could be a `RateTable` that contains a flat rate fee for a reservation, plus a tiered time-based rate structure for charging during the `Session`. Each `RateLineCollection` represents one of those charging elements. A `RateLineCollection` is constructed of one to many `RateLine`.

The `RateLine` concept is flexible and supports a range of different characterizations, which include the following.

— Flat rate: where the `Rateline` is active, the applied fee charge is a flat rate, unrelated to the duration or timing of the parking or other type of `Session`; an example is a flat rate reservation fee for a `Session`. Flat rate `RateLine` are defined by use of a defined value, but no use of the durationStart, durationEnd or incrementingPeriod attributes.

— Flat rate tier: where the applied fee charged is charged in full if the parking or other type of `Session` indicates that that specific `RateLine` is active. For example, in the second hour of a `Session` the fee is USD 1,00 (USD) for any part of that hour. For a flat rate within a tier, the `RateLine` defines the time boundary of the tier by use of the durationStart and durationEnd attributes and the value attribute to define the charge amount. If used, the incrementingPeriod attribute, in this case, shall be the same as the period between the durationStart and durationEnd (i.e. there is one increment). Charging is assumed to occur at the start of each increment.

— IncrementingRate: where the applied fee charged is related to the duration of the specific tier activated by the parking or other type of `Session`. This charge type supports a `RateTable` that applies for short incrementing periods or time-based small increments of charge. For example, in the second hour of a `Session`, charging is done at a rate of EUR 0,05 every 3 minutes. For an incrementing rate within a tier, the `RateLine` defines the time boundary of the tier by use of the durationStart and durationEnd attributes. The incrementingPeriod and the value attribute indicate the charge amount of each increment (e.g. EUR 0,05 each 3 minutes). Charging is assumed to occur at the start of each increment.

— Per unit: where the applied fee charged is based on a per unit of measure where the unit of measure is user-defined. The user-defined unit is required to be specific and explicit in definition. This user-defined unit of measure could be kWh to collect fees for electric vehicle charging or it could be a passenger count to support a fee structure similar to USD 5,00 per passenger in a car.

Under most circumstances the start and end of charging periods are fixed and relative to local time (e.g. between 08:00 and 17:00 weekdays). In some instances, the charging period and related tiers can be relevant to a specific event. The reference of a charging period time to another event is is indicated by the use of the relativeTimes set to TRUE in the `RateLineCollection` and the use of the `RelativeTimeRates` class. All times are defined relative to the `referenceTimeStart`, which is the start time of the reference event

The applicable currency is defined in the `RateLineCollection`.

Individual `RateLine` support the identification of whether tax is applicable within the defined `RateTable` or applied in addition to the defined `Rate`. The value of tax, if included, can be specified as either a

© ISO 2023 – All rights reserved

AVA Documents Document Sharing Hub
Full document details: ISO TS 05206-1-2023

monetary amount or a percentage rate. Taxes may also be applied to a `RateLineCollection` in a similar manner. It is common practice for taxes to be applied at the `RateLine` level, for example the application of value-added tax (VAT) in Europe which is added to a basic parking fee and declared in the cost of the parking to the end user.

A `RateLineCollection` indicates whether the child `RateLine` are a chargeable tariff or represent a surcharge, which may be partially or fully refundable.

A `RightSpecification` is linked to `Eligibility` without reference to a `RateTable` when no `RateTable` is applicable (i.e. the `RightSpecification` has eligibility constraints but no cost is associated).

6.4.3 Eligibility

Each `RateTable` is applicable to a singular set of criteria and a `RightSpecification`. The specification of the qualifying criteria is specified using `Eligibility`, see Figure 22. Typically, `Eligibility` is defined at the `RightSpecification`. However, if within a single `RightSpecification` access to specific `RateTable` requires further qualification, additional `Eligibility` criteria are defined at the `RateTable`.

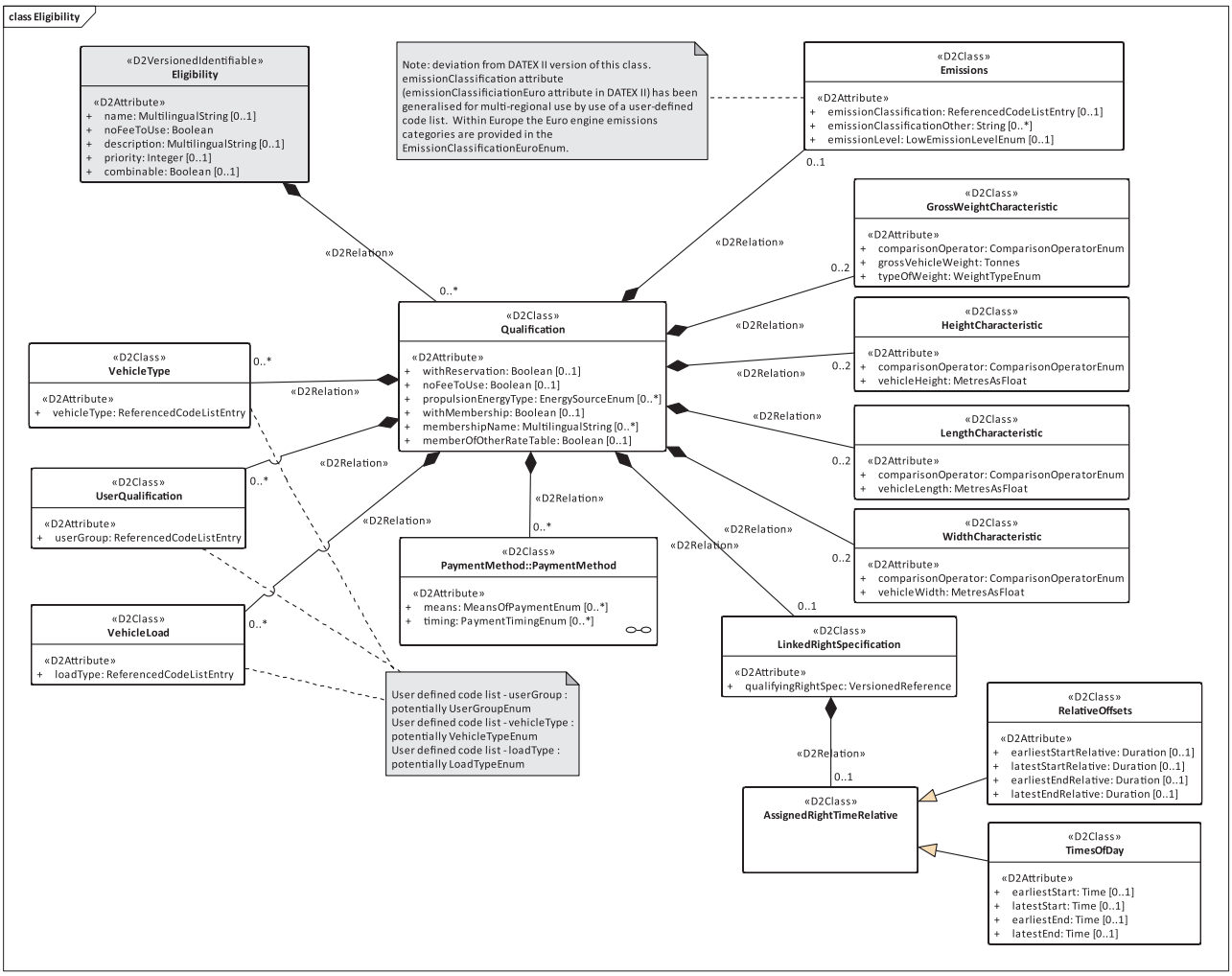


Figure 22 — Eligibility class diagram

`Eligibility` is specified as a collection of individual Qualifications. A Qualification is specified as a test with any of the attributes in the `Qualification` class set. In addition, Qualifications can be specified for other criteria, including the vehicle's emissions classes, maximum and/or minimum values for gross vehicle weight, load types (passenger, goods, hazardous materials), heights, widths and length measures.

© ISO 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

`Eligibility` may be related or defined by membership. Typically, the `Qualification` in this case is defined by the `withMembership` attribute set to TRUE and the `membershipName` attribute set to the names of the relevant memberships (e.g. J-Park frequent users club, shoppers, cinema attendees, event ticket holders, military, industry specific vehicle, etc). `Eligibility` may also be based on a user-defined codelist. This allows an entity to define specific lists relevant to their operations to be shared. User-defined codelists are available for `Emissions`, `UserGroup`, `VehicleLoad` and `VehicleType`.

Furthermore, `Eligibility` may be defined in relation to a specified active use of a `RightSpecification` (`linkedRightSpecification` attribute). The attribute `linkedRightSpecification` identifies the `RightSpecification`, and when used in an `AssignedRight` associated to a `Segment`, qualifies the user for an additional `RightSpecification` or `RateTable` (i.e. an active use of a `RightSpecification`, such as parking a car). This attribute enables an entity to define a `RateTable` that is eligible for use if a `RightSpecification` with a specific `RateTable` has been used. (For example, because a parker paid to park this morning in Zone 1, the parker is eligible to park in the afternoon in Zone 1 or Zone 2.)

`Eligibility` may be specified on a time-restricted basis (`AssignedRightTimeRelative`) with respect to an existing held `RightSpecification (which is identified` via possession of an `AssignedRight)`. `AssignedRightTimeRelative` identifies the duration of time for which a specific `AssignedRight` can be referenced to access a `RateTable` via the `RelativeOffsets` class (for example, Rightholder is eligible to use a new specific `RateTable` if they have used a RightSpecification within 2 hours of requesting access to the new `Ratetable`. The time constraints are defined in the `AssignedRightRelative` and/or specific times of day for the offset). When access to the qualification criteria exists, it can be defined via `TimesOfDay` class (for example, Rightholder is eligible to use a new specific `RateTable` if they have used a RightSpecification within 2 hours of requesting access to the new `RateTable` between 08:00 and 14:00).

6.4.4 Enumerations (for rates)

The following enumerations are defined in the <Rates> namespace. See Figure 23.

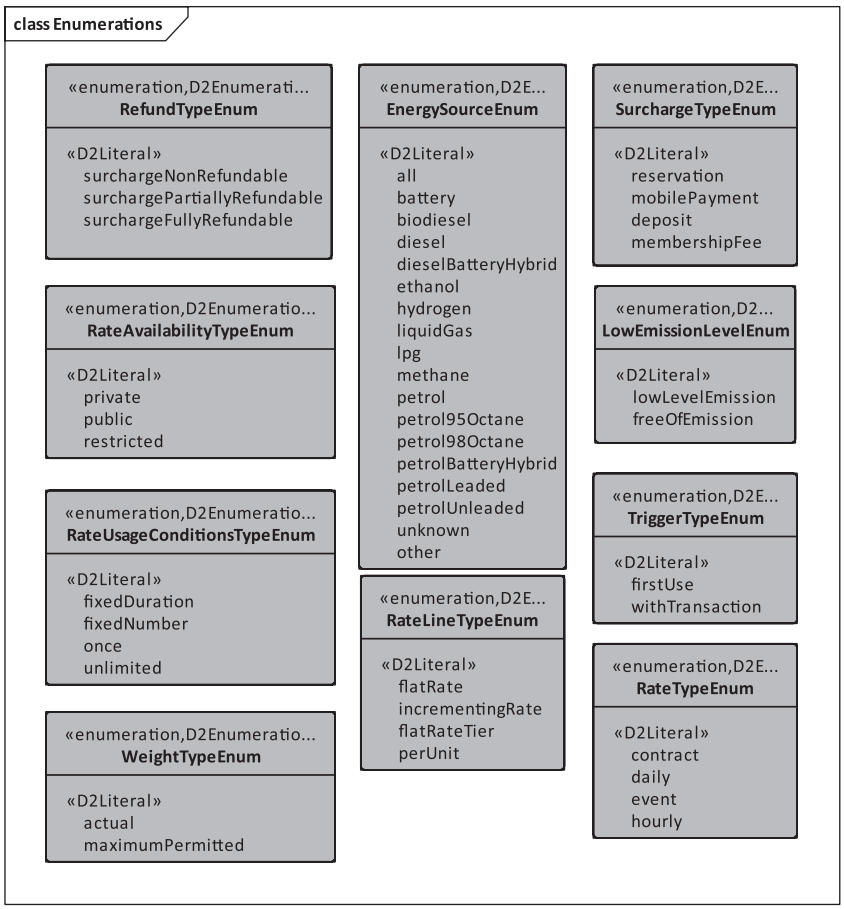


Figure 23 — Enumerations in the rates domain

© ISO 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

6.4.5 External codelists (for rates)

Multiple external codelists exist for <Rates> which enables a place owner to define a custom list for the following.

1) Emissions: associate to a country or geographic standard list of emission codes for vehicles.

2) UserGroup: define a list of relevant user groups for the Place that defines qualification criteria for a RateTable or RightSpecification.

3) VehicleType: define a list of relevant vehicle types for the Place that defines qualification criteria for a RateTable or RightSpecification.

4) VehicleLoad: define a list of relevant vehicle loads for the Place that defines qualification criteria for a RateTable or RightSpecification.

For general external codelists, see 6.9.5. Also see Annex D.

6.5 Right

The data concepts with the <Right> domain are presented in Figure 24. Figure 25, Figure 26 and Figure 27 provide submodels for the <Right> domain.

© ISO 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

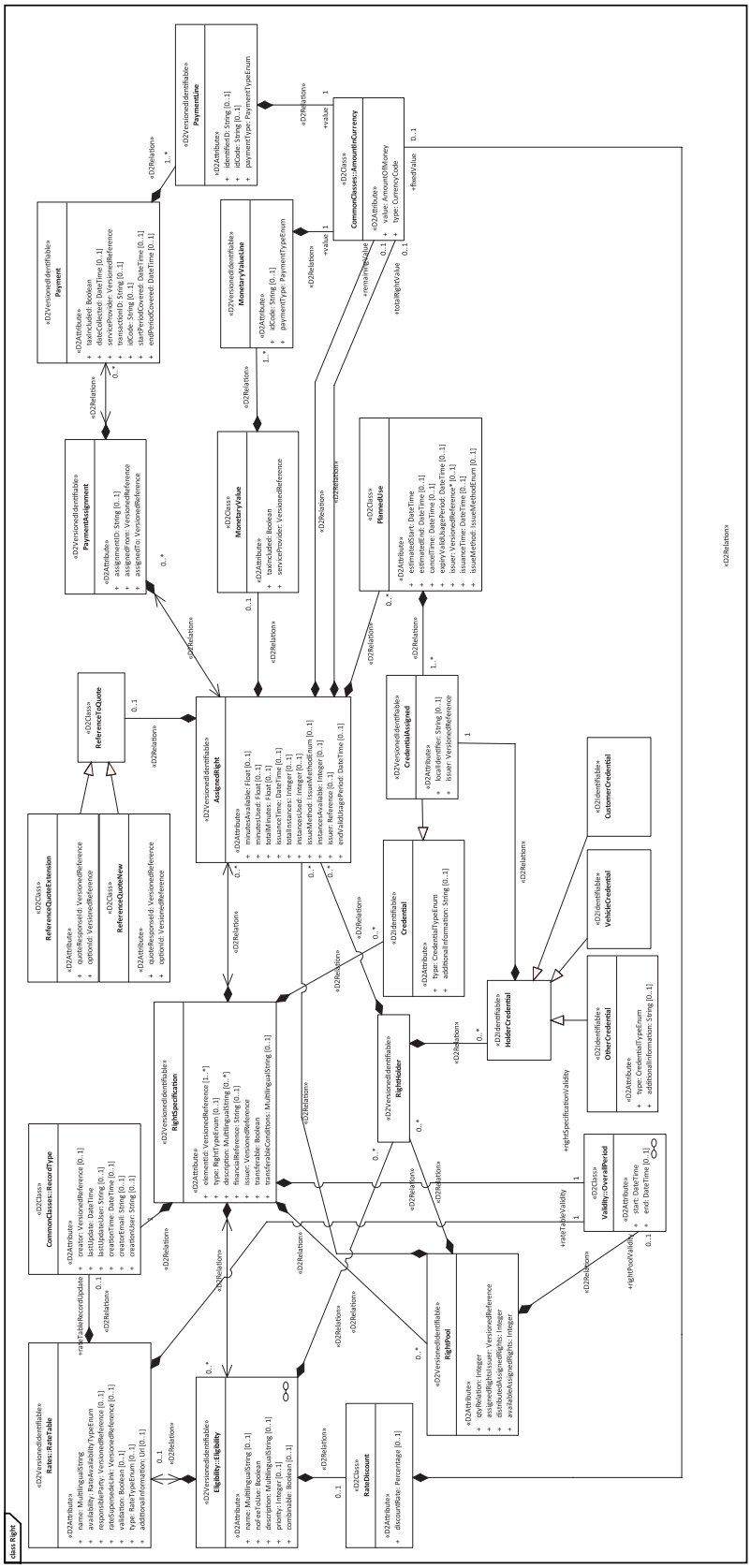


Figure 24 — Right domain model (overview)

© ISO 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

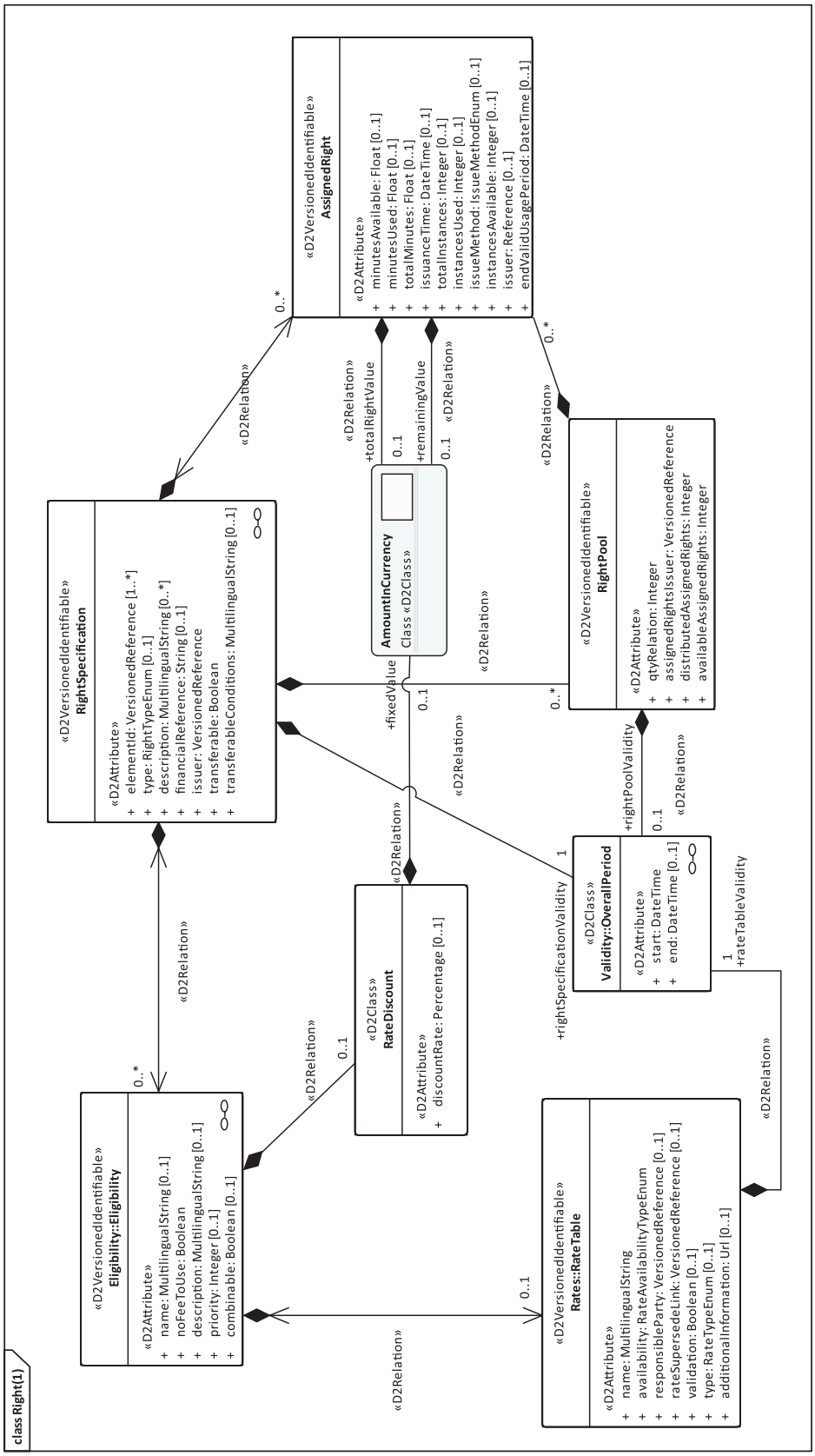


Figure 25 — Right domain model – submodel 1

© ISO 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

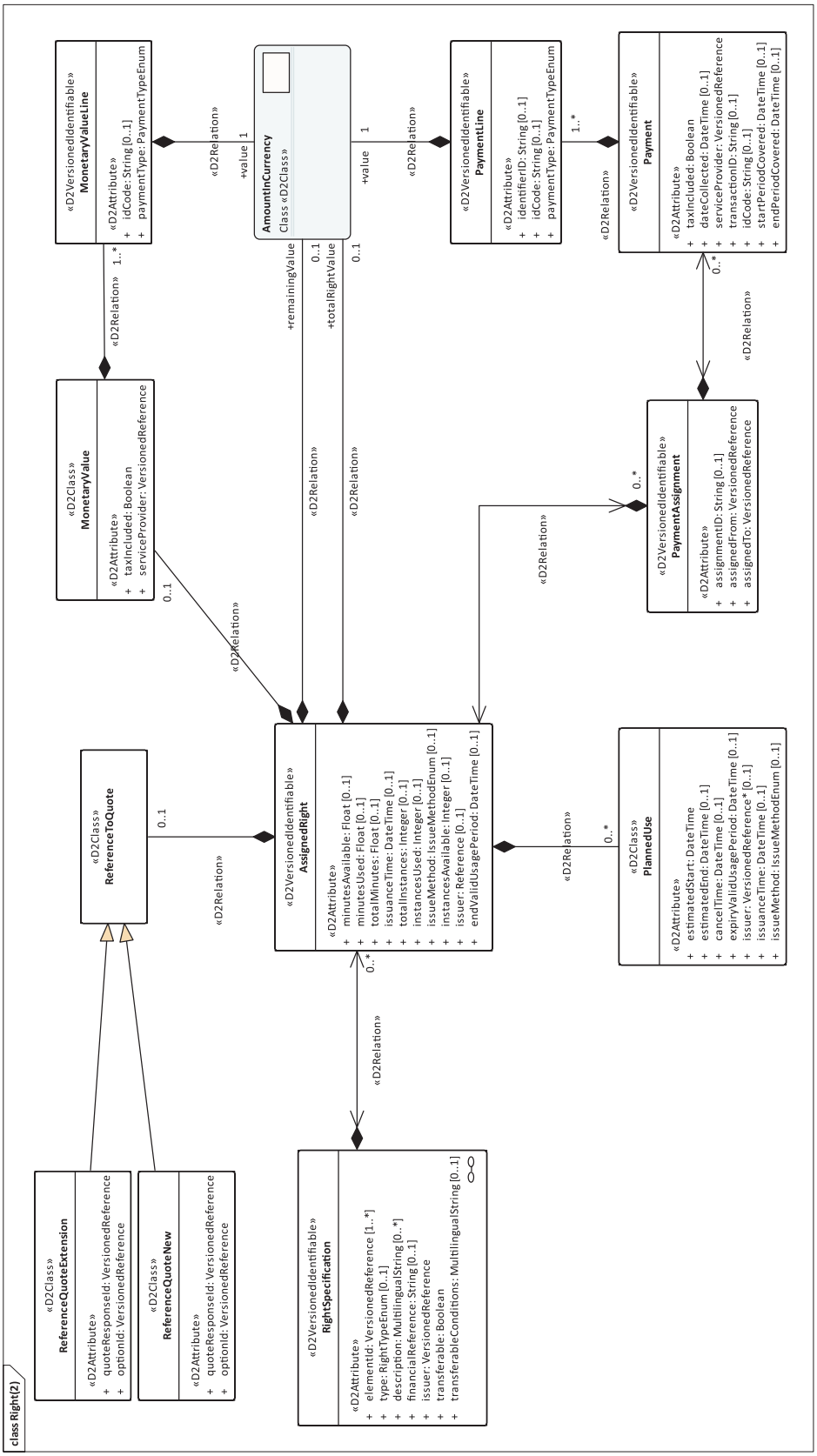


Figure 26 — Right domain model – submodel 2

© ISO 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

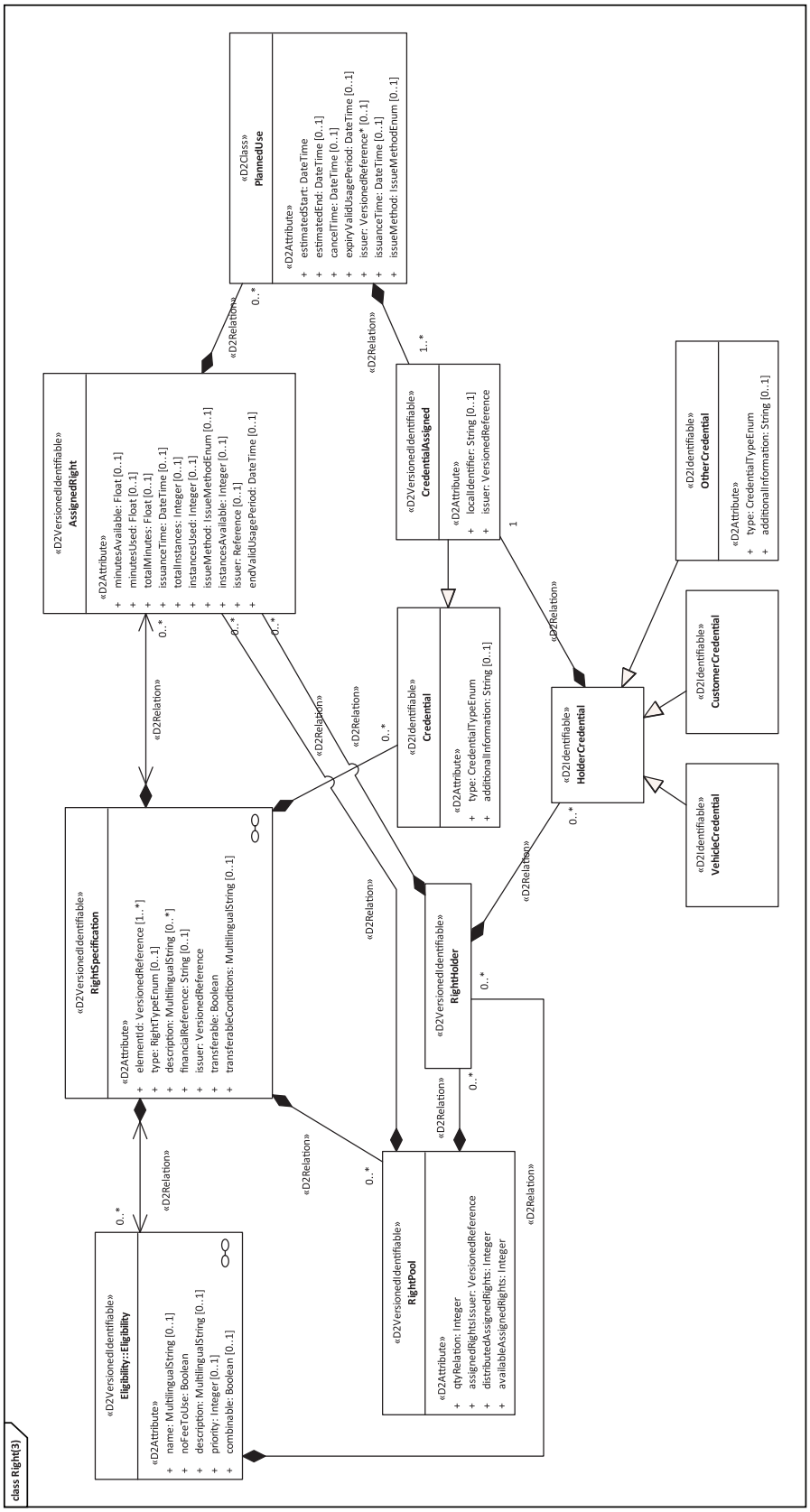


Figure 27 — Right domain model – submodel 3

Figure 28 provides an illustration of key data attributes to RightSpecification.

© ISO 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

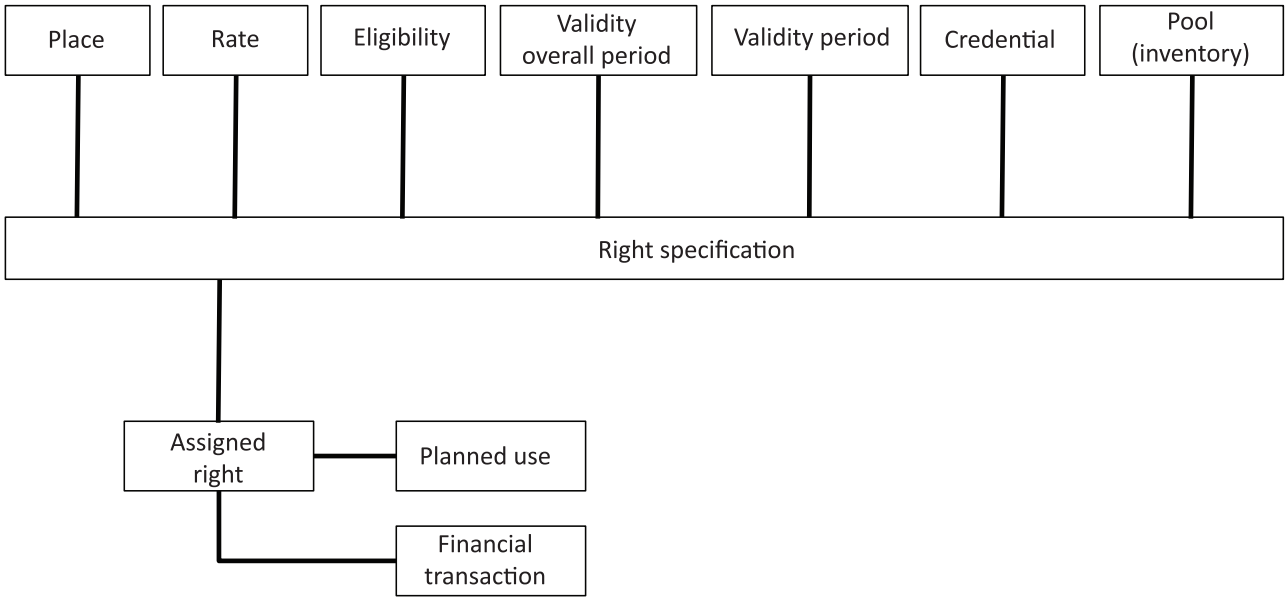


Figure 28 — Simplified `RightSpecification` and key data relationships

These are defined in the <Right> namespace of the data model.

The <Right> domain includes the data concepts that define how a facility/place owner or manager authorizes the use (e.g. park, delivery, pick up, etc.) of a specific `Place` to various users or vehicles via a `Credential` at a specific set of `RateTable`(s), if a `RateTable` applies.

Figure 29 shows the relation of `Rightspecification`, `RateTable` and `Eligibility`. To use `RightSpecification` and a related `RateTable`, a user or vehicle shall meet the eligibility criteria for both the `RightSpecification` and the `RateTable`. A `Rightspecification` defines the permission that is being granted to a vehicle or user to perform a specific activity (park, deliver, pick up, etc.) or to use a specific service in an `IdentifiedArea`. The `RateTable` defines how the `RightSpecification` is priced. `Eligibility` defines the qualification criteria to use a `RightSpecification` and `RateTable` combination.

© ISO 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

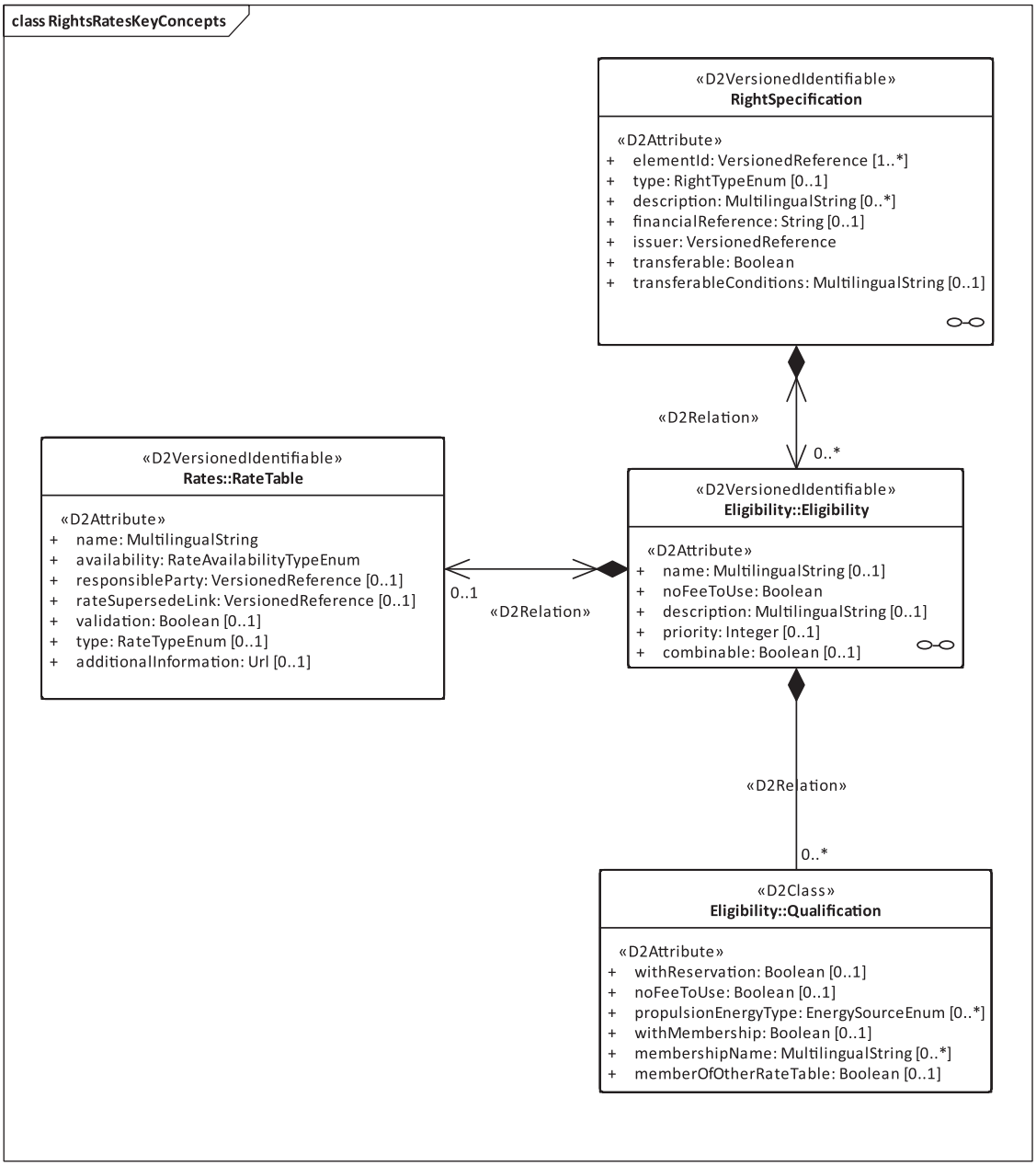


Figure 29 — Right — rates key data concepts

NOTE A `RightSpecification` can potentially be available to many, while specific `RateTables` associated to the `RightSpecification` can be limited based on specific qualifications. For example, a `RightSpecification` to park in a facility can have the following set of `RateTables`:

— Public RateTable: qualification is limited to passenger vehicles;

— Resident RateTable: qualification is limited to passenger vehicles and users that are members of the Resident group (i.e. live in the building).

The `RightSpecification` defines the operating parameters for parking or related mobility activities (delivery, pick-up/drop-off, electric vehicles only, etc.)

The `RightSpecification` is best described as the template of a right (permission to do a specific action) as defined by the place owner. A `RightSpecification` is granted to a specific `RightHolder` by an entity identified by the `AssignedRight.issuer` attribute. The `AssignedRight.issuer` can be the place owner,

© ISO 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

a reservation service, or other entities. When identified as the `AssignedRight.issuer`, the entity is authorized to sell or distribute the `RightSpecification` on behalf of the place owner.

When a `RightSpecification` is granted to a specific `RightHolder`, an `AssignedRight` is created. The `AssignedRight` includes the information from the `RightSpecification` as well as specific information related to the `RightHolder` (expiration of the `AssignedRight`, number of uses, etc). In some cases, an `AssignedRight` can include the ability to perform a specific `RightSpecification` multiple times (example, Rightholder has prepaid for five (5) parking events). When a specific, future use of the `AssignedRight` is initiated, a `PlannedUse` is generated.

RightSpecification:

— has a unique identification within a place;

— has a description of the permission (rightspecification) being granted;

— has an expiration: the date/time when the permission is no longer valid for any user as defined by the place owner;

— has a creator: the entity, typically the place owner that defines and authorizes the permission;

— has authorized credential types to identify methods of proof of holding an AssignedRight.

Data associations to a `RightSpecification` or `AssignedRight`:

1) Place: `Place` defines where a `RightSpecification` is valid. A `RightSpecification` can be associated to multiple `Places`. When multiple `Places` are associated to a `RightSpecification`, each place owner needs to authorize the `RightSpecification` and reference the proper `RightSpecification` IDs when sharing data.

2) Eligibility: This defines the type of vehicle, customer or other qualification criteria that is able to access the `RightSpecification`. `Eligibility` may be associated to being a member of a group (office employee, resident, etc.), a vehicle type (electric car, truck, passenger car, etc.), or use of a `RightSpecification` in a previous `Session`. Eligibility, in addition to defining the qualifications to access a `RightSpecification`, defines the `Qualifications` requirements to access specific `RateTables`. Within the `Eligibility` class, the ability to combine instances of `RateTable` within a `RightSpecification` is defined via a Yes/No (Boolean) definition. Either all instances of `RateTable` associated to a `RightSpecification` with a common `Eligibility` can be combined or they cannot. If `Eligibility` has the combinable flagged YES, then instances of `RateTable` can be combined in a `Session` or `Segment`.

3) Rate Table: `RateTable` is a specific rate structure that defines how a `RightSpecification` is priced. A `RightSpecification` can have multiple instances of `RateTable` associated to it as long as the instances of `RateTable` all apply to the same `Eligibility` requirements defined at the `RightSpecification`. Actual pricing in a `RateTable` may vary based on time of day.

4) Right Pool: shares the number of instances of `AssignedRight` that are available for use, are currently in use, or intended for use within a specific `RightSpecification` in specific date/time periods by a specific `AssignedRight.issuer`. A specific calendar (i.e. March 23, March 25, April 7) can be defined to represent the number of instances of `AssignedRight` available for distribution via the `Validity` class attributes `Rightpool.activeStart` and `Rightpool.activeEnd` or a recurring time period can be defined (Monday-Friday, Saturday-Sunday) via `RelativeValidity`.

5) The `Validity` class associated to the `RightSpecification` table defines two important time periods related to a `RightSpecification`.

   a) `overallPeriod.start` and `overallPeriod.end` identifies when the `RightSpecification` and associated attributes are valid for usage. This enables an entity to schedule the release of new `RightSpecification(s)` and communicate when current RightSpecifications will expire. If the `RightSpecification.overallPeriod.end` is not set, the RightSpecification is considered to be valid until it is replaced.

© ISO 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

b) `period.startOfPeriod` and `period.endOfPeriod` define when the `RightSpecification` are available during a specific period to be applied to a Session. In association with `RightPool`, these attributes define the specific number of instances of `RightSpecification` that are available for use, in use, or intended for use in specific date/time periods. A `RightSpecification` for monthly parking for a business employee can have `RightSpecification.Period.startOfPeriod` and `RightSpecification. endOfPeriod` valid Monday through Friday from 06:00 to 21:00, while a resident can have a `RightSpecification` with `RightSpecification.Period.startOfPeriod` and `RightSpecification.Period.endOfPeriod` of Sunday through Saturday, 24 hours a day, for example.

6) RightHolder: this is a specific entity [e.g. individual, corporation, and vehicle] that is issued a `RightSpecification` by a place owner or `AssignedRight.issuer` via an `AssignedRight`. The `RightSpecification`, along with additional data related to the `RightHolder`, is contained in the `AssignedRight` class. Associated to a `RightHolder` are the individuals and/or vehicles able to access the `AssignedRight`.

   a) A `RightHolder` may have multiple vehicles associated to one or more credentials.

      EXAMPLE 1 A person that has access to more than one vehicle and uses them interchangeably.

   b) A `RightHolder` may have multiple users associated to one or more credentials.

      EXAMPLE 2 A company that provides `RightSpecification` to its employees under one contract, or that issues validations to customers to discount their parking.

      EXAMPLE 3 A family share a defined number of credentials.

7) AssignedRight: when a place owner or an `AssignedRight.issuer` (authorized issuer) grants a `RightSpecification` to a specific person or entity, an `AssignedRight` is created. The `AssignedRight` includes the key data from the `RightSpecification` as well as specific data related to the `RightHolder's` use of the `RightSpecification` which includes:

   a) approved credential;

   b) expiration date and time of `AssignedRight`;

   c) rate(s);

   d) valid number of uses, minutes or some other value describing the quantity of use;

   e) additional data on the entity that issued the `RightSpecification` to the `RightHolder`. The issuer can be the place owner, place manager, or an `AssignedRight.issuer` (third party entity authorized by the place owner);

   f) additional data about when the `AssignedRight` was issued to the `RightHolder`;

   g) `AssignedRight` has one to multiple `PlannedUse`(s);

   h) `AssignedRight` has financial data defined in two data concepts: `MonetaryValue` and `Payment`;

      i) `MonetaryValue`: the calculated total value of the `AssignedRight` based on duration, number of uses, and `RateTable` value. This is the expected total value of the `AssignedRight`.

      ii) `Payment`: the completed actual payments by an entity for the `AssignedRight`. A payment can be the transfer of funds between two entities (i.e. a payment) or a validation. A validation is a transaction where one entity has the ability to reduce or agrees to fund a portion of an `AssignedRight` for a specific entity.

8) PlannedUse: When an `AssignedRight` is "activated" for a future use, a `PlannedUse` class instance is created. `PlannedUse` defines the proposed time to use the `AssignedRight` in the future and identifies the credential of the `AssignedRight`.

© ISO 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

The specification defines data models to share financial transaction data as defined in the `MonetaryValue` and `Payments` classes associated to a specific `AssignedRight` as described above. The specification also supports sharing financial transaction data between entities for funds disbursement related to the aggregation of multiple `AssignedRight(s)` transactions.

Monetary Value

`MonetaryValue` allows an entity to share the expected value of a transaction (a specific `AssignedRight`) in addition to relevant attributes to further define the transaction. The additional attributes include:

— `AssignedRight.issuanceDate` – the date and time the `AssignedRight` was issued and the monetary value was defined;

— `MonetaryValue.taxIncluded` – Boolean attribute that denotes if tax is embedded in the fee presented to the `AssignedRightholder` or if the taxes are explicitly identified as a separate charge;

— `serviceProvider` – the serviceprovider presenting and collecting the `MonetaryValue`;

— `MonetaryValueLine` – a data class that enables transaction details to be accumulated based on the type of payment (i.e. base fee, taxes, surcharges, etc.):

  — base fee - expected value to be received by place owner or operator (i.e. amount of calculated `RateTable` based on `Session` times),

  — tax – expected value of various taxes as defined in rate table,

  — surcharges – expected value of surcharges as defined in rate table,

  — discount – expected value of discounts or validations to be applied;

— `currency` – the currency used to define the monetary value.

Payment

`Payment` allows an entity to share the actual payment values received for a transaction. Attributes related to `Payment` include:

— `dateCollected` – the date the payment was collected;

— `serviceProvider` – the service provider collecting the monetary value;

— `transactionID` – a reference number related to the transaction;

— `idCode` – a user-defined field to identify additional details to the transaction such as Customer Account number or G/L account number;

— `periodCovered` – note field to describe the period a payment covers (e.g. USD 100 for monthly parking in August);

— `PaymentLine` – a data class that enables transaction details to be accumulated based on the type of payment (i.e. base fee, taxes, surcharges, etc.):

  — base fee – value received by place owner or operator (i.e. amount of calculated `RateTable` based on `Session` times),

  — tax – value of various taxes as defined in rate table,

  — surcharges – value of surcharges as defined in rate table,

  — discount – value of discounts or validations to be applied;

— `currency` – the currency used to define the monetary value.

© ISO 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

Using `PaymentAssignment` and `Payment`, an entity can document financial transactions related to the disbursement of aggregated transaction amounts to various entities and provide reference to the individual `AssignedRight` transactions included in the funds disbursement to facilitate reconciliation activities.

`PaymentAssignment` includes the following attributes:

— `assignmentID`: identification of the specific transaction for the aggregated funds disbursement;

— `assignedFrom`: the entity sending the aggregated funds disbursement;

— `assignedTo`: the entity receiving the aggregrated funds disbursement;

— `AssignedRights`: the matrix of AssignedRight IDs that constitute the aggregated funds disbursement

— `Payment`: as defined above.

An example of the use of `PaymentAssignment` is when a PLACE owner disburses funds to a taxing entity for tax payments and service providers for their services fees. Each disbursement represents the aggregate amount for multiple AssignedRights. An example is provided below:

PLACE collects the following payments on the listed AssignedRights:

	AssignedRight #1	AssignedRight #2	AssignedRight #3	AssignedRight #4	
Base fee	USD 8,36	USD 12,23	USD 10,05	USD 3,00	USD 33,64
Taxes/VAT	USD 1,64	USD 2,77	USD 1,95	USD 1,00	USD 7,36
Surcharge (payable to Reservation.com)	USD 1,00	USD 1,50	USD 1,50		USD 4,00
Surcharge (payable to phone.com)				USD 0,35	USD 0,35
Total	USD 11,00	USD 16,50	USD 13,50	USD 4,35	USD 45,35

The Placeowner makes payments to the taxing authority and its two service providers.

PaymentAssignment

Assignment ID: 1234

Assignedfrom: PLACEowner

Assignedto: TaxingAuthority

AssignedRights: AssignedRight#1, AssignedRight#2, AssignedRight#3, AssignedRight#4

Payment

Taxincluded: N

Datecollected: October 5, 2021

PaymentLine:

PaymentType: Tax

AmountinCurrency

CurrencyValue: 7,36

Currencycode: USD

© ISO 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

Assignment ID: 3456

Assignedfrom: PLACEowner

Assignedto: reservation.com

AssignedRights: AssignedRight#1, AssignedRight#2, AssignedRight#3, AssignedRight#4

Payment

Taxincluded: N

Datecollected: October 5, 2021

PaymentLine:

PaymentType: Tax

AmountinCurrency

Currencyvalue: 4,00

Currencycode: USD

Assignment ID: 3458

Assignedfrom: PLACEowner

Assignedto: phone.com

AssignedRights: AssignedRight#1, AssignedRight#2, AssignedRight#3, AssignedRight#4

Payment

Taxincluded: N

Datecollected: October 5, 2021

PaymentLine:

PaymentType: Tax

AmountinCurrency

Currencyvalue: 0,35

Currencycode: USD

Financial data will be expanded in future editions of this document.

6.5.1 Enumerations (for right)

The following enumerations are defined in the <Right> namespace, see Figure 30.

© ISO 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

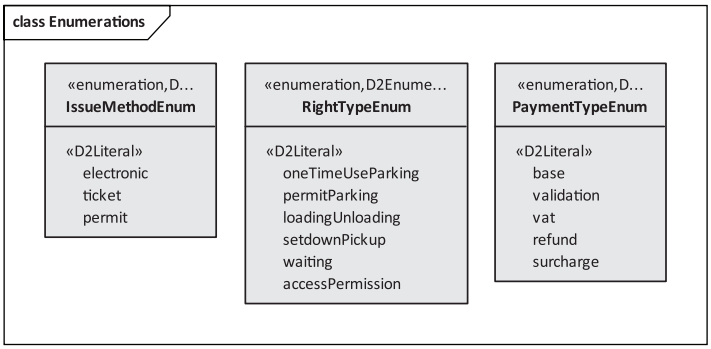


Figure 30 — Enumerations in the right domain

6.5.2 External codelists (for right)

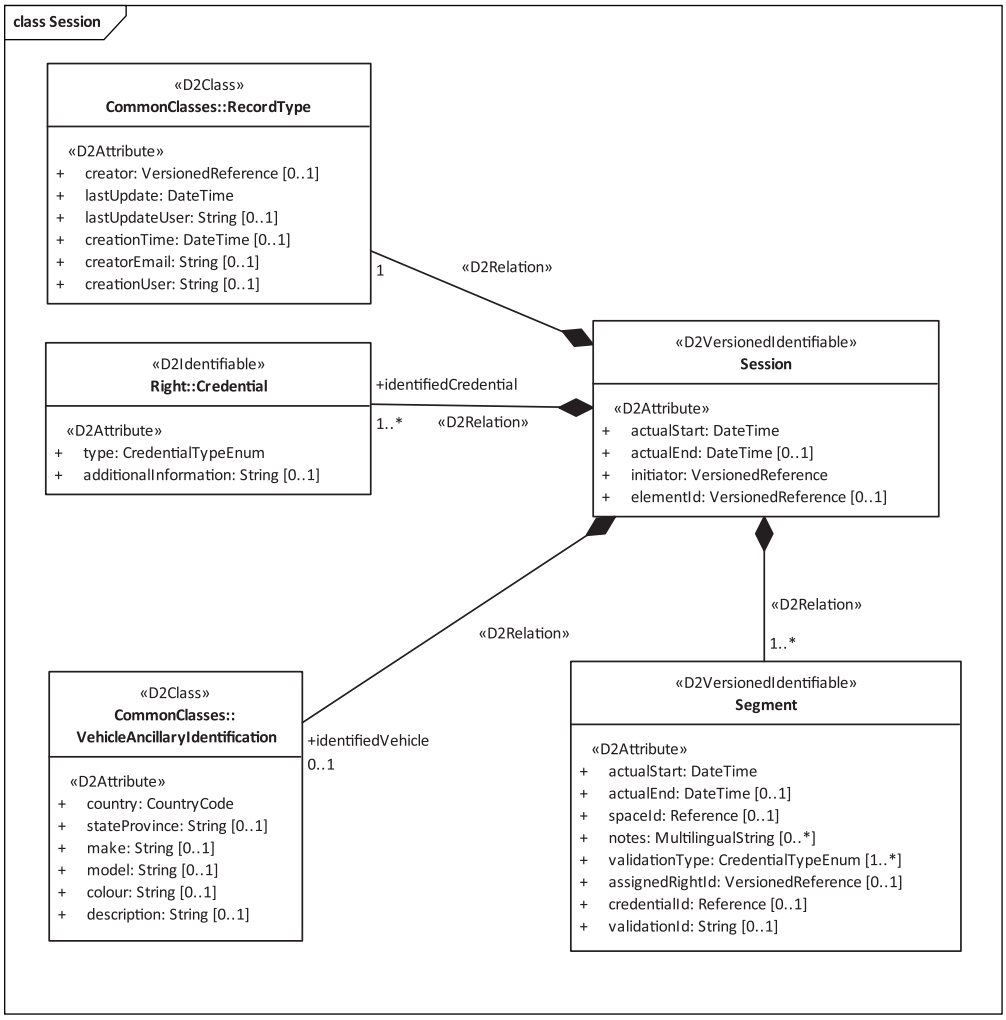
No specific external codelists are defined in <Right>. For general external codelists, see 6.9.5.

6.6 Session

6.6.1 General

The data concepts with the <Session> domain are presented in Figure 31.

© ISO 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023



NOTE A session occurs in one place.

Figure 31 — Session domain model

These data concepts are defined in the <Session> namespace of the data model.

`Session` includes the data concepts that document the actual act of parking, or other use of an `AssignedRight` such as delivery, pickup, etc. A `Session` captures the actual use of an `AssignedRight` and includes start time, end time, credential and other relevant data related to an operational transaction. A `Session` is not used for future activities.

A `Session` is broken into one or multiple `Segments`.

`Segments` enable a single `Session` to capture and report changes in `RateTable` and `AssignedRight` during a `Session`.

A `Segment` can only have one `AssignedRight` and `RateTable` associated to it.

`Session`:

— has a unique identification within a `Place`;

— has a start time and end time;

— has at least one `Segment`;

— is associated to a `Place`.

© ISO 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

`Segment`:

— is associated to a `Session`;

— is associated to an `AssignedRight`;

— is associated to a specific `RateTable` via the `AssignedRight`;

— has a start and end time;

— has a `Credential` via the `AssignedRight`;

— has version control on the `AssignedRight` (i.e. a `Segment` starts with one `AssignedRight` and then is updated to a new `AssignedRight` at the end of the `Segment`, replacing the original);

— has a `Place` reference or Space ID;

— can be associated to one or multiple `Observations`.

6.6.2 Enumerations (for session)

No specific enumerations defined in <Session>.

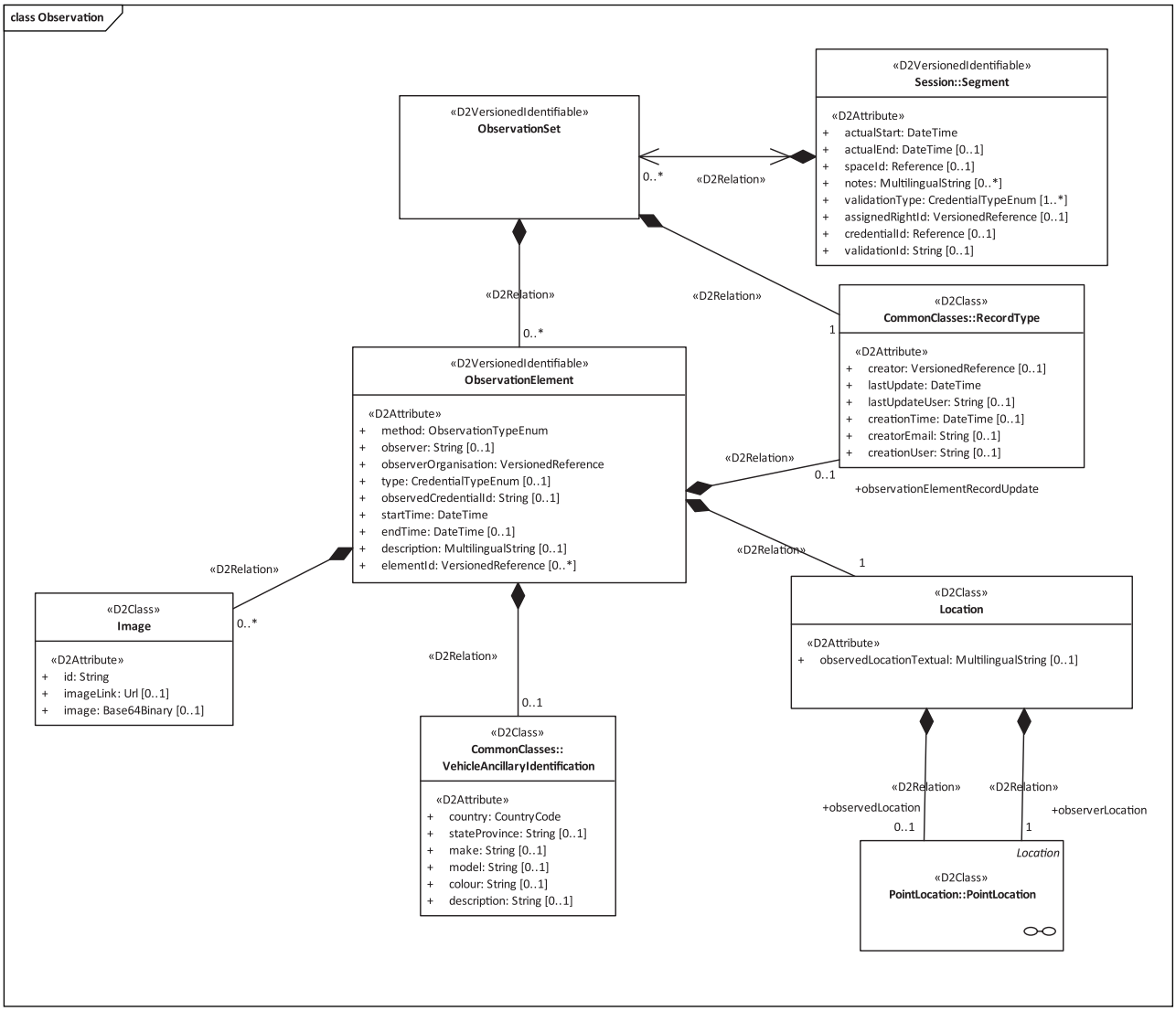
6.6.3 External codelists (for session)

No specific external codelists are defined in <Session>. For general external codelists, see 6.9.5.

6.7 Observation

The data concepts with the <Observation> domain are presented in Figure 32.

© ISO 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023



NOTE Image can be either image or video, linked via URL or directly embedded BLOB.

Figure 32 — Observation domain model

These data concepts are defined in the <Observation> namespace of the data model.

Observation data includes the data concepts that document the `Observation` of an entity, individual, vehicle, etc. in a `Place` performing an action. `Observations` can be visual or electronic (e.g. ANPR read, RFID read, etc).

Observation:

— has a date/time when the observation is made;

— has a location where the observation is made;

— has an association to a credential observation.

Credential observation:

— has a reference identification of the `Observation`;

— has a method of observation (ANPR, visual, ticket, RFID, etc);

© ISO 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

— has an observer name (who made the observation);

— has a credential type observed (licence plate, tag, hang tag);

— has the credential identification observed.

6.7.1 Enumerations (for observation)

The following enumerations are defined in the <Observation> namespace, see Figure 33.

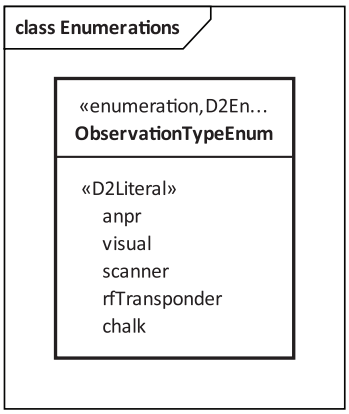


Figure 33 — Enumerations in the observation domain

6.7.2 External codelists (for observation)

No specific external codelists are defined in <Observation>. For general external codelists, see 6.9.5.

6.8 Quote

6.8.1 Introduction to quote

This subclause describes the concepts used to define a Quote. There are two data structures to support the use of Quotes:

1) A `QuoteRequest` – a request for quote prices on `RightSpecification(s)` where `Eligibility` requirements are met for a specific time range. In some cases, a specific `Place` can be defined.

2) A `QuoteResponse` – a response to a `QuoteRequest` which provides the option(s) available to a `QuoteRequest` that meet `Eligibility` requirements. Each option in a `QuoteResponse` is versionable.

   NOTE "`QuoteRequest`" and "`QuoteResponse`" are abstract super-types (i.e. they define classes of requests and responses, and are not directly instantiated themselves).

A `Quote` is a request for a firm price and confirmation of availability of a specific `RightSpecification`, at a specific time, by a specific user. A specific `Place` may also be specified in the request and is required in the response. A `QuoteRequest` should initiate a `QuoteResponse` being received which includes relevant option(s) that the meet the criteria of the `QuoteRequest` and typically includes an expiration date/time on the `QuoteResponse` for the recipient to act (i.e. purchase) to obtain/or purchase an `AssignedRight`.

A `QuoteResponse` confirms the requester meets `Eligibility` requirements and provides the requester with a price for the use of a requested `RightSpecification`. `QuoteRequests` and `QuoteResponses` are used between systems to allows users to evaluate a purchase of a `RightSpecification`. `QuoteResponses` are not authorizations to use a `RightSpecification` and do not create an `AssignedRight` or assign `Credentials`. A `QuoteResponse` needs to be converted into an `AssignedRight` by the issuer to provide the acquiring user the right or permission to perform a specific action, as defined in the `RightSpecification` in a specific `Place`.

© ISO 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

A `QuoteRequest` is not the proper means to request general `RateTable` or `RightSpecification` details about a `Place`. Other data structures exist to support messaging to retrieve these details. The purpose of a `QuoteRequest` is to enable a user to inquire about the purchase or acquisition of a specific `RightSpecification` at a specific time for a specific `Place`.

These concepts are defined in the <Quote> namespace of the data model.

6.8.2 QuoteRightRequest — Request for a new transaction

A `QuoteRightRequest` class, see Figure 34, (which is versioned identifiable) references one or several `RightSpecification` (which is versioned identifiable). A `QuoteRightRequest` can include the relevant `RightSpecification`(s) for the desired `Place` to be defined. This implies that the `QuoteRightRequest` initiator has a current list of available `RightSpecification` for the `Place`.

NOTE A `QuoteRightRequest` can include requests for multiple `RightSpecification` in a single request.

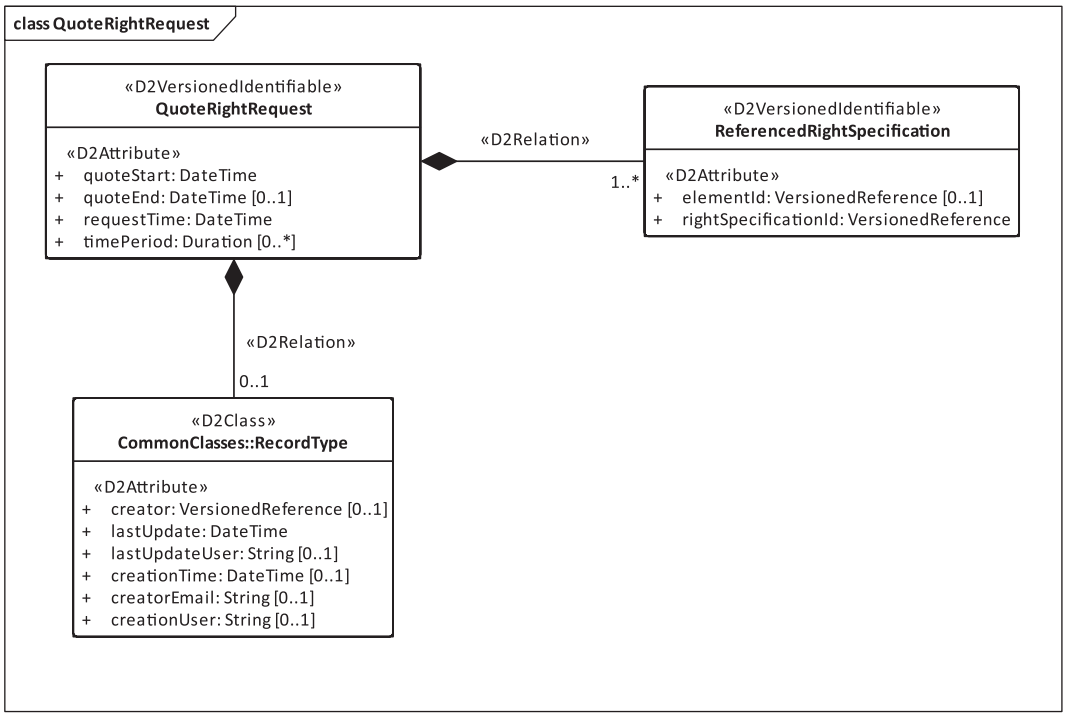


Figure 34 — QuoteRightRequest class diagram

A `QuoteRightRequest` includes the following data concepts in its data structure:

1) Desired `RightSpecification`(s) to check availability and confirm `Eligibility`.
   a) A `QuoteRightRequest` can include several `RightSpecification`(s) .
   b) At a minimum, one `RightSpecification` is required.
2) Desired `Place`(s).
   a) A `QuoteRightRequest` can include several `Place`(s). `Place` is optional.
3) Targeted period for intended use of a relevant `RightSpecification` as defined by options:
   a) A `quoteStart`[1] and a `quoteEnd`. [1]
   b) A `quoteStart` with one or several `timePeriod` durations – which supports the idea of retrieving quotations for a range of periods with a defined start time (e.g. from 09:30 for a period of 15 minutes and also for a period of 30 minutes, 1 hour, 3 hours, etc).

© ISO 2023 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023

c) A `QuoteRightRequest` with `quoteStart` that is not defined then defaults to the time of the `QuoteRightRequest` received.

d) Constraint: either `quoteEnd` or `timePeriod` shall be used.

4) Submission time of the `QuoteRightRequest`.

a) `requestTime` is the date and time in UTC, at which the sending party sends `QuoteRightRequest` submission.

5) Known `Eligibility` requirements.

a) Certain `RightSpecification` and `RateTable` require defined `Eligibility` requirements be met to be accessible by a requestor. To access `Eligibility` controlled `RightSpecifications` and `RateTables`, the `Eligibility` requirements need to be included in the `QuoteRightRequest`.

The `QuoteRightRequest` can be optionally qualified with attribution using the sub-model under `Eligibility` (although this can be unnecessary as the requesting part can collect this information prior by seeking available `RightSpecification`(s)).

6.8.3 QuoteRightResponse — Response to a QuoteRightRequest for a new transaction

A `QuoteRightResponse` class, see Figure 35, (which is versioned identifiable) is the documented response to a `QuoteRightRequest` and includes a time-period covered for the intended new future `Session`. This class documents the `Option`(s) presented to a requester of a `QuoteRightRequest`. An `Option` lists the specific `RightSpecification`(s) and `RateTable`(s) that meet the requirements defined in the `QuoteRightRequest` and for which the requester meets `Eligibility` requirements. When a `QuoteRightResponse` includes more than one possibility, the possibilities are defined as `Option`(s) and each `Option` is versioned identifiable.

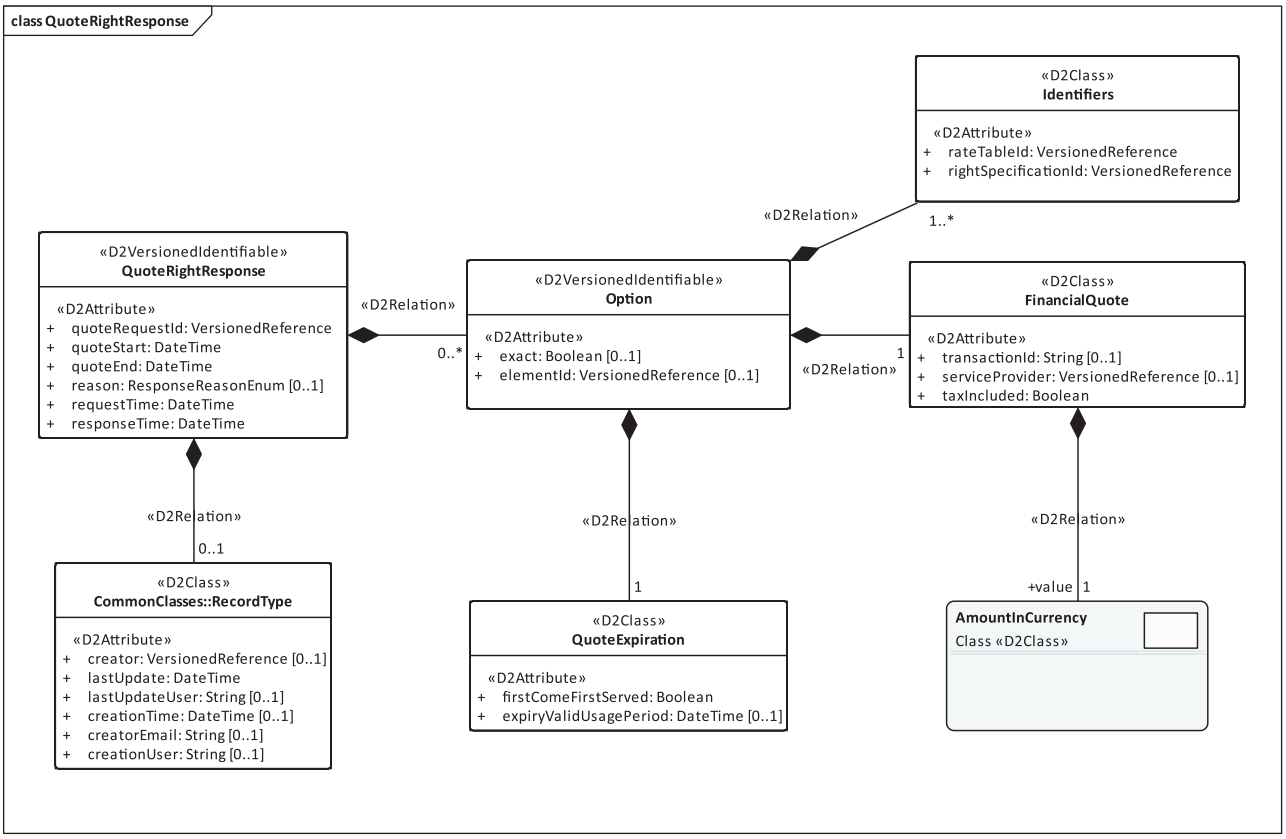


Figure 35 — QuoteRightResponse class diagram

© ISO 2023 – All rights reserved
Documents
Document Sharing Hub
Full document details: ISO TS 05206-1-2023