

Information technology — City data model— Part 3: Service level concepts -Transportation planning

WD stage

Warning for WDs and CDs

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

© ISO 20XX

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

Draft	Date	Changes
2	3 Dec 2021	<ul style="list-style-type: none"> Removed reference to Observations and Loop Detectors in last paragraph of section 5.1
	4 Jan 2022	<ul style="list-style-type: none"> Addressed TC204 comments Added property to Vehicle for engine type, distinguishing combustion and electric
	29 Oct 2023	<ul style="list-style-type: none"> Updated content to reflect new core model as agreed upon between TC204 and WG11
3	20 May 2024	<ul style="list-style-type: none"> Updated based on JWG1 input
4	12 Feb 2025	<ul style="list-style-type: none"> Update to add alerts

Contents

Foreword	vi
Introduction	vii
1 Scope	1
2 Normative references	1
3 Terms and definitions.....	1
4 Symbols and abbreviated terms	1
5 Conventions	2
5.1 Documentation format	2
5.2 Unique identifiers	3
6 Service-level ontologies for transportation.....	3
6.1 General	3
6.2 Transportation network pattern	4
6.2.1 Overview	4
6.2.2 Key classes and properties	4
6.2.3 Formalization	10
6.3 Road network pattern	11
6.3.1 Overview	11
6.3.2 Key classes and properties	11
6.3.3 Formalization	15
6.4 Rail network pattern.....	16
6.4.1 Overview	16
6.4.2 Key classes and properties	16
6.4.3 Formalization	18
6.5 Micromobility network pattern	19
6.5.1 Overview	19
6.5.2 Key classes and properties	19
6.5.3 Formalization	22
6.6 Pedestrian network pattern.....	23
6.6.1 Overview	23
6.6.2 Key classes and properties	23
6.6.3 Formalization	26
6.7 Public transport network pattern	27
6.7.1 Overview	27
6.7.2 Key classes and properties	28
6.7.3 Formalization	28
6.8 Travel corridor pattern.....	29
6.8.1 Overview	29
6.8.2 Key classes and properties	29
6.8.3 Formalization	30
6.9 Transport alert pattern.....	31
6.9.1 Overview	31
6.9.2 Key classes and properties	32
6.9.3 Formalization	32
Annex A (informative) Understanding UML diagrams	34
A.1 General	34
A.2 Classes	34

A.3 Subclasses.....	34
A.3.1 General	34
A.3.2 Subclass Hierarchy.....	35
A.3.3 Multiple inheritance.....	36
A.4 Properties.....	36
A.4.1 Object properties.....	37
A.4.2 Data properties	37
A.4.3 Multiplicity.....	37
A.4.4 Inverse properties.....	38
A.5 Namespaces.....	38
A.6 Diagrams versus model	39
Annex B (informative) Example use cases.....	40
Annex C (informative) Relationship to existing standards	41
C.1 TC204 relevant standards	41
C.2 OpenDrive.org.....	41
C.3 CityGML.....	41
C.3.1 Scope	41
C.3.2 Relevance.....	41
C.3.3 Data Mappings	41
C.4 INSPIRE.....	41
C.4.1 Scope	41
C.4.2 Relevance.....	41
C.4.3 Data Mappings	41
C.5 37166: Smart community infrastructures—Urban data integration framework for smart city planning (SCP)	41
C.5.1 Scope	41
C.5.2 Relevance.....	42
Annex D (informative) Location of pattern implementations.....	43
Bibliography	44

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

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 document 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 or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and <https://patents.iec.ch>. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

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. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by the Joint Working Group between Technical Committee 204 and the Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

A list of all parts in the ISO/IEC 5087 series can be found on the ISO and IEC websites.

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 and www.iec.ch/national-committees.

Introduction

The intended audience for this document includes municipal information systems departments, municipal software designers and developers, and organizations that design and develop software for municipalities.

Cities today face a challenge of how to integrate data from multiple, unrelated sources where the semantics of the data are imprecise, ambiguous and overlapping. This is especially true in a world where more and more data of interest is being openly published from various organizations. A morass of data is increasingly becoming available to support city planning and operations activities. In order to be used effectively, the data must be unambiguously understood so that it can be correctly combined, avoiding data silos. Early successes in data “mash-ups” relied upon an independence assumption, where unrelated data sources were linked based solely on geospatial location, or a unique identifier for a person or organization. More sophisticated analytics projects that require the combination of datasets with overlapping semantics entail a significantly greater effort to transform data into something useable. It has become increasingly clear that integrating separate datasets for this sort of analysis requires an attention to the semantics of the underlying attributes and their values.

A common data model enables city software applications to share information, plan, coordinate, and execute city tasks, and support decision making within and across city services, by providing a precise, unambiguous representation of information and knowledge commonly shared across city services. This requires a clear understanding of the terms used in defining the data, as well as how they relate to one another. This requirement goes beyond syntactic integration (e.g. common data types and protocols), it requires semantic integration: a consistent, shared understanding of the meaning of information.

To motivate the need for a standard city data model, consider the evolution of cities. Cities deliver physical and social services that traditionally have operated as silos. In other words, data is generated, processed and used by systems that are nominally under the control of one or more parties with similar interest, and thus handled in such a way as to facilitate the business processes of those parties. Parties with other interests and business needs develop their own data, often with different requirements and data sharing among the different groups is limited.

If during the process of becoming smarter, transportation, social services, utilities, etc. were to develop their own data models, then we would have smarter silos. To create truly smart cities data must be shared across these silos which can only be accomplished through the use of a common data model. For example, “Household” is a category of data that is commonly used by city services. Members of Households are the source of transportation, housing, education, and recreation demand. It represents who occupies a home, age, occupations, where they work, abilities, etc. Though each city service can gather and/or use different aspects of a Household, much of the data needs to be shared with each other.

Supporting this interoperability among city datasets is particularly challenging due to the diversity of the domain and the heterogeneity of its data sources. The purpose of this document is to support the precise and unambiguous specification of city data using the technology of Ontologies (Gruber, 1993) (Gruninger & Fox, 1995) as implemented in the Semantic Web. (Berners-Lee, Hendler, & Lassila, 2001) By doing so it will:

- enable the computer representation of precise definitions thereby reducing the ambiguity of interpretation,
- remove the independence assumption, thereby allowing the world of Big Data, open-source software, mobile apps, etc., to be applied for more sophisticated analysis,
- achieve semantic interoperability, namely the ability to access, understand, merge and use data available from datasets spread across the semantic web,
- enable the publishing of city data using Semantic Web and ontology standards, and

- enable the automated detection of city data inconsistency, and the root causes of variations, and
- support semantic interoperability of city data in SMART transformation and smart governance.

With a clear semantics for the terminology, it is possible to perform consistency analysis, and thereby validate the correct use of the standard.

Figure 0-1 identifies the three levels of the ISO/IEC 5087 series. The lowest level, defined in ISO/IEC 5087-1, provides the classes, properties and logical, computational definitions for representing the concepts that are foundational to representing any data. The middle level, defined in ISO/IEC 5087-2, provides the classes, properties and logical, computational definitions for representing urban specific concepts common to all city services but not specific to any service. The top level provides the classes, properties, and logical, computational definitions for representing service specific concepts that are used by other services across the city. For example, ISO/IEC 5087-3 (this document) defines the Transportation concepts. In the future, additional parts will be added to the ISO/IEC 5087 series covering services such as Education, Water, Sanitation, Energy, etc.

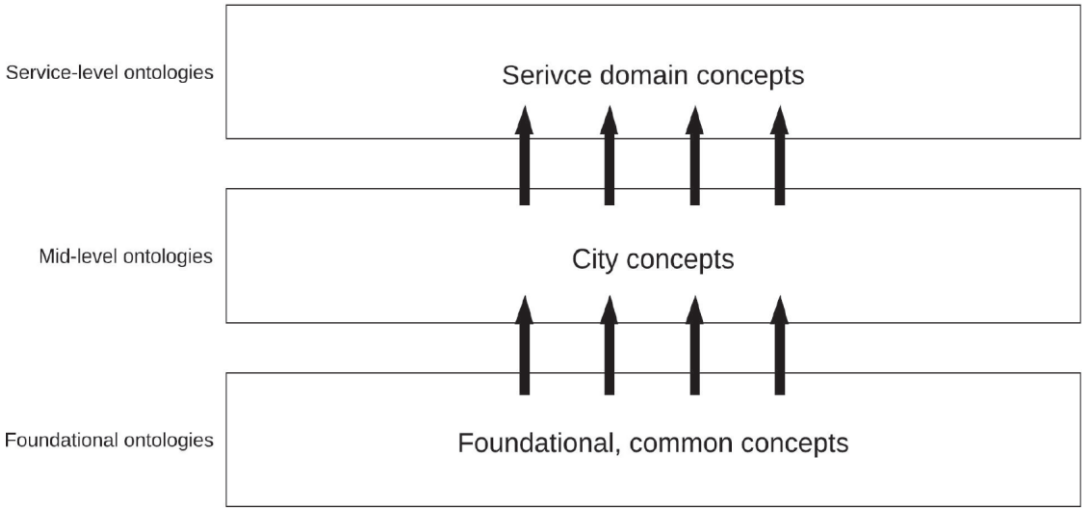


Figure 0-1— Stratification of city data model

Figure 0-2 depicts example concepts for the three levels.



Figure 0-2 — Example concepts for each level

There are other existing standards that overlap conceptually with some of the terms presented in this document. The relationship between ISO/IEC 5087-3 and existing standards that address similar or related concepts is identified in Annex C.

Information technology — City data model— Part 3: Service level concepts -Transportation planning

1 Scope

This document defines an ontology for service-level concepts defined for transportation using terms specified in ISO/IEC 5087-1 and ISO/IEC-5087-2. City-level concepts defined in ISO/IEC-5087-2 and service-level concepts defined in ISO/IEC 5087-3 are distinguished by city-level concepts' data being read and updated by multiple city services and stakeholders, whereas service-level concepts should be read but not necessarily written by multiple city services and stakeholders.

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 documents (including any amendments) applies.

SEMANTIC SENSOR NETWORK ONTOLOGY. W3C Recommendation 19 October 2017, <https://www.w3.org/TR/vocab-ssn/>

ISO/IEC 5087-1:2023, *Information technology — City Data Model — Part 1: Foundation Level Concepts*

ISO/IEC 5087-2:2024, *Information technology — City Data Model — Part 2: City Level Concepts*

ISO/IEC/IEEE 24765:2017, *Systems and software engineering — Vocabulary*

ISO/TS 14812:2022, *Intelligent transport systems — Vocabulary*

Manchester Syntax, W3C Editor's Draft 28 November 2008, <https://www.w3.org/2007/OWL/draft/ED-owl2-manchester-syntax-20081128/>

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC/IEEE 24765, ISO/TS 14812, ISO/IEC 5087-1 and ISO/IEC 5087-2 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/>

4 Symbols and abbreviated terms

DL	description logic
OWL	ontology web language
RDF	resource description framework
RDFS	resource description framework schema
IRI	international resource identifier

The following namespace prefixes are used in this document:

- code: <https://standards.iso.org/iso-iec/5087/-2/ed-1/en/ontology/Code/>
- geo: <http://www.opengis.net/ont/geosparql#>
- owl: <https://www.w3.org/2002/07/owl#>
- partwhole: <https://standards.iso.org/iso-iec/5087/-1/ed-1/en/ontology/Mereology/>
- rdf: <https://www.w3.org/1999/02/22-rdf-syntax-ns#>
- rdfs: <https://www.w3.org/2000/01/rdf-schema#>
- loc: <https://standards.iso.org/iso-iec/5087/-1/ed-1/en/ontology/SpatialLoc/>
- service: <https://standards.iso.org/iso-iec/5087/-2/ed-1/en/ontology/CityService/>
- transnet: <https://standards.iso.org/iso-iec/5087/-3/ed-1/en/ontology/TransportationNetwork/>
- transinfras: <https://standards.iso.org/iso-iec/5087/-2/ed-1/en/ontology/TransportationInfrastructure/>
- time: <https://www.w3.org/2006/time#>
- xsd: <https://www.w3.org/2001/XMLSchema#>

5 Conventions

5.1 Documentation format

The formalization of the classes in this document is specified using the following table format, which is a simplification of description logic (DL) where the first column identifies the class name, the second column its properties, third column indicates the multiplicity of the property (i.e., how many instances of the property can occur for each instance of the class), and the fourth column each property's range restriction. It shall be read as: "The <Class> is characterized by <Mult> instances of <property> of the specified <restriction>". For example, Table 1 specifies that Agent is a (1) subclass of the intersection of (Person or Organization) and can be a org:memberOf of zero or more Organizations.

•

Table 1—Example formalization of the Agent class

Class	Property	Mult.	Value Restriction
Agent	rdfs:subClassOf		Person or org_s:Organization
	org_s:memberOf	0..*	Organization
	individual		{joe, frank}

CamelCase is used for specifying classes, properties and instances. For example, "legalName" instead of "legal_name". The first letter of a class name is capitalized. The first letter of a property and instance name are not capitalized.

An instance of a class shall satisfy the class's definition. The instance's properties and values shall satisfy the multiplicity and value restrictions of the class it is an instance of.

The formalization of properties is done similarly to the formalization of classes, using the following table format that allows for the identification of properties and their sub-properties, inverse properties, or other characteristics. It is to be read as: The <property> is <characteristic> of <value>, or simply the <property> is <characteristic> if no value is applicable. For example, in Table 2 hasPrivilege is a sub-property of the agentInvolvedIn property. Characteristics are specified using the Manchester syntax.

Table 2—Example property formalization

Property	Characteristic	Value (if applicable)
hasPrivilege	rdfs:subPropertyOf	agentInvolvedIn
	Irreflexive	

In the case of DL definitions of classes where the simplified table representation is insufficient, the DL specification will be supplied as an addition to the content in the table.

The patterns defined in this document have also been implemented in OWL and made available at <https://standards.iso.org/iso-iec/5087/-3/ed-1/en/>.

5.2 Unique identifiers

All classes, properties and instances of classes have a unique identifier that conforms to Linked Data/Semantic Web standards. The unique identifier is an IRI. When using this document in an application, a class is identified by the IRI for the pattern of which it is a member, followed by the class name. In the Agent example in Clause 5, the unique identifier for the Agent class, which is formally defined in ISO/IEC 5087-1 would be:

<https://standards.iso.org/iso-iec/5087/-1/ed-1/en/ontology/Agent/Agent>

Breaking the IRI down:

- “5087” identifies the series number
- “-1” identifies the part number
- “ed-1” indicates that the class is defined in edition 1 of the document
- “en” indicates that the class is defined in a pattern implemented in English
- The first “Agent” identifies the Agent Pattern
- The second “Agent” identifies the Agent class within the Agent Pattern

The IRI can be shortened using the prefix’s defined in Clause 4:

agent:Agent

where agent: is the prefix for the Agent Pattern.

Properties are identified in the same manner. The IRI’s of individuals created by an application of ISO/IEC 5087-2 would have IRI’s unique to the application.

6 Service-level ontologies for transportation

6.1 General

The service-level documents of the ISO/IEC 5087 series provide representations for concepts that are used by services that operate in the city context. The following concepts may be used (consumed) by multiple different city services and stakeholders, but typically only generated by the city service area

identified in the part name (i.e., in the case of this part, the data is typically only generated by transportation services).

The patterns defined in the service-level documents of the ISO/IEC 5087 series conform to the foundational and city-level concepts defined in ISO/IEC 5087-1 and ISO/IEC 5087-2, respectively. Specific references to content defined in ISO/IEC 5087-1 and ISO/IEC 5087-2 are identified in text descriptions of pattern imports, as well as through the explicit identification of terms from ISO/IEC 5087-1 and ISO/IEC 5087-2.

6.2 Transportation network pattern

6.2.1 Overview

The Transportation Network Pattern models the core concepts involved in describing a transportation network. This includes an identification of both physical and administrative characteristics. The most general class is that of the NetworkElement, which can be further classified as one of several types of NetworkElements.

A key feature of this pattern is the formalization of the hasProperPart relationship from a NetworkElement to another NetworkElement. This allows for a representation of networks at multiple levels of detail. For example, at one level, a motorway interchange can be modelled as a single Node. But that node is a NetworkElement that can be defined to consist of an entire TransportNetwork that has a node for each junction within the interchange.

6.2.2 Key classes and properties

6.2.2.1 Network element

A NetworkElement represents any element of a transport network. It can be a part of another NetworkElement and can be decomposed into smaller NetworkElements. Each NetworkElement is characterized with a unique identifier. A NetworkElement can be characterized by its status. Figure 1 provides an overview of a NetworkElement.

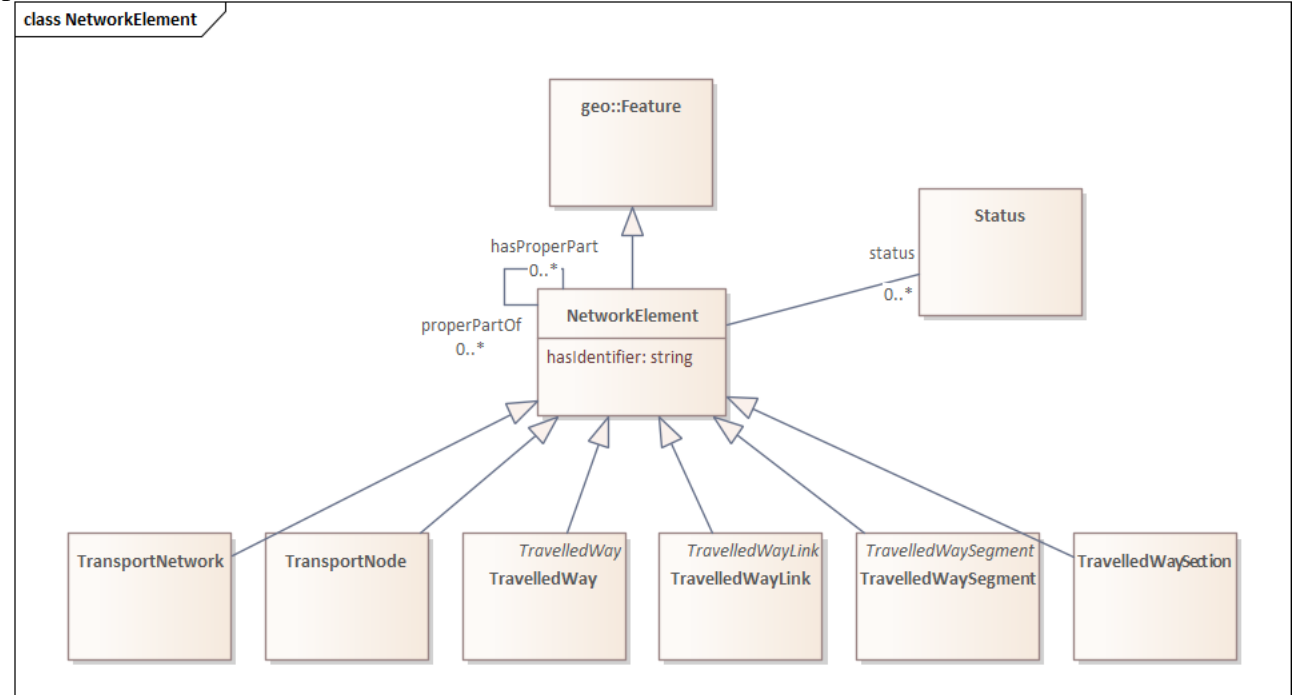


Figure 1—Network element model

6.2.2.2 Transport network

A TransportNetwork is a NetworkElement that is a collection of other network elements that jointly represent a network of paths along which entities (e.g., vehicles, pedestrians) of a specified mode can operate. Figure 2 provides an overview of a TransportNetwork.

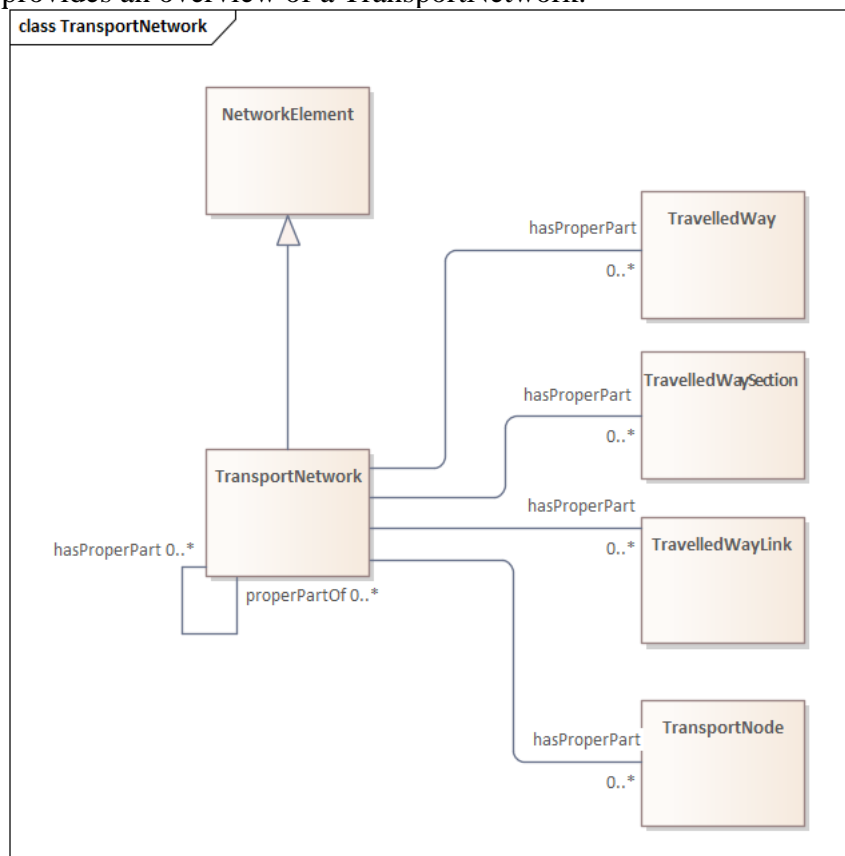


Figure 2—Transport network model

6.2.2.3 Transport node

A TransportNode is a NetworkElement that represents a node on the transport network that can be used to designate an end to a link or to join links. Each TransportNode is a part of at least one TransportNetwork. A TransportNode is characterized by ingress and egress TravelledWayLinks. The ingress and egress link can be the same link if it is bi-directional (e.g., a cul-de-sac). Figure 3 provides an overview of a TransportNode.

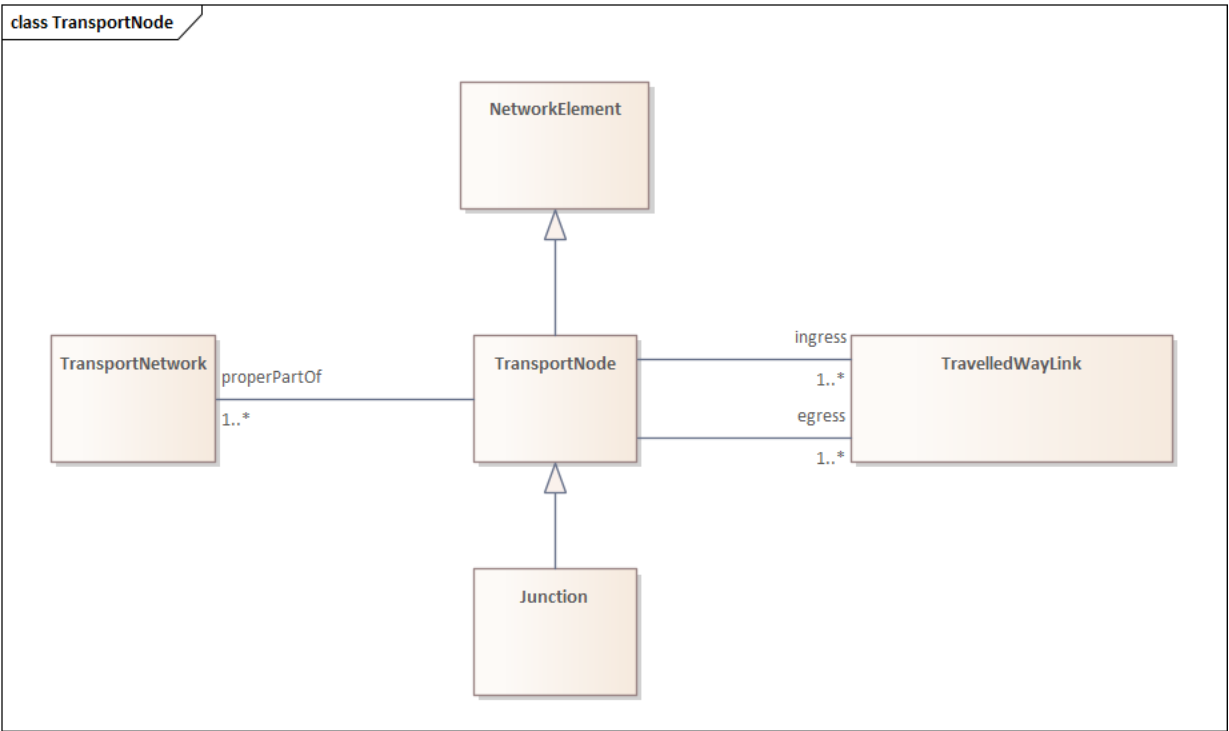


Figure 3—Transport node model

6.2.2.4 Junction

A Junction is a TransportNode that allows a traveller to connect from one TravelledWayLink to another.

6.2.2.5 Travelled way

A TravelledWay is a type of NetworkElement and transinfras:TravelledWay that represents the curvilinear length of a transport route that is identified by a specific designator. It is a generalized class that can be specialized into mode-specific terms (e.g., for a road, rail corridor). Each TravelledWay is defined as being a part of at least one TransportNetwork and can be decomposed into TravelledWayLinks. TravelledWays are identified with a unique designator. Figure 4 provides an overview of a TravelledWay.

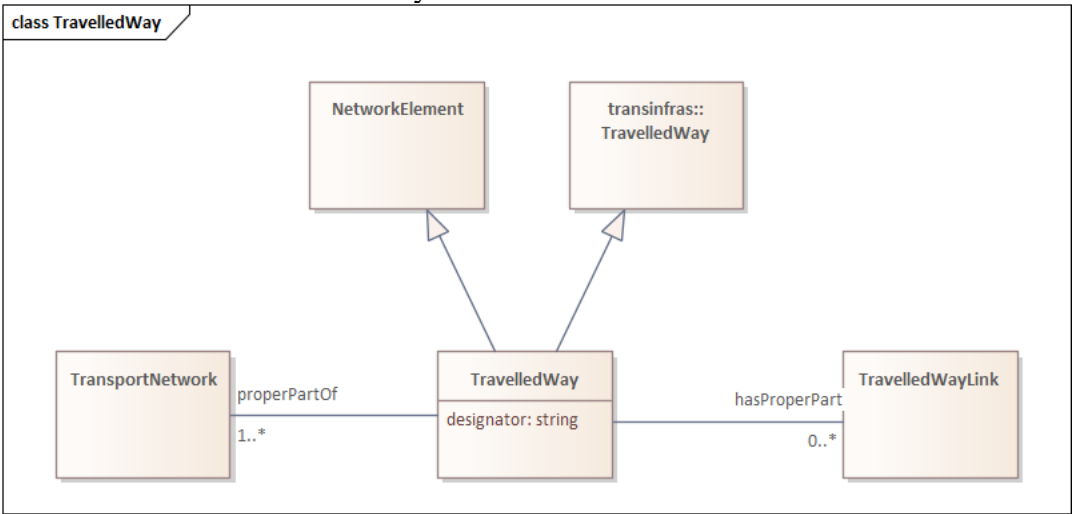


Figure 4—Travelled way model

6.2.2.6 Travelled way link

A TravelledWayLink is a type of NetworkElement and transinfras:TravelledWayLink that serves as the basic building block of a TransportNetwork. It represents a contiguous length of a TravelledWay between two TransportNodes of operational or managerial significance. Each TravelledWayLink is a part of at least one TransportNetwork and can form a part of TravelledWays and TravelledWaySections. A TravelledWayLink can be composed of only TravelledWaySegments. A TravelledWayLink starts from one TransportNode and connects to a second TransportNode. A TravelledWayLink can be characterized by its allowed directions of travel (e.g., forward from the first node to the second node, reverse from the second node to the first, bi-directional, closed, etc.). For example, a TravelledWayLink can be defined from one signalized intersection to the next or from one bus stop to the next. Figure 5 provides an overview of a TravelledWayLink.

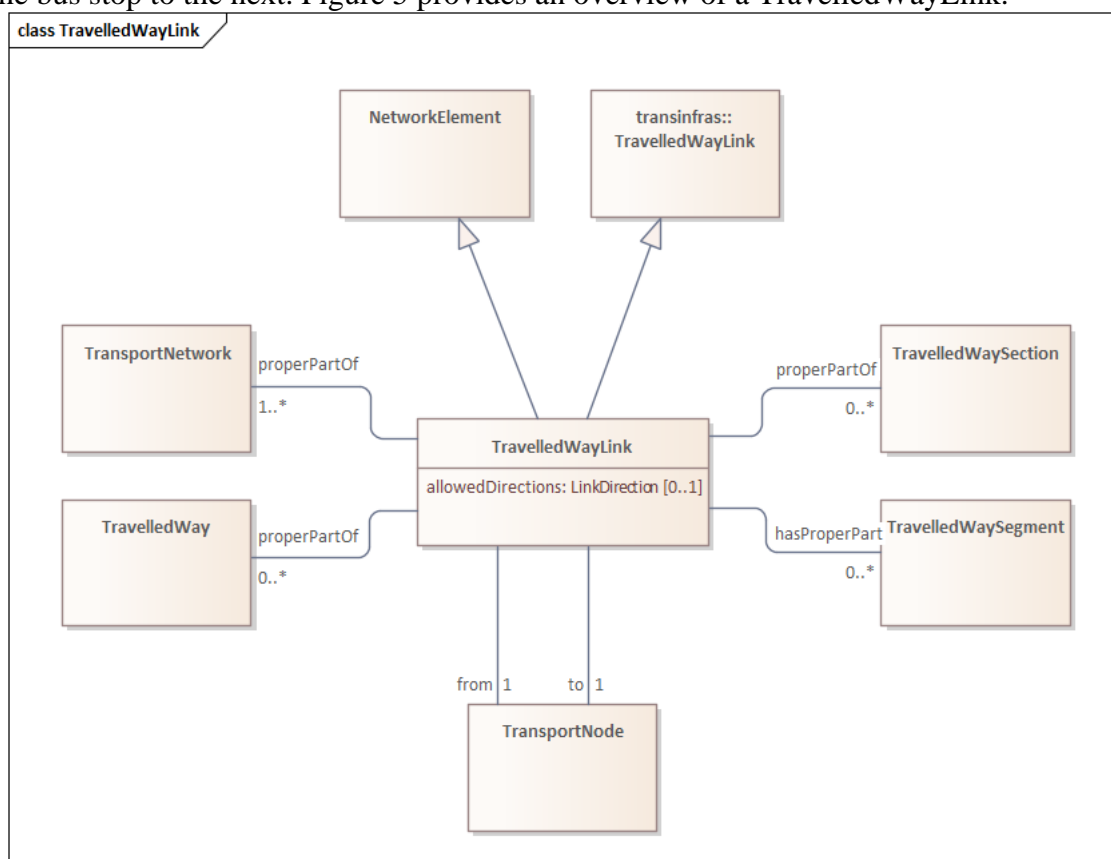


Figure 5—Travelled way link model

6.2.2.7 Travelled way section

A TravelledWaySection is a type of NetworkElement that represents an aggregation of TravelledWayLinks and TravelledWaySegments that jointly represent a contiguous length of a path that shares the same management and operational strategies (within the scope of interest of the implementation). Each TravelledWaySection is part of at least one TransportNetwork. It can only be parts of TransportNetworks and TravelledWays. A TravelledWaySection consists of only TravelledWayLinks.

For example, a TravelledWaySection can be used to represent the portion of a TravelledWay that uses the same traffic signal coordination strategy. It can also be used to represent a bus line or subway line. Figure 6 provides an overview of a TravelledWaySection.

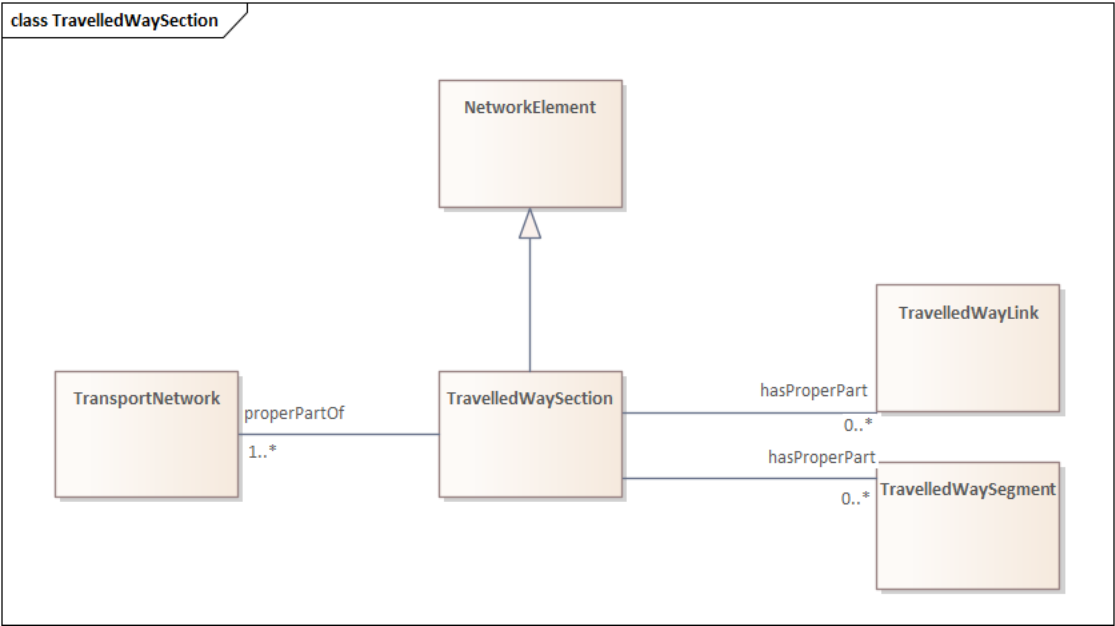


Figure 6—Travelled way section model

6.2.2.8 Travelled way segment

A `TravelledWaySegment` is a type of a `transinfras:TravelledWaySegment` and `NetworkElement` that represents a contiguous length of a `TravelledWayLink` characterized by the same physical characteristics. Each `TravelledWaySegment` is a part of at least one `TravelledWayLink` and can be a part of a `TravelledWaySection` (e.g., if the section does not end at a `TransportNode`). A `TravelledWaySegment` consists of one or more `TravelledWayLanes`.

For example, a `TravelledWayLink` can be defined to connect two `TransportNodes`. If the number of lanes change in the middle of the `TravelledWayLink`, the unique physical characteristics of each component part of the `TravelledWayLink` can be defined by defining separate `TravelledWaySegments`.

Any `TravelledWaySegment` can be defined to be a part of a `TravelCorridorSegment`. Figure 7 provides an overview of a `TravelledWaySegment`.

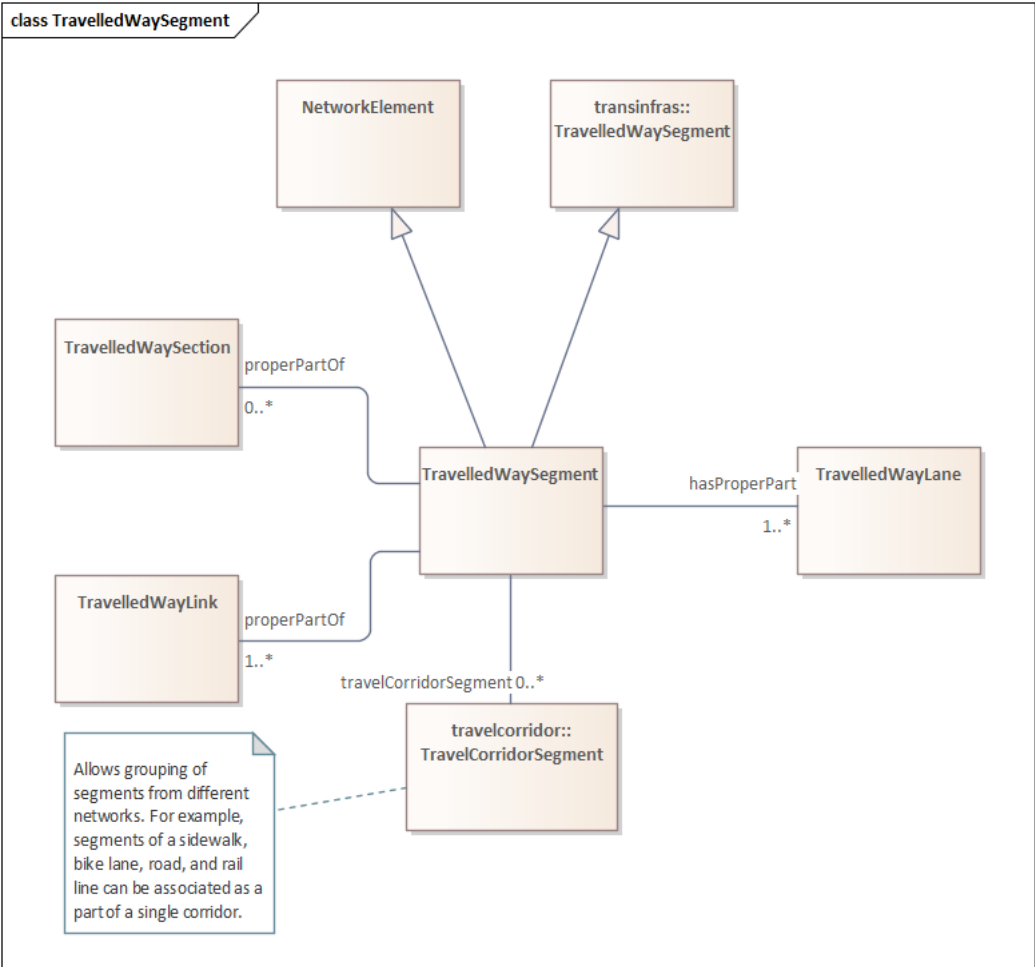


Figure 7—Travelled way segment model

6.2.2.9 Travelled way lane

A **TravelledWayLane** is a **NetworkElement** that is a portion of **TravelledWaySegment** intended to accommodate a single line of moving material entities (e.g., vehicles) along its length. Figure 8 provides an overview of a **TravelledWayLane**.

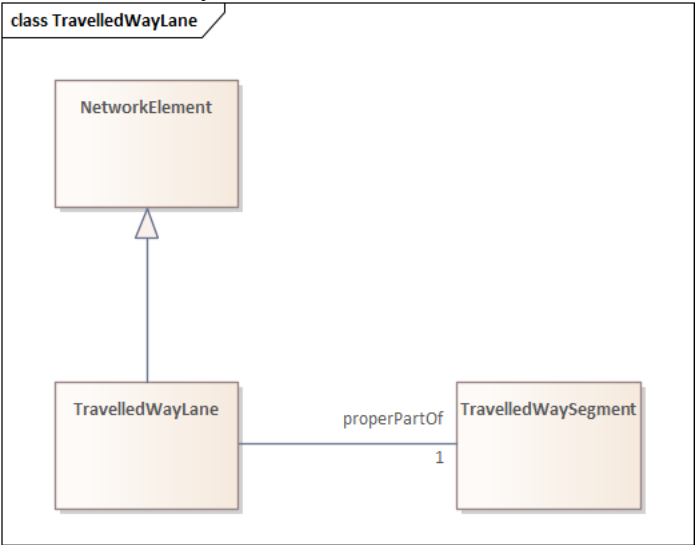


Figure 8—Travelled way lane model

6.2.2.10 Status

A NetworkElement may have a status that specifies the status of the node, e.g., Open or Closed. The status property links to an instance of Status which in turn specifies the period of time for the status using both a period of time representation and separate hasStartTime and hasEndTime properties that use xsd DateTime formats. The value of the code property is an instance of the code:Code class which can refer to a unique identifier in any existing taxonomy for node/link statuses. Figure 9 provides an overview of Status.

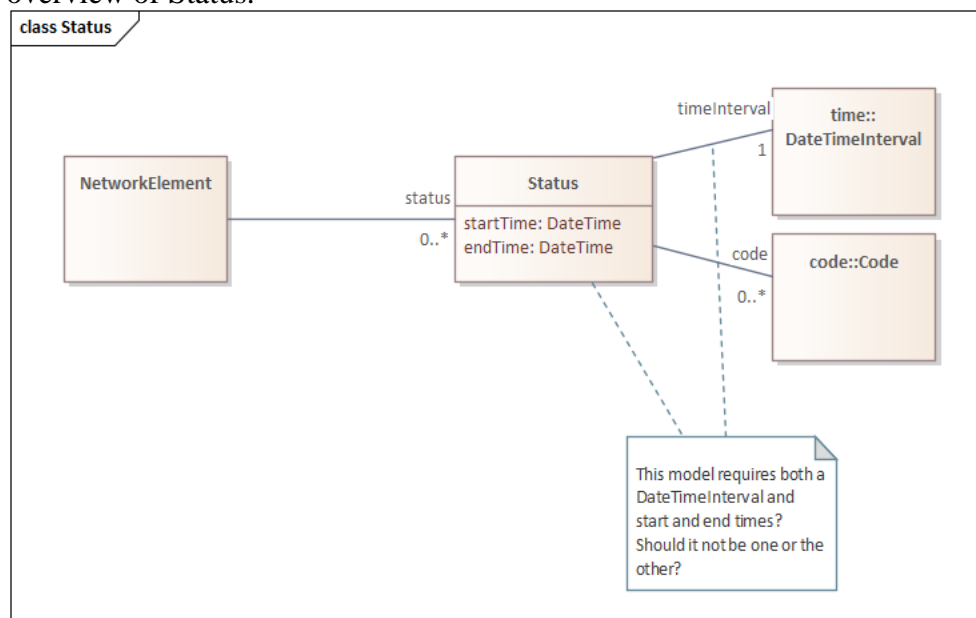


Figure 9—Status model

6.2.3 Formalization

The key classes and properties are formalized in Table 3.

Table 3—Key classes in the transportation network pattern

Class	Property	Mult.	Value restriction
Junction	rdfs:subClassOf		TransportNode
NetworkElement	rdfs:subClassOf		geo:Feature
NetworkElement	status	0..*	Status
NetworkElement	partwhole:properPartOf	0..*	NetworkElement
NetworkElement	partwhole:hasProperPart	0..*	NetworkElement
NetworkElement	genProp:hasIdentifier	1..1	string
Status			NetworkElement
Status	timeInterval	1	time:DateTimeInterval
Status	code	0..*	code:Code
Status	NoteLink		
Status	startTime	1..1	DateTime
Status	endTime	1..1	DateTime
TransportNetwork	rdfs:subClassOf		NetworkElement
TransportNetwork	partwhole:hasProperPart	0..*	TravelledWay
TransportNetwork	partwhole:hasProperPart	0..*	TravelledWaySection
TransportNetwork	partwhole:hasProperPart	0..*	TravelledWayLink
TransportNetwork	partwhole:hasProperPart	0..*	TransportNode

Class	Property	Mult.	Value restriction
TransportNetwork	partwhole:properPartOf	0..*	TransportNetwork
TransportNetwork	partwhole:hasProperPart	0..*	TransportNetwork
TransportNode	rdfs:subClassOf		NetworkElement
TransportNode	ingress	1..*	TravelledWayLink
TransportNode	egress	1..*	TravelledWayLink
TransportNode	partwhole:properPartOf	1..*	TransportNetwork
TravelledWay	rdfs:subClassOf		NetworkElement
TravelledWay	rdfs:subClassOf		transinfrs:TravelledWay
TravelledWay	partwhole:properPartOf	1..*	TransportNetwork
TravelledWay	partwhole:hasProperPart	0..*	TravelledWayLink
TravelledWay	designator	1..1	string
TravelledWayLane	rdfs:subClassOf		NetworkElement
TravelledWayLane	partwhole:properPartOf	1	TravelledWaySegment
TravelledWayLink	to	1	TransportNode
TravelledWayLink	from	1	TransportNode
TravelledWayLink	partwhole:properPartOf	0..*	TravelledWay
TravelledWayLink	partwhole:properPartOf	0..*	TravelledWaySection
TravelledWayLink	rdfs:subClassOf		NetworkElement
TravelledWayLink	rdfs:subClassOf		transinfrs:TravelledWayLink
TravelledWayLink	partwhole:properPartOf	1..*	TransportNetwork
TravelledWayLink	partwhole:hasProperPart	0..*	TravelledWaySegment
TravelledWayLink	allowedDirections	0..1	LinkDirection
TravelledWaySection	rdfs:subClassOf		NetworkElement
TravelledWaySection	partwhole:properPartOf	1..*	TransportNetwork
TravelledWaySection	partwhole:hasProperPart	0..*	TravelledWayLink
TravelledWaySection	partwhole:hasProperPart	0..*	TravelledWaySegment
TravelledWaySegment	partwhole:properPartOf	1..*	TravelledWayLink
TravelledWaySegment	rdfs:subClassOf		transinfrs:TravelledWaySegment
TravelledWaySegment	rdfs:subClassOf		NetworkElement
TravelledWaySegment	partwhole:properPartOf	0..*	TravelledWaySection
TravelledWaySegment	partwhole:hasProperPart	1..*	TravelledWayLane
TravelledWaySegment	travelCorridorSegment	0..*	travelcorridor:TravelCorridorSegment

6.3 Road network pattern

6.3.1 Overview

The road network pattern specializes the transportation network pattern to address the needs for vehicular travel. Road networks that are primarily designed for the movement of micromobility vehicles should generally be defined using the micromobility network pattern, which is a specialization of this pattern.

6.3.2 Key classes and properties

6.3.2.1 Road network

A RoadNetwork is a type of TransportNetwork using a stabilized base designed for the movement of vehicles, other than rail or air vehicles, that conform to a specified set of requirements but may be used by others as well.

RoadNetworks are made up of RoadLinks, Roads, and RoadSections.

Figure 10 provides an overview of the RoadNetwork.

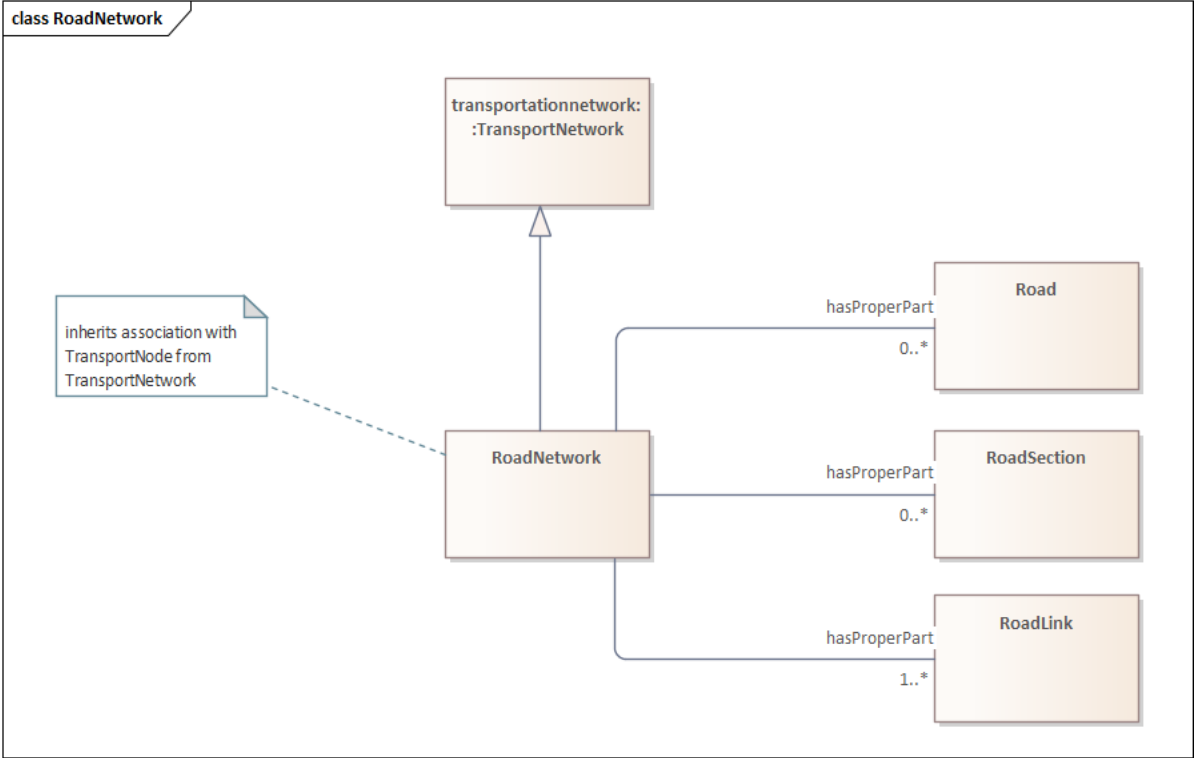


Figure 10—Road network

6.3.2.2 Road

A Road is a type of TravelledWay and transinfras:Road that is made up of RoadLinks. Roads form a proper part of RoadNetworks.

The extent of a Road is defined by the extent of the roadway that shares the designator assigned to the Road. As such, a single RoadLink can be designated to be part of multiple Roads. For example, Interstate 10 (I-10) and US-90 are defined as two distinct roads that run across the southern United States. In various locations, I-10 and US-90 share the same physical infrastructure and those RoadLinks form part of both Roads. Further, portions of I-10 have localized names. For example, the stretch of I-10 from downtown Houston to Katy, Texas is also known as "Katy Freeway". This can be defined as its own distinct Road, which is entirely aligned with I-10, but has a much shorter length. Figure 11 provides an overview of Road.

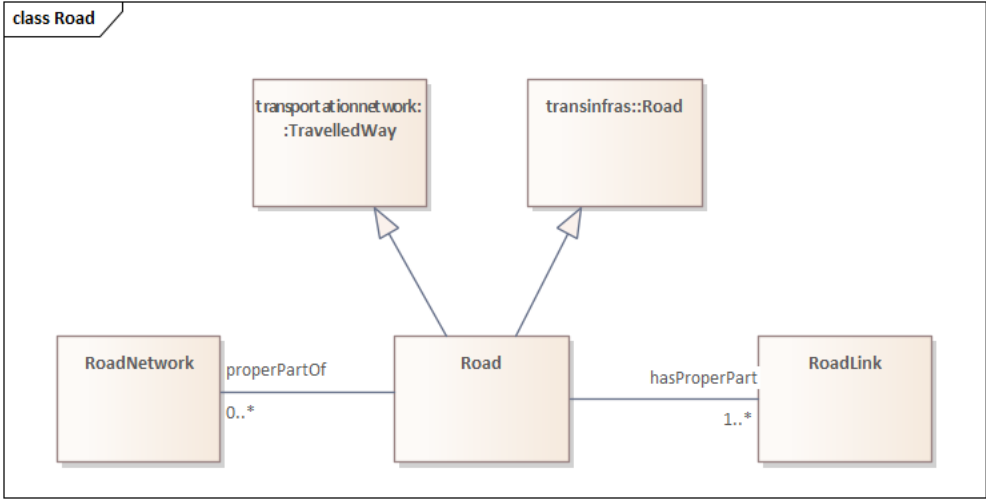


Figure 11—Road

6.3.2.3 Road link

A RoadLink is a type of TravelledWayLink and transinfras:RoadLink using a stabilized base designed for the movement of vehicles that conform to a specified set of requirements but may be used by others as well. TravelledWayLinks that are primarily designed as air strips or for use by rail vehicles should not be defined as a RoadLinks. A RoadLink is a basic building block of a RoadNetwork.

RoadLinks can be grouped to form Roads (i.e., a sequence of RoadLinks that share a common designator) and RoadSections (i.e., a sequence of RoadLinks that are grouped for operational purposes). RoadLinks can be subdivided into multiple RoadSegments.

A RoadLink can be characterized by its capacity, the maximum number of lanes defined in the link, and RoadLinkUser characteristics that can define characteristics such as the speed limit or average travel time for the link based on the type of user. For example, the speed limit for lorries can be different than the speed limit for light duty vehicles.

Figure 12 provides an overview of RoadLink.

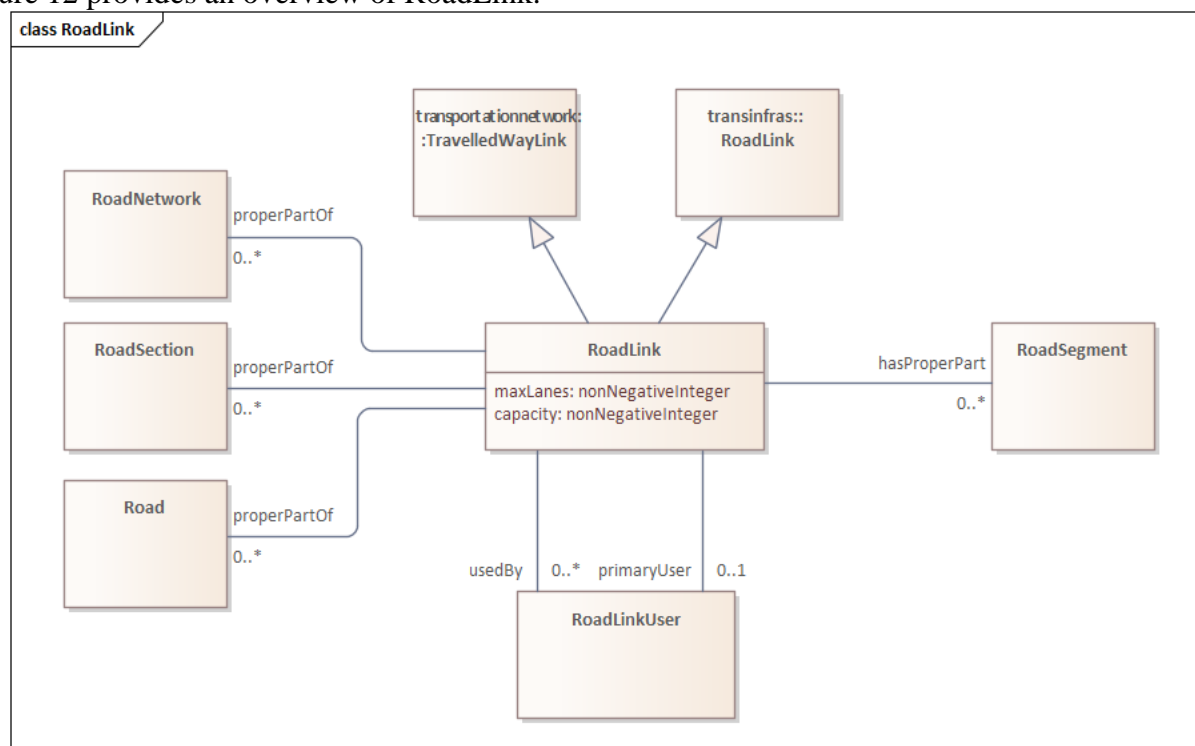


Figure 12—Road link model

6.3.2.4 Road section

A RoadSection is a type of TravelledWaySection that groups RoadLinks and RoadSegments for a useful operational purpose (e.g., assigning a speed limit, designating a traffic control scheme). RoadSections can be defined as part of a RoadNetwork. Figure 13 provides an overview of RoadSection.

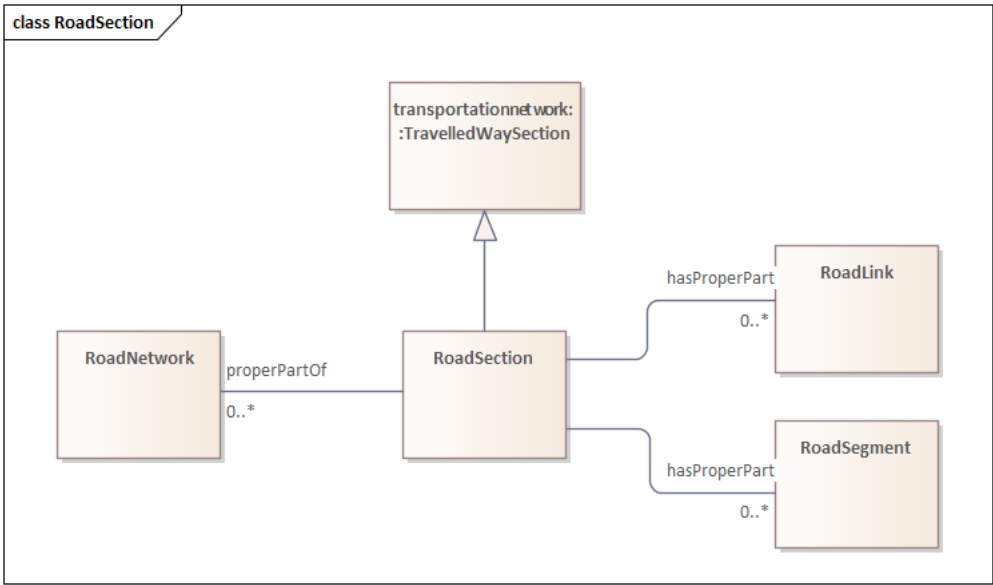


Figure 13—Road section model

6.3.2.5 Road segment

A RoadSegment is a type of TravelledWaySegment and transinfas:RoadSegment that represents a portion of a RoadLink with common physical characteristics. A RoadSegment can be defined to be a part of a RoadSection, especially when the RoadSection does not span an entire RoadLink. A RoadSegment consists of one or more RoadLanes. A RoadSegment can be characterized by its number of lanes, width, and pavement type. Figure 14 provides an overview of RoadSegment.

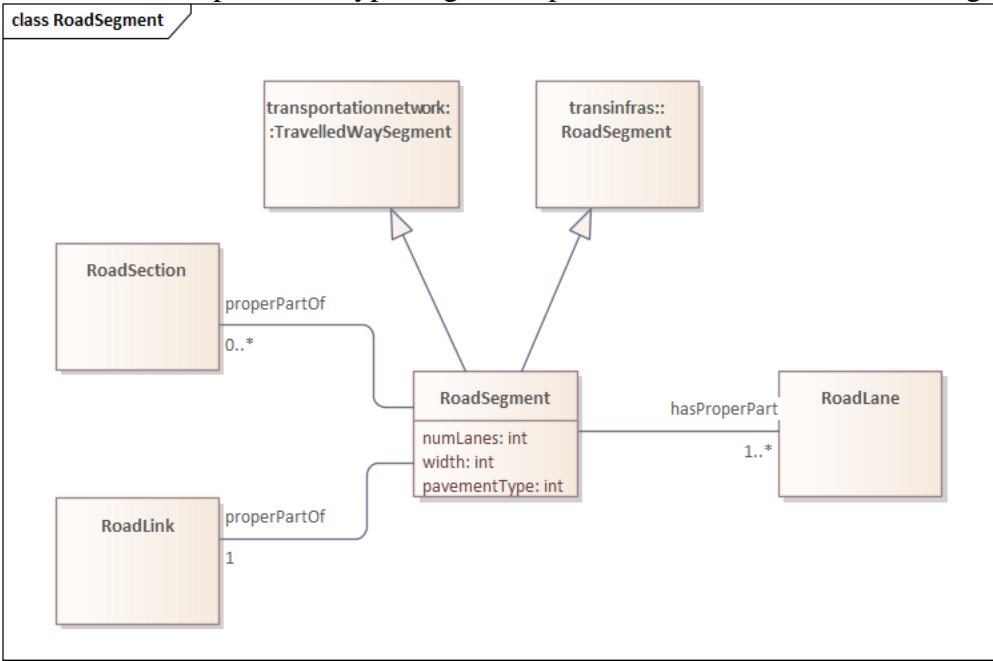


Figure 14—Road segment model

6.3.2.6 Road lane

A RoadLane is a type of TravelledWayLane that forms part of a RoadSegment. A RoadLane can be characterized by its width. A road lane can connect to road lanes associated with the next or previous RoadSegment to indicate how the lanes of the two segments align. For example, when a lane is added expanding a RoadLink from three to four lanes, there will be two RoadSegments. The lanes of each RoadSegment can be linked to indicate where the new lane is added. Figure 15 provides an overview of RoadLane.

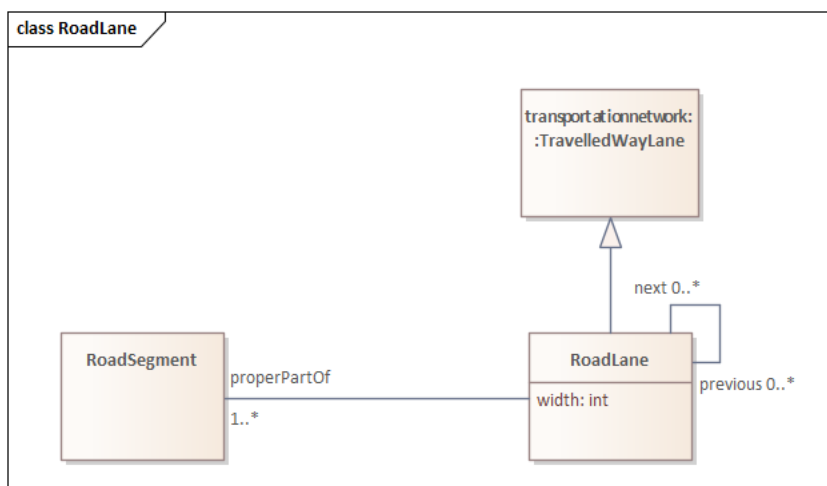


Figure 15—Road lane model

6.3.3 Formalization

The key classes and properties are formalized in Table 4.

Table 4—Key classes in the road network pattern

Class	Property	Mult.	Value restriction
Road	rdfs:subClassOf		transportationnetwork:TravelledWay
Road	rdfs:subClassOf		transinfras:Road
Road	partwhole:hasProperPart	1..*	RoadLink
Road	partwhole:properPartOf	0..*	RoadNetwork
RoadLane	rdfs:subClassOf		transportationnetwork:TravelledWayLane
RoadLane	partwhole:properPartOf	1..*	RoadSegment
RoadLane	next	0..*	RoadLane
RoadLane	previous	0..*	RoadLane
RoadLane	width	1..1	int
RoadLink	rdfs:subClassOf		transportationnetwork:TravelledWayLink
RoadLink	rdfs:subClassOf		transinfras:RoadLink
RoadLink	partwhole:properPartOf	0..*	Road
RoadLink	partwhole:properPartOf	0..*	RoadNetwork
RoadLink	partwhole:properPartOf	0..*	RoadSection
RoadLink	partwhole:hasProperPart	0..*	RoadSegment
RoadLink	usedBy	0..*	RoadLinkUser
RoadLink	primaryUser	0..1	RoadLinkUser
RoadLink	maxLanes	1..1	nonNegativeInteger
RoadLink	capacity	1..1	nonNegativeInteger
RoadLinkUser	travelTime	0..1	cityunits:Duration
RoadLinkUser	speedLimit	0..1	cityunits:Speed
RoadLinkUser	uses	1	RoadLink
RoadLinkUser	uses	1	RoadLink
RoadLinkUser	userCategory	1..1	RoadUserCategoryEnum
RoadNetwork	partwhole:hasProperPart	0..*	Road
RoadNetwork	partwhole:hasProperPart	0..*	RoadSection
RoadNetwork	partwhole:hasProperPart	1..*	RoadLink
RoadNetwork	rdfs:subClassOf		transportationnetwork:TransportNetwork
RoadNetwork	NoteLink		
RoadSection	partwhole:properPartOf	0..*	RoadNetwork

Class	Property	Mult.	Value restriction
RoadSection	partwhole:hasProperPart	0..*	RoadLink
RoadSection	rdfs:subClassOf		transportationnetwork:TravelledWaySection
RoadSection	partwhole:hasProperPart	0..*	RoadSegment
RoadSegment	partwhole:hasProperPart	1..*	RoadLane
RoadSegment	partwhole:properPartOf	1	RoadLink
RoadSegment	partwhole:properPartOf	0..*	RoadSection
RoadSegment	rdfs:subClassOf		transportationnetwork:TravelledWaySegment
RoadSegment	rdfs:subClassOf		transinfras:RoadSegment
RoadSegment	numLanes	1..1	int
RoadSegment	width	1..1	int
RoadSegment	pavementType	1..1	int

6.4 Rail network pattern

6.4.1 Overview

The rail network pattern specializes the transportation network pattern to address the needs for travel of rail vehicles (e.g., trains, subways, trams).

6.4.2 Key classes and properties

6.4.2.1 Rail network

A RailNetwork is a type of TransportNetwork using rails on a stabilized base. RailNetworks are made up of RailCorridors, RailSections, and TrackLinks. Figure 16 provides an overview of RailNetwork.

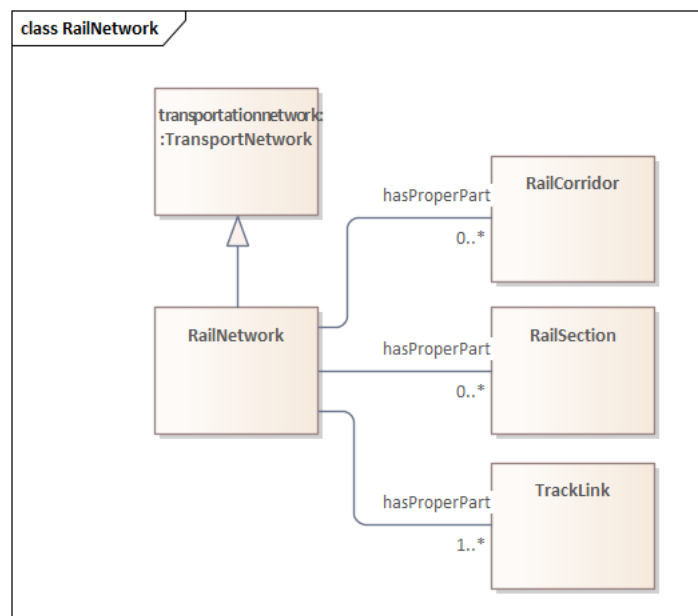


Figure 16—Rail network

6.4.2.2 Rail corridor

A RailCorridor is a type of TravelledWay that is made up of TrackLinks. RailCorridors form a proper part of RailNetworks.

The extent of a RailCorridor is defined by the extent of the railway that shares the designator assigned to the RailCorridor. Figure 17 provides an overview of RailCorridor.

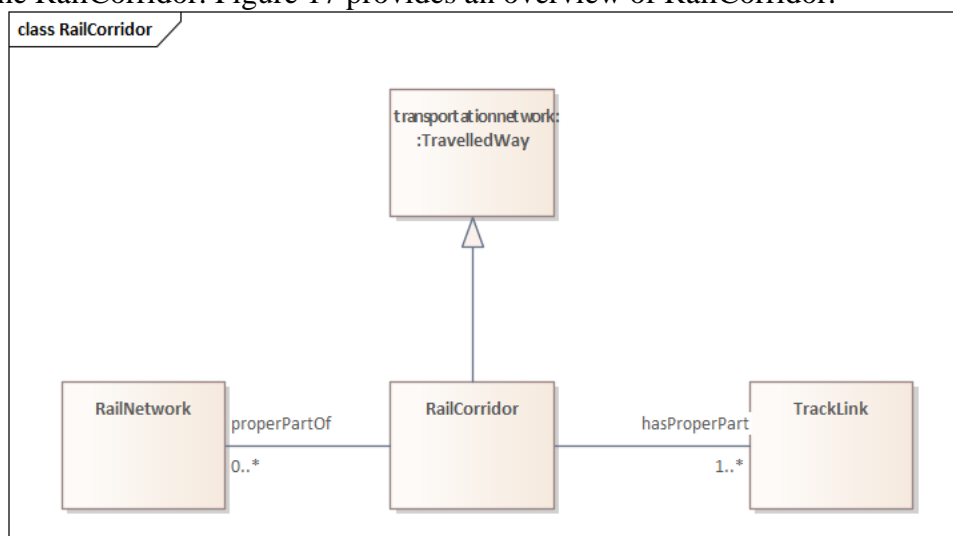


Figure 17—Rail corridor

6.4.2.3 Track link

A TrackLink is a type of TravelledWayLink that uses rails on a stabilized base. A TrackLink is a basic building block of a RailNetwork. Due to the nature of rail, each TrackLink consists of a single lane, but multiple TrackLinks can exist along the same RailCorridor.

TrackLinks can be grouped to form RailCorridors (i.e., a sequence of TrackLinks that share a common designator) and RailSections (i.e., a sequence of TrackLinks that are grouped for operational purposes). TrackLinks can be subdivided into multiple TrackSegments. Figure 18 provides an overview of TrackLink.

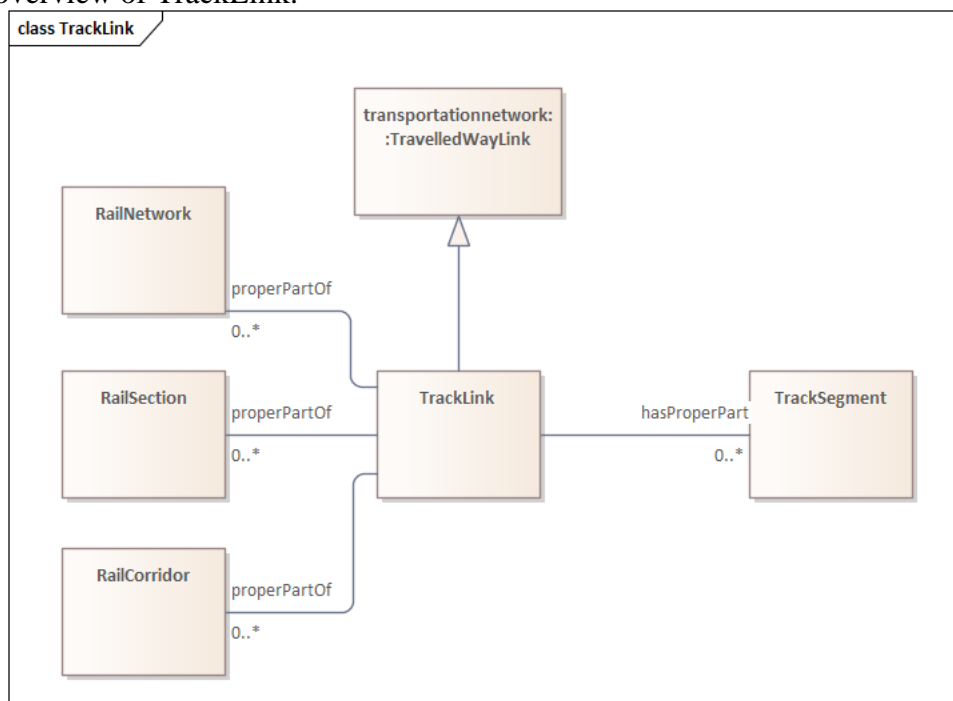


Figure 18—Track link

6.4.2.4 Rail section

A RailSection is a type of TravelledWaySection that groups TrackLinks and TrackSegments for a useful operational purpose (e.g., assigning a speed limit, designating a traffic control scheme). RoadSections can be defined as part of a RoadNetwork. Figure 19 provides an overview of RailSection.

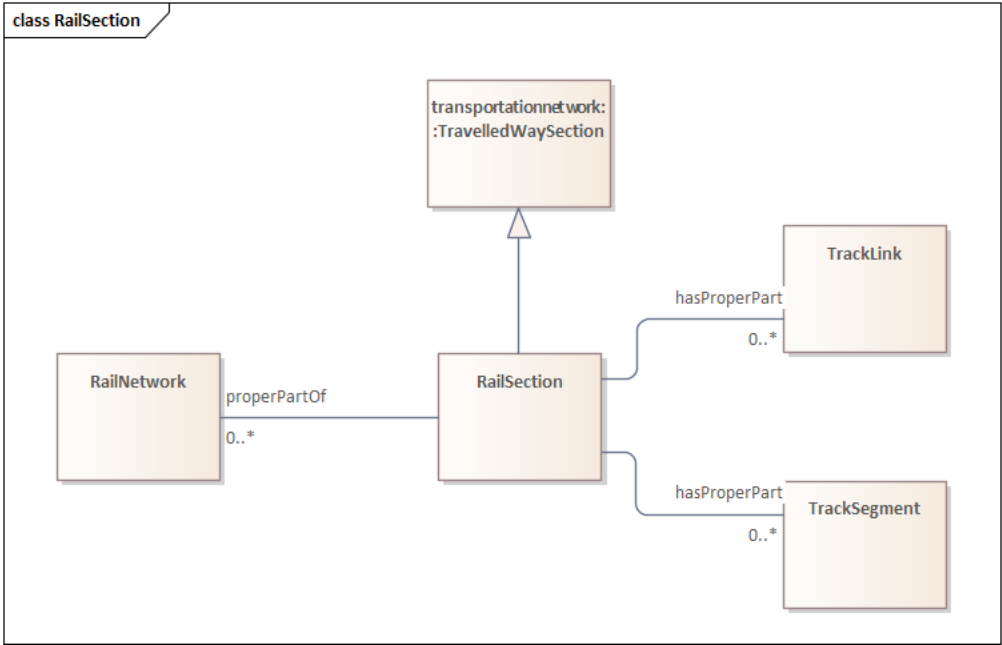


Figure 19—Rail section

6.4.2.5 Track segment

A TrackSegment is a type of TravelledWaySegment that represents a portion of a TrackLink with common physical characteristics. A TrackSegment can be defined to be a part of a RailSection, especially when the RailSection does not span an entire TrackLink. Figure 20 provides an overview of TrackSegment.

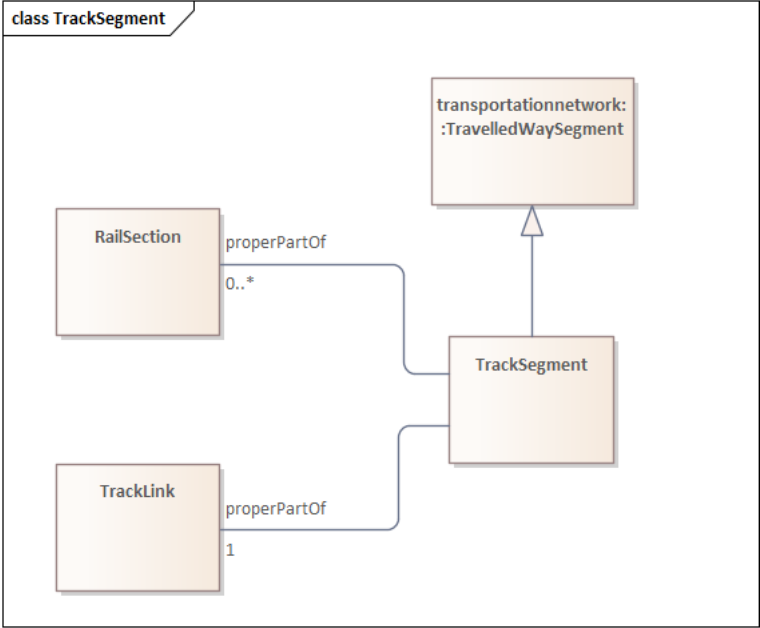


Figure 20—Track segment

6.4.3 Formalization

The key classes and properties are formalized in Table 5..

Table 5—Key classes in the rail network pattern

Class	Property	Mult.	Value restriction
RailCorridor	rdfs:subClassOf		transportationnetwork:TravelledWay
RailCorridor	partwhole:properPartOf	0..*	RailNetwork
RailCorridor	partwhole:hasProperPart	1..*	TrackLink
RailNetwork	rdfs:subClassOf		transportationnetwork:TransportNetwork
RailNetwork	partwhole:hasProperPart	0..*	RailCorridor
RailNetwork	partwhole:hasProperPart	0..*	RailSection
RailNetwork	partwhole:hasProperPart	1..*	TrackLink
RailSection	rdfs:subClassOf		transportationnetwork:TravelledWaySection
RailSection	partwhole:properPartOf	0..*	RailNetwork
RailSection	partwhole:hasProperPart	0..*	TrackLink
RailSection	partwhole:hasProperPart	0..*	TrackSegment
TrackLink	rdfs:subClassOf		transportationnetwork:TravelledWayLink
TrackLink	partwhole:properPartOf	0..*	RailNetwork
TrackLink	partwhole:properPartOf	0..*	RailCorridor
TrackLink	partwhole:properPartOf	0..*	RailSection
TrackLink	partwhole:hasProperPart	0..*	TrackSegment
TrackSegment	rdfs:subClassOf		transportationnetwork:TravelledWaySegment
TrackSegment	partwhole:properPartOf	0..*	RailSection
TrackSegment	partwhole:properPartOf	1	TrackLink

6.5 Micromobility network pattern

6.5.1 Overview

The micromobility network pattern specializes the road network pattern to address the needs for the travel of micromobility vehicles.

6.5.2 Key classes and properties

6.5.2.1 Micromobility network

A MicromobilityNetwork is a type of RoadNetwork designed for the use of micromobility vehicles, which have more limited performance characteristics than motor vehicles.

MicromobilityNetworks are made up of MicromobilityLinks, MicromobilityPaths, and MicromobilityPathSections. Figure 21 provides an overview of MicromobilityNetwork.

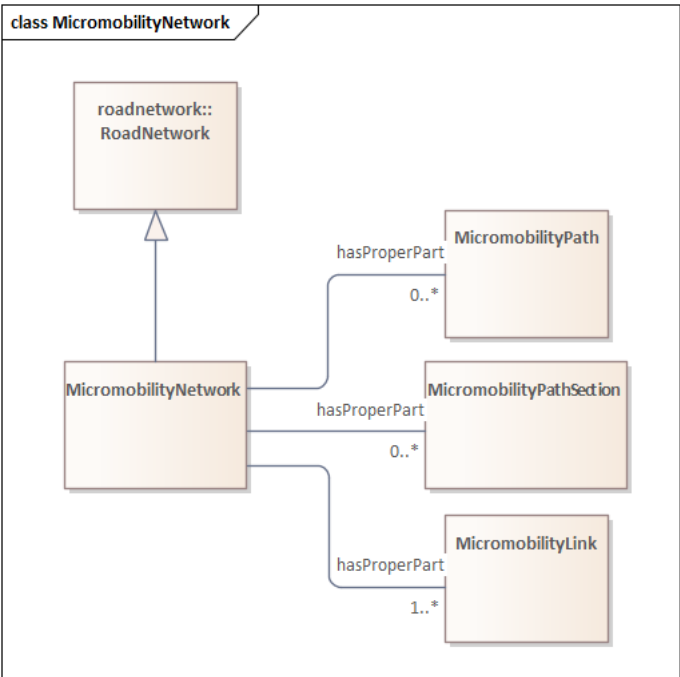


Figure 21—Micromobility network

6.5.2.2 Micromobility path

A MicromobilityPath is a type of Road that is made up of MicromobilityPathLinks. MicromobilityPaths form a proper part of MicromobilityNetworks. Figure 22 provides an overview of MicromobilityPath.

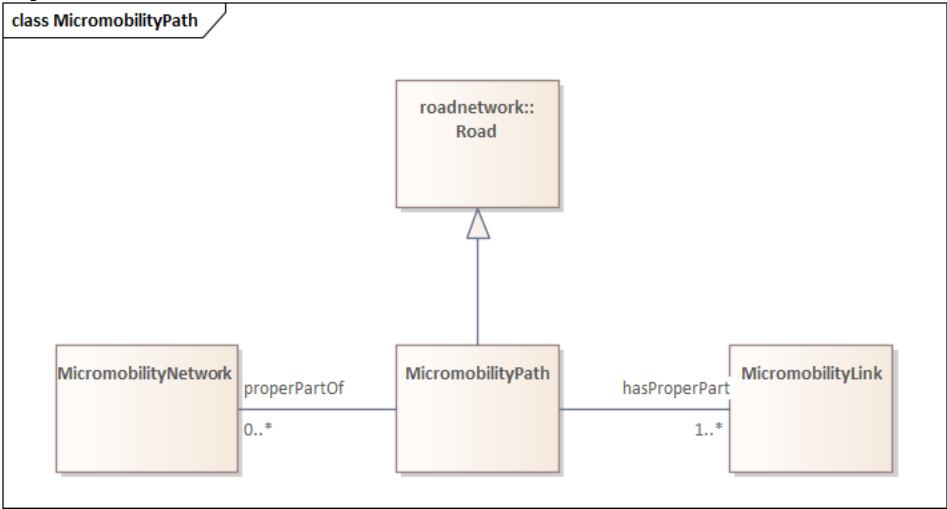


Figure 22— MicromobilityPath model

6.5.2.3 Micromobility link

A MicromobilityLink is a type of RoadLink designed for micromobility vehicles. A MicromobilityLink is a basic building block of MicromobilityNetworks.

MicromobilityLinks can be grouped to form can form MicromobilityPaths (i.e., a sequence of MicromobilityLinks that share a common designator) and MicromobilityPathSections (i.e., a sequence of RoadLinks that are grouped for operational purposes). MicromobilityLinks can be subdivided into multiple MicromobilityPathSegments. Figure 23 provides an overview of MicromobilityLink.

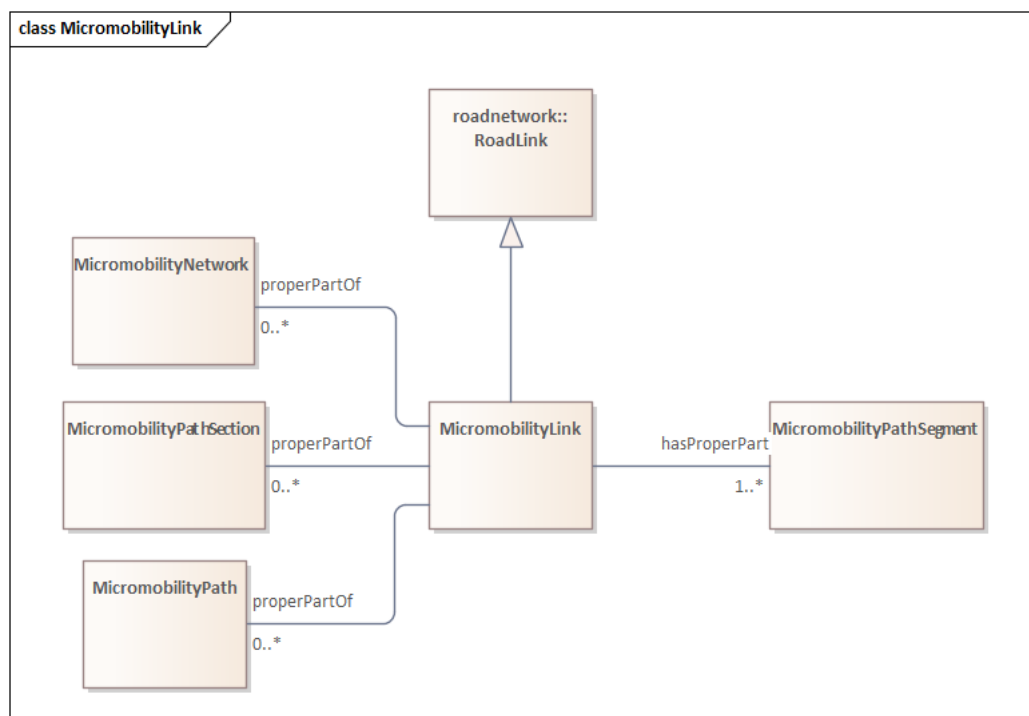


Figure 23—Micromobility link

6.5.2.4 Micromobility path section

A **MicromobilityPathSection** is a type of **RoadSection** that groups **MicromobilityLinks** and **MicromobilityPathSegments** for a useful operational purpose (e.g., assigning a speed limit, designating areas of shared use). **MicromobilityPathSections** can be defined as part of a **MicromobilityNetwork**. Figure 24 provides an overview of **MicromobilityPathSection**.

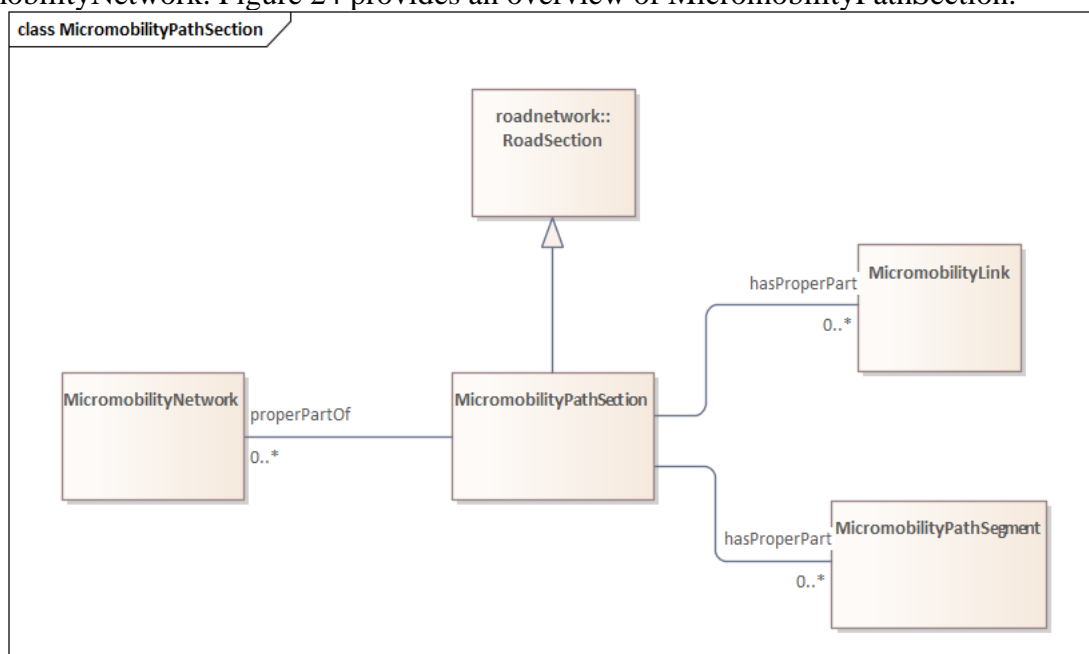


Figure 24— Micromobility path section model

6.5.2.5 Micromobility path segment

A **MicromobilityPathSegment** is a type of **RoadSegment** that represents a portion of a **MicromobilityLink** with common physical characteristics. A **MicromobilityPathSegment** can be defined to be a part of a **MicromobilityPathSection**, especially when the **MicromobilityPathSection**

does not span an entire MicromobilityLink. A MicromobilityPathSegment consists of one or more MicromobilityLanes. A MicromobilityPathSegment can be characterized by its number of lanes, width, and pavement type, as inherited from RoadSegment. Figure 25 provides an overview of MicromobilityPathSegment.

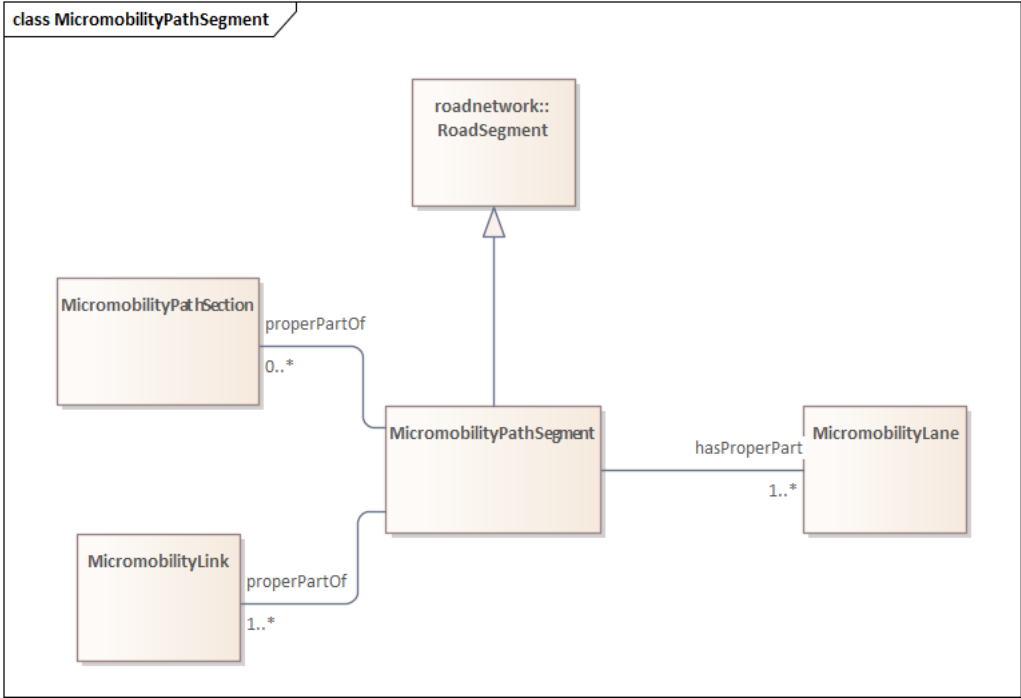


Figure 25—Road segment model

6.5.2.6 Micromobility lane

A MicromobilityLane is a type of RoadLane that forms part of a MicromobilityPathSegment. Figure 26 provides an overview of MicromobilityLane.

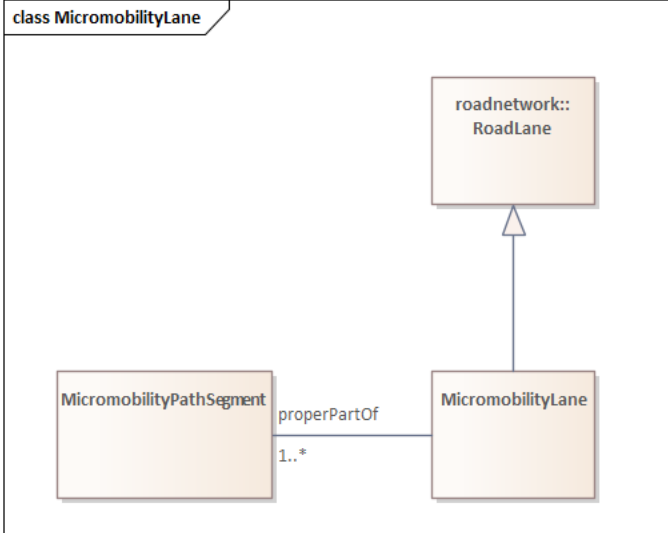


Figure 26—Micromobility lane model

6.5.3 Formalization

The key classes and properties are formalized in Table 6..

Table 6—Key classes in the micromobility network pattern

Class	Property	Mult.	Value restriction
MicromobilityLane	rdfs:subClassOf		roadnetwork:RoadLane
MicromobilityLane	partwhole:properPartOf	1..*	MicromobilityPathSegment

Class	Property	Mult.	Value restriction
MicromobilityLink	rdfs:subClassOf		roadnetwork:RoadLink
MicromobilityLink	partwhole:properPartOf	0..*	MicromobilityNetwork
MicromobilityLink	partwhole:hasProperPart	1..*	MicromobilityPathSegment
MicromobilityLink	partwhole:properPartOf	0..*	MicromobilityPathSection
MicromobilityLink	partwhole:properPartOf	0..*	MicromobilityPath
MicromobilityNetwork	rdfs:subClassOf		roadnetwork:RoadNetwork
MicromobilityNetwork	partwhole:hasProperPart	0..*	MicromobilityPath
MicromobilityNetwork	partwhole:hasProperPart	0..*	MicromobilityPathSection
MicromobilityNetwork	partwhole:hasProperPart	1..*	MicromobilityLink
MicromobilityPath	rdfs:subClassOf		roadnetwork:Road
MicromobilityPath	partwhole:properPartOf	0..*	MicromobilityNetwork
MicromobilityPath	partwhole:hasProperPart	1..*	MicromobilityLink
MicromobilityPathSection	rdfs:subClassOf		roadnetwork:RoadSection
MicromobilityPathSection	partwhole:properPartOf	0..*	MicromobilityNetwork
MicromobilityPathSection	partwhole:hasProperPart	0..*	MicromobilityLink
MicromobilityPathSection	partwhole:hasProperPart	0..*	MicromobilityPathSegment
MicromobilityPathSegment	rdfs:subClassOf		roadnetwork:RoadSegment
MicromobilityPathSegment	partwhole:hasProperPart	1..*	MicromobilityLane
MicromobilityPathSegment	partwhole:properPartOf	1..*	MicromobilityLink
MicromobilityPathSegment	partwhole:properPartOf	0..*	MicromobilityPathSection

6.6 Pedestrian network pattern

6.6.1 Overview

The pedestrian network pattern specializes the transportation network pattern to address the needs for pedestrian travel.

6.6.2 Key classes and properties

6.6.2.1 Footpath network

A FootpathNetwork is a type of TransportNetwork designed for the use of pedestrians but may be used by others as well.

FootpathNetworks are made up of FootpathLinks, Footpaths, and FootpathSections. Figure 27 provides an overview of FootpathNetwork.

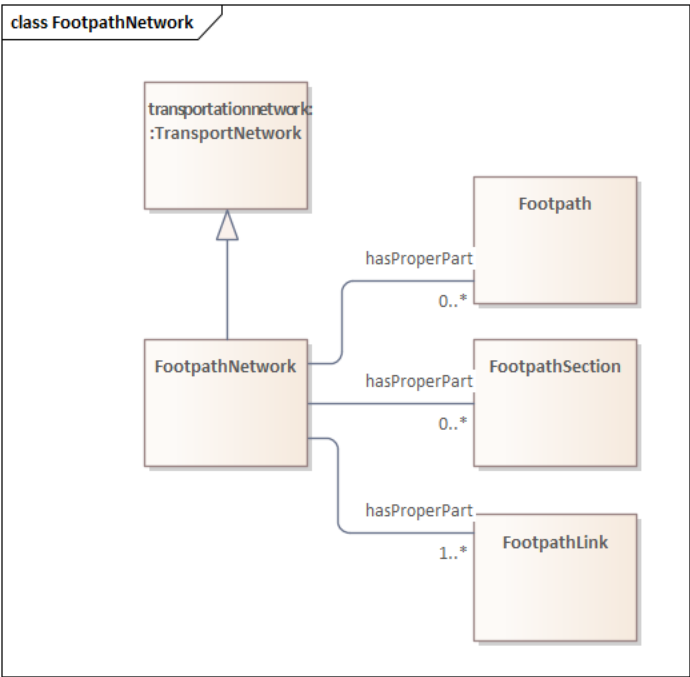


Figure 27—Footpath network

6.6.2.2 Footpath

A Footpath is a type of TravelledWay that is made up of FootpathLinks. Footpaths form a proper part of FootpathNetworks.

The extent of a Footpath is defined by the extent of the path that shares the designator assigned to the Footpath. Figure 28 provides an overview of Footpath.

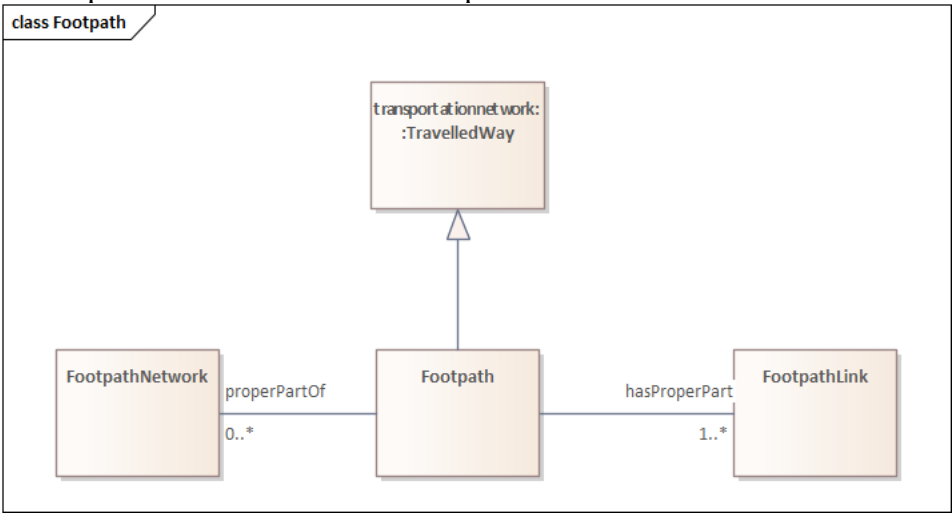


Figure 28—Footpath

6.6.2.3 Footpath link

A Footpath Link is a type of TravelledWayLink designed for pedestrians. A FootpathLink is a basic building block of a FootpathNetwork.

FootpathLinks can be grouped to form Footpaths (i.e., a sequence of FootpathLinks that share a common designator) and FootpathSections (i.e., a sequence of FootpathLinks that are grouped for operational purposes). FootpathLinks can be subdivided into multiple FootpathSegments. Figure 29 provides an overview of FootpathLink.

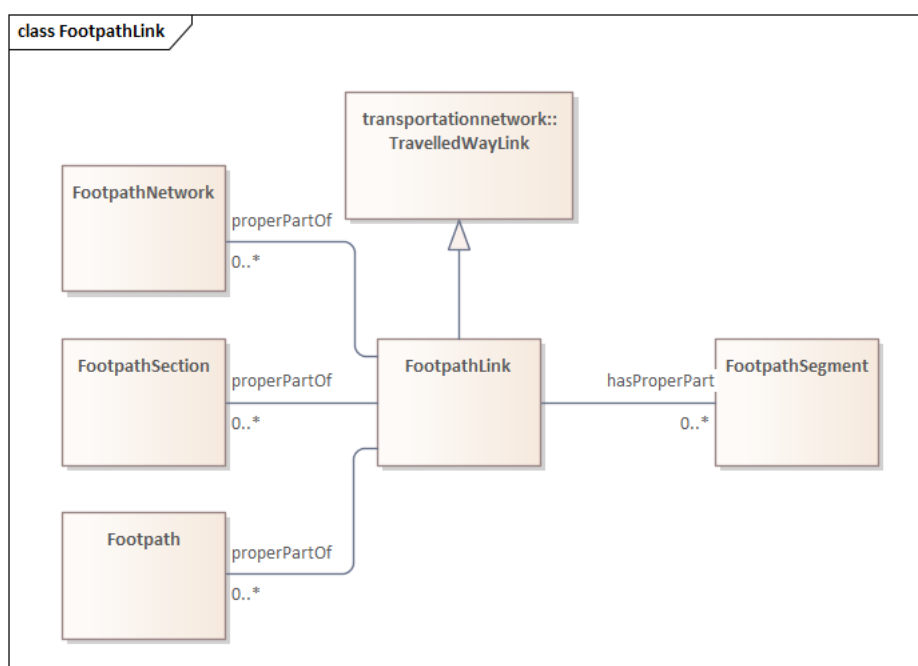


Figure 29—Footpath link model

6.6.2.4 Footpath section

A FootpathSection is a type of TravelledWaySection that groups FootpathLinks and FootpathSegments for a useful operational purpose. FootpathSections can be defined as part of a FootpathNetwork. Figure 30 provides an overview of FootpathSection.

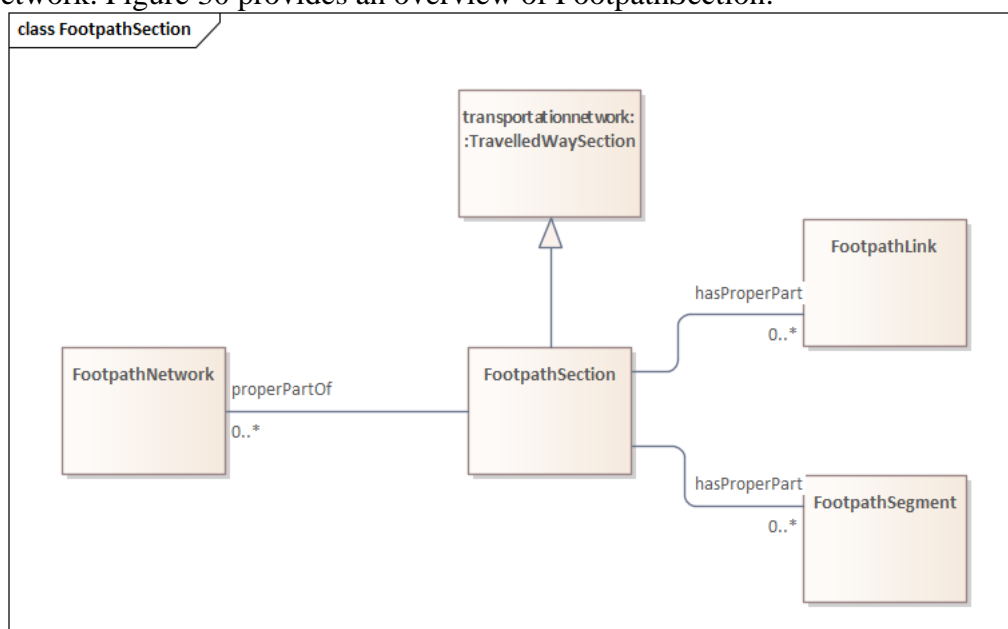


Figure 30—Footpath section model

6.6.2.5 Footpath segment

A FootpathSegment is a type of TravelledWaySegment that represents a portion of a FootpathLink with common physical characteristics. A FootpathSegment can be defined to be a part of a FootpathSection, especially when the FootpathSection does not span an entire FootpathLink. A FootpathSegment consists of one or more FootpathLanes. Figure 31 provides an overview of FootpathSegment.

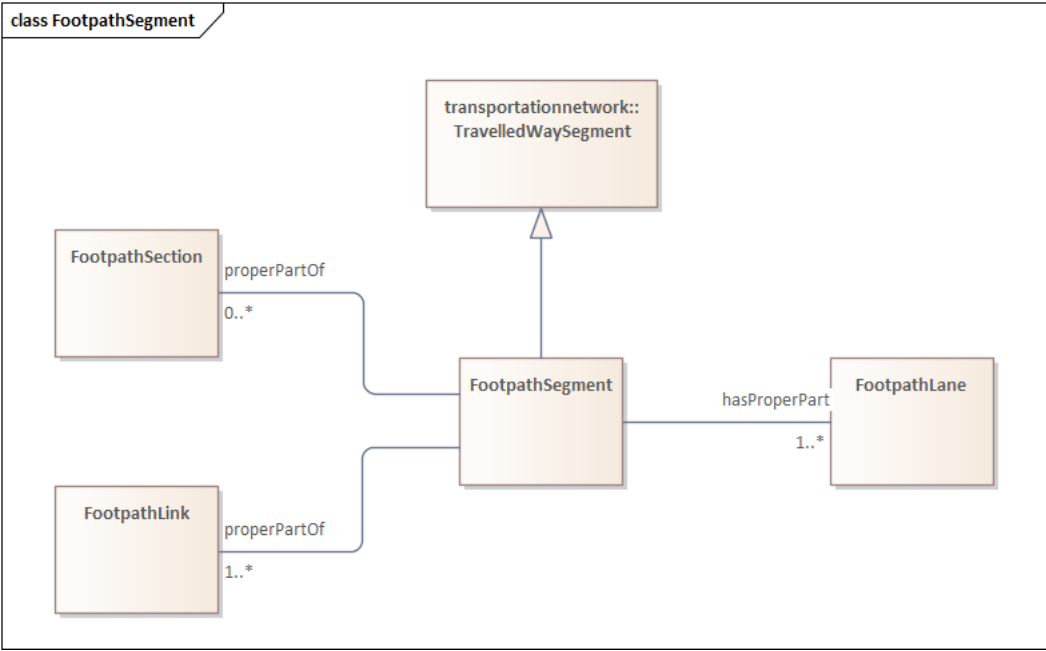


Figure 31—Footpath segment model

6.6.2.6 Footpath lane

A FootpathLane is a type of TravelledWayLane that forms part of a FootpathSegment. Figure 32 provides an overview of FootpathLane.

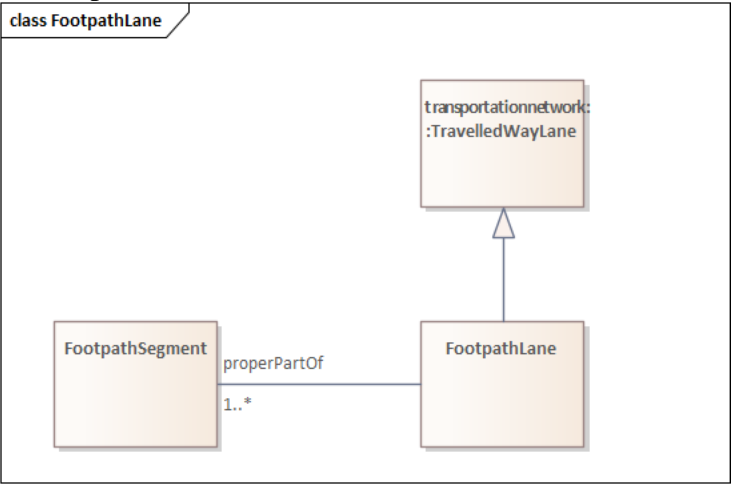


Figure 32—Footpath lane model

6.6.3 Formalization

The key classes and properties are formalized in Table 7.

Table 7—Key classes in the road network pattern

Class	Property	Mult	Value restriction
Footpath	rdfs:subClassOf		transportationnetwork:TravelledWay
Footpath	partwhole:properPartOf	0..*	FootpathNetwork
Footpath	partwhole:hasProperPart	1..*	FootpathLink
FootpathLane	partwhole:properPartOf	1..*	FootpathSegment
FootpathLane	rdfs:subClassOf		transportationnetwork:TravelledWayLane
FootpathLink	partwhole:properPartOf	0..*	Footpath
FootpathLink	rdfs:subClassOf		transportationnetwork:TravelledWayLink

Class	Property	Mult	Value restriction
FootpathLink	partwhole:properPartOf	0..*	FootpathSection
FootpathLink	partwhole:properPartOf	0..*	FootpathNetwork
FootpathLink	partwhole:hasProperPart	0..*	FootpathSegment
FootpathNetwork	partwhole:hasProperPart	0..*	Footpath
FootpathNetwork	partwhole:hasProperPart	1..*	FootpathLink
FootpathNetwork	rdfs:subClassOf		transportationnetwork:TransportNetwork
FootpathNetwork	partwhole:hasProperPart	0..*	FootpathSection
FootpathSection	partwhole:hasProperPart	0..*	FootpathLink
FootpathSection	partwhole:properPartOf	0..*	FootpathNetwork
FootpathSection	rdfs:subClassOf		transportationnetwork:TravelledWaySection
FootpathSection	partwhole:hasProperPart	0..*	FootpathSegment
FootpathSegment	partwhole:hasProperPart	1..*	FootpathLane
FootpathSegment	partwhole:properPartOf	1..*	FootpathLink
FootpathSegment	partwhole:properPartOf	0..*	FootpathSection
FootpathSegment	rdfs:subClassOf		transportationnetwork:TravelledWaySegment

6.7 Public transport network pattern

6.7.1 Overview

The public transport network pattern specializes the transportation network pattern to address the needs for public transport travel. Whereas the other specializations of the transportation network focus on infrastructure, the public transport network pattern focuses on the operations that rely on the infrastructure provided by the other networks. The public transport network pattern is summarized in Figure 33.

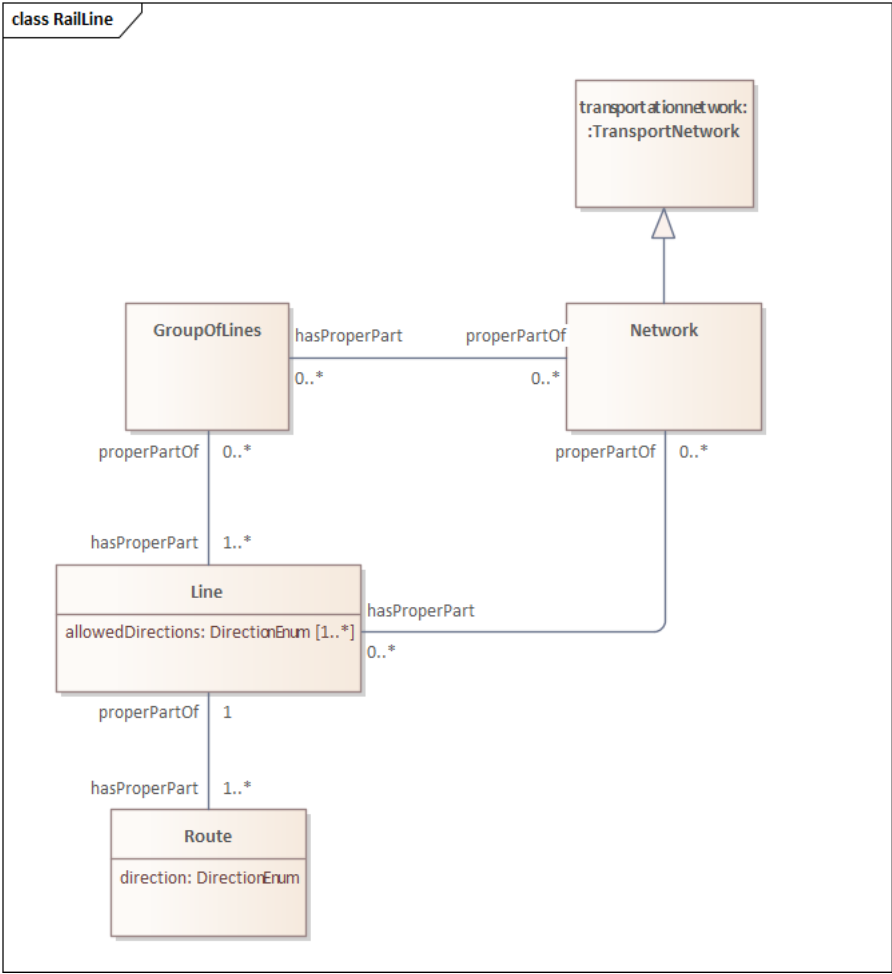


Figure 33—Public transport

6.7.2 Key classes and properties

6.7.2.1 Public transport network

A publictrans:Network is a type of TransportNetwork that provides transport services to members of the public.

A publictrans:Network is made up of Lines and GroupOfLines.

6.7.2.2 Public transport line

A publictrans:Line is one or more routes used by public transport vehicles to transport passengers to and from designated locations. A Line can be used to form Networks and Groups of Lines. A Line can be characterized by its allowed directions of operation.

6.7.2.3 Public transport group of lines

A GroupOfLines is a logical grouping of Lines for any useful purpose.

6.7.2.4 Public transport route

A Route represents one specific path used by a public transport vehicle to transport passengers to and from designated locations. For example, a Line might use an alternate Route to reach its next stop during periods of congestion.

6.7.3 Formalization

The key classes and properties are formalized in Table 8.

Table 8—Key classes in the road network pattern

Class	Property	Mult	Value restriction
GroupOfLines	partwhole:properPartOf	0..*	Network
GroupOfLines	partwhole:hasProperPart	1..*	Line
Line	partwhole:properPartOf	0..*	Network
Line	partwhole:properPartOf	0..*	GroupOfLines
Line	partwhole:hasProperPart	1..*	Route
Line	allowedDirections	1..*	DirectionEnum
Network	rdfs:subClassOf		transportationnetwork:TransportNetwork
Network	partwhole:hasProperPart	0..*	Line
Network	partwhole:hasProperPart	0..*	GroupOfLines
Route	partwhole:properPartOf	1	Line
Route	direction	1..1	DirectionEnum

6.8 Travel corridor pattern

6.8.1 Overview

The travel corridor pattern specializes the transportation network pattern to address the needs for a more holistic travel view.

6.8.2 Key classes and properties

6.8.2.1 Travel corridor

A TravelCorridor is a type of TravelledWay that is made up of TravelCorridorLinks. The extent of a TravelCorridor is defined by the extent of the path that shares the designator assigned to the TravelCorridor. Figure 34 provides an overview of TravelCorridor.

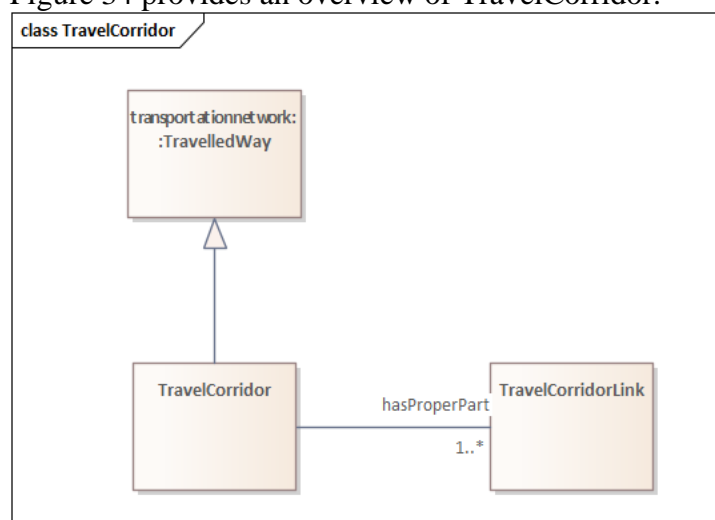


Figure 34—Travel corridor

6.8.2.2 Travel corridor link

A TravelCorridorLink is a type of TravelledWayLink that is made up of TravelCorridorSegments. TravelCorridorLinks can be grouped to form TravelCorridors (i.e., a sequence of TravelCorridorLinks that share a common designator). Figure 35 provides an overview of TravelCorridorLink.

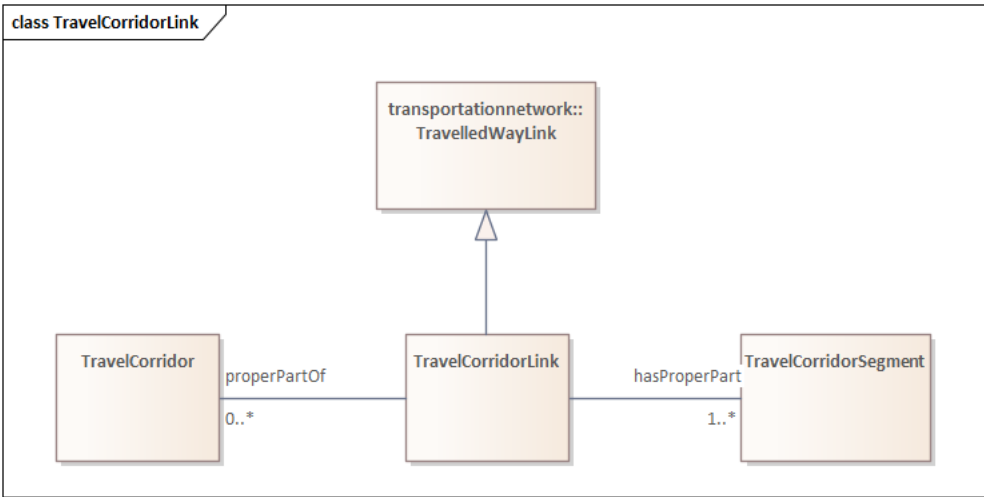


Figure 35—Travel corridor link

6.8.2.3 Travel corridor segment

A TravelCorridorSegment is a type of TravelledWaySegment that logically groups multiple TravelledWaySegments together as being co-located or side-by-side. TravelCorridorSegments can be grouped together to form TravelCorridorLinks. Figure 36 provides an overview of TravelCorridorSegment.

EXAMPLE 1 A TravelCorridorSegment can include a segment containing express lanes and a segment containing local lanes

EXAMPLE 2 A TravelCorridorSegment can include rail, pedestrian, micromobility, and motor vehicle segments into a single corridor segment

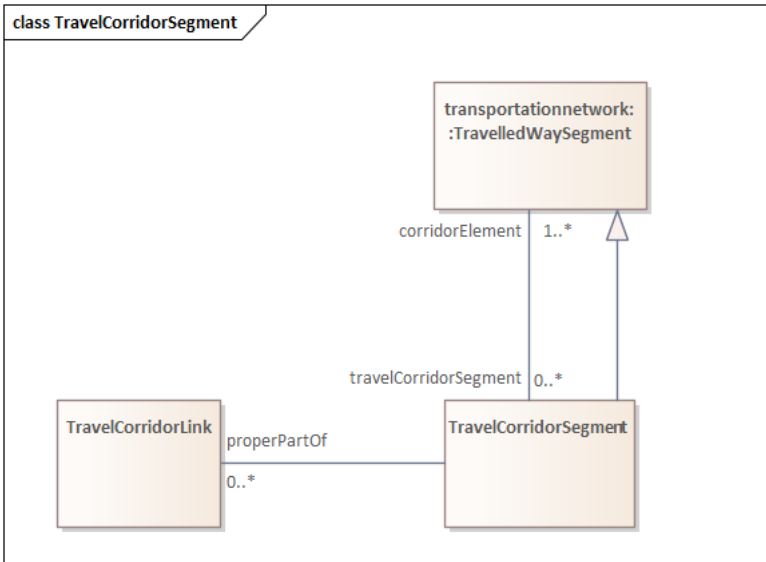


Figure 36—Travel corridor segment

6.8.3 Formalization

The key classes and properties are formalized in Table 9.

Table 9—Key classes in the road network pattern

Class	Property	Mult	Value restriction
TravelCorridor	partwhole:hasProperPart	1..*	TravelCorridorLink
TravelCorridor	rdfs:subClassOf		transportationnetwork:TravelledWay
TravelCorridorLink	partwhole:hasProperPart	1..*	TravelCorridorSegment
TravelCorridorLink	rdfs:subClassOf		transportationnetwork:TravelledWayLink

Class	Property	Mult	Value restriction
TravelCorridorLink	partwhole:properPartOf	0..*	TravelCorridor
TravelCorridorSegment	corridorElement	1..*	transportationnetwork:TravelledWaySegment
TravelCorridorSegment	rdfs:subClassOf		transportationnetwork:TravelledWaySegment
TravelCorridorSegment	partwhole:properPartOf	0..*	TravelCorridorLink

6.9 Transport alert pattern

6.9.1 Overview

The transport alert pattern addresses the need for public agencies to alert groups to unusual transport-related conditions. Most of the features of the transport alert pattern can also be used for other types of alerts (e.g., meteorological events) and as such, its design includes the definition of a generalized alert. The transport alert pattern is summarized in Figure 37.

NOTE It is expected that the definition of the generalized Alert will migrate to ISO/IEC 5087-1 or ISO/IEC 5087-2 at some point in the future.

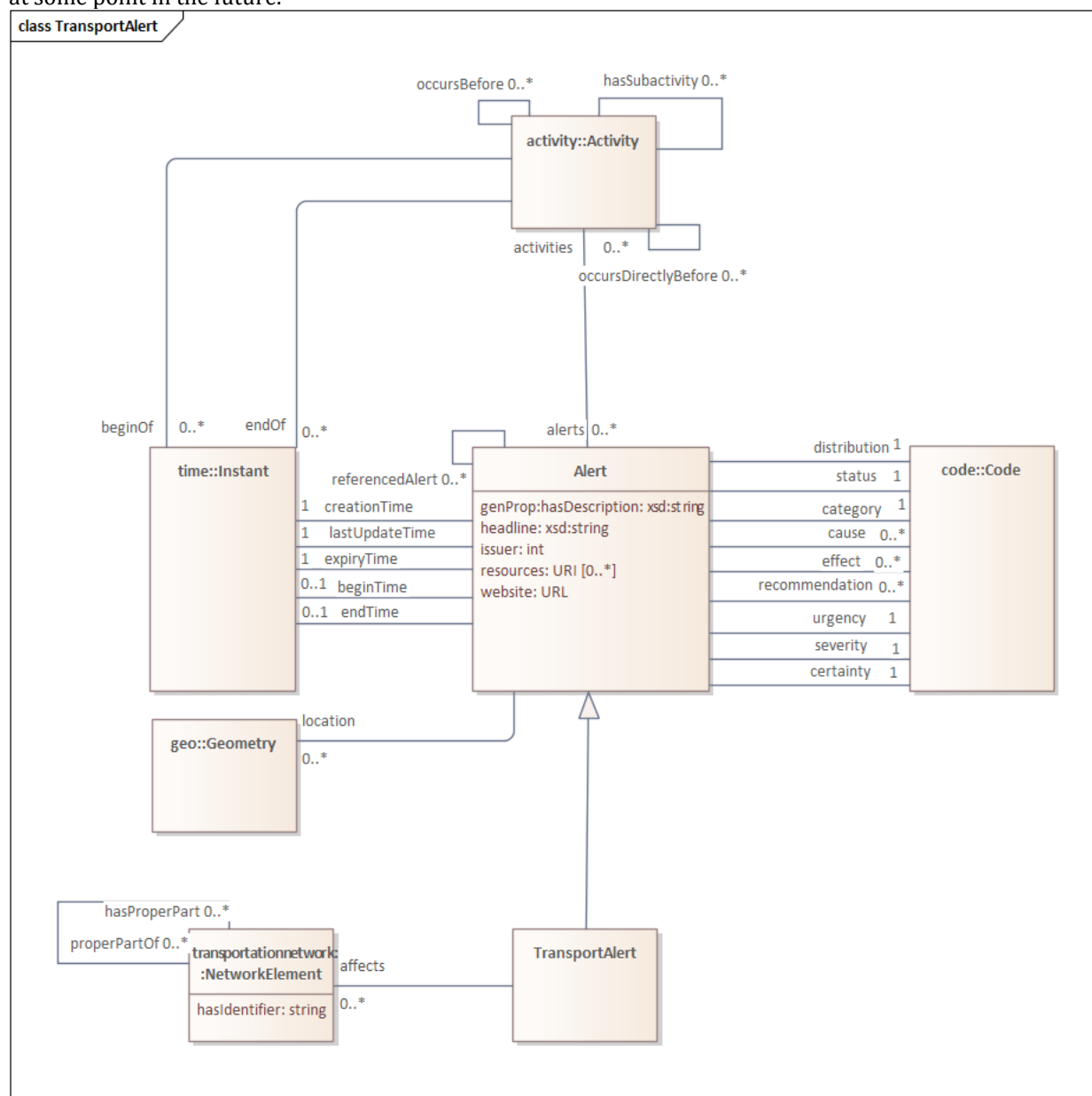


Figure 37—Transport alert

6.9.2 Key classes and properties

6.9.2.1 Alert

A transalert:Alert can be used to notify people of important information. An alert includes the following properties:

- **Headline:** A short textual caption of the alert;
- **Description:** A concise textual explanation providing all major details of the alert;
- **Website:** An optional URI that identifies a website that can provide additional information about the alert;
- **Resources:** Zero or more additional files that are relevant to the alert (e.g., image files);
- **Issuer:** The organization that issued the alert;
- **Creation time:** The time at which the alert was originally created;
- **Last update time:** The time at which the alert was last updated;
- **Expiration time:** The time at which the alert will expire unless it is updated;
- **Begin time:** The time that the alert began or is expected to begin; if no begin time is specified, it is assumed to be the creation time. For example, an alert might be created in advance of major construction work with a begin time of when the construction is expected to begin.
- **End time:** The time that the alert is expected to end; if no end time is specified, it is assumed to be indeterminate.
- **Distribution:** A code indicating the groups who are authorized to receive the alert.
- **Status:** A code indicating whether the alert is real, draft, a test, part of an exercise, etc.
- **Category:** A code indicating the domain that the code relates. Transport alerts should always indicate the transport domain.
- **Cause:** A code indicating the cause of the alert (e.g., construction, collision, weather).
- **Effect:** A code indicating the effect of the alert (e.g., congestion, delays).
- **Recommendation:** A code providing advice on actions to take (e.g., evacuate, shelter in-place).
- **Urgency:** A code indicating how quickly to respond (e.g., immediately, soon)
- **Severity:** A code indicating the expected severity of the condition (e.g., life threatening, injury potential)
- **Certainty:** The likelihood of the condition occurring
- **Activities:** Activities that caused the alert (e.g., a link to construction information)
- **Referenced Alert:** Other alerts associated with this alert.
- **Location:** A geographic area associated with the alert.

6.9.2.2 Transport alert

A transalert:TransportAlert is a type of alert that can be used to notify people of important transport information. A transport alert includes the following properties:

- **Affects:** The network elements of the transport network that have been, are, or expected to be affected by the condition that the alert describes;

6.9.3 Formalization

The key classes and properties are formalized in Table 10.

Table 10—Key classes in the transport alert

Class	Property	Mult	Value restriction
Alert	activities	0..*	activity:Activity
Alert	beginTime	0..1	time:Instant
Alert	category	1	code:Code
Alert	cause	0..1	code:Code
Alert	certainty	1	code:Code

Alert	creationTime	1	time:Instant
Alert	distribution	1	code:Code
Alert	effect	0..*	code:Code
Alert	endTime	0..1	time:Instant
Alert	expiryTime	1	time:Instant
Alert	genProp:hasDescription	1	xsd:string
Alert	headline	1	xsd:string
Alert	issuer	1	Object identifier
Alert	lastUpdateTime	1	time:Instant
Alert	location	0..*	geo:Geometry
Alert	recommendation	0..*	code:Code
Alert	referencedAlert	0..*	Alert
Alert	resources	0..*	URI
Alert	severity	1	code:Code
Alert	status	1	code:Code
Alert	urgency	1	code:Code
Alert	website	1	URL
TransportAlert affects		0..*	transportationnetwork:NetworkElement
TransportAlert rdfs:subClassOf			Alert

Annex A (informative)

Understanding UML diagrams

A.1 General

This document depicts ontological models using the Ontology Definition Metamodel (ODM), v1.1 [6], which is a profile of the Unified Modelling Language (UML), version 2.4.1 [5]. Although UML was originally developed as an object-oriented tool to assist in analysis, design, and implementation of software-based systems, the resultant modelling artifacts have many similarities with ontology artifacts. This annex provides an overview of the key modelling conventions used in this document along with a comparison of how ontological models parallel and contrast with UML concepts when used for object-oriented models.

A.2 Classes

The concept of a class is virtually identical in both ontologies and UML. A class is a representation of a set of individuals (objects, in UML), each of which can be described by a set of common characteristics. In ODM, classes are represented as rectangles with the name of the class near the top of the rectangle in boldface. For example, Figure 38 depicts the class for "Dog". Within this document, classes are always assigned a name starting with an uppercase letter and contain no spaces. Names with multiple words start subsequent words with an uppercase letter. This convention is sometimes called "UpperCamelCase".

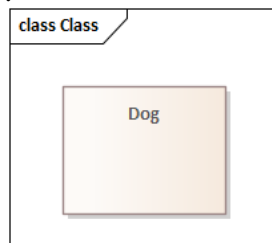


Figure 38—ODM class

A.3 Subclasses

A.3.1 General

A subclass (specialization) relationship indicates that one class is a specialized type of another, generalized class. For example, a Dog is a specialization of a Mammal. The specialized class inherits all of the properties and characterizations of the generalized class. In ODM, the "subclass of" relationship is depicted with a line from the specialized class to the generalized class with a closed arrowhead, as shown in Figure 39.

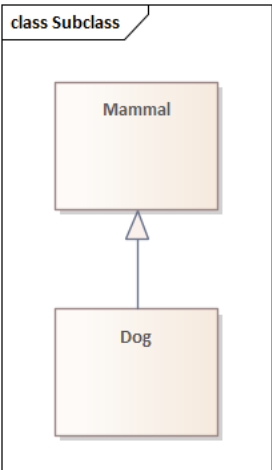


Figure 39—ODM subclass relationship

A.3.2 Subclass Hierarchy

Models can have multiple levels of subclassing. For example, a Dog can be defined as a subclass of Mammal, which can be defined as a subclass of Animal, as shown in Figure 40. As an alternative to showing each generalized class in a diagram, the name(s) of the next level of generalized classes can be shown in italics in the upper right corner of a class rectangle as shown in Figure 41.

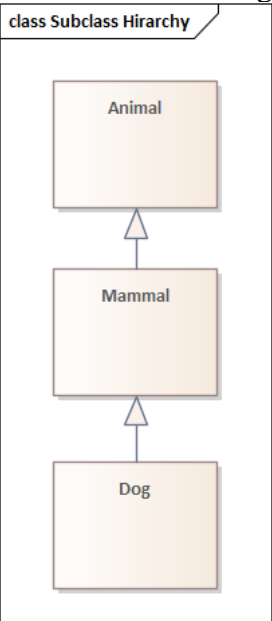


Figure 40—ODM subclass hierarchy

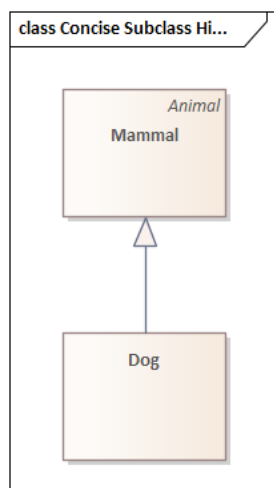


Figure 41—ODM concise subclass hierarchy

A.3.3 Multiple inheritance

Classes can be defined to inherit from multiple other classes. In object-oriented programming this is generally discouraged due to coding problems and ambiguities that can arise; however, the open specification nature of ontologies encourage multiple inheritance because it often is a useful way capture relationships.

And of course, the purpose of specializations is to allow for the creation of multiple specialized subclasses. These concepts are depicted in Figure 42.

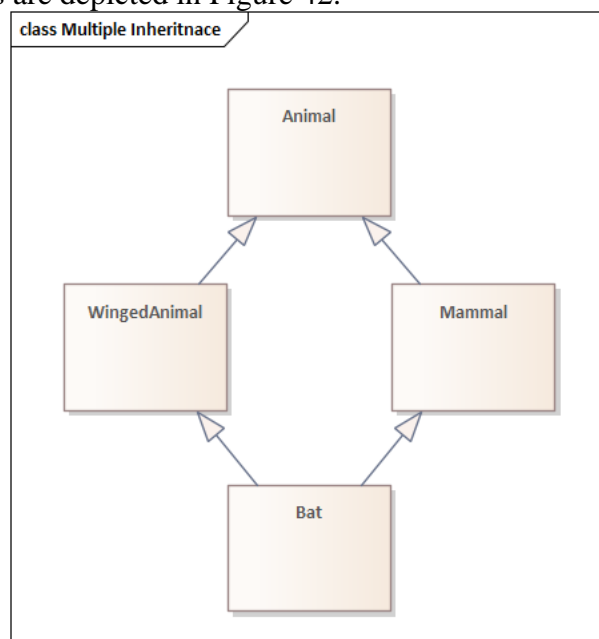


Figure 42—ODM multiple inheritance

A.4 Properties

In both UML and ontological terms, the characteristics of individuals (objects) are called "properties". When a property is represented by a simple value (e.g., an integer or string), it is called "data property" in an ontology (or an "attribute" within UML). When a property is represented by a complex value (e.g., a structure containing multiple fields), it is called an "object property" in an ontology (or an association in UML). Within this document, properties are always assigned a name starting with a lowercase letter and contain no spaces. Names with multiple words start subsequent words with an uppercase letter. This convention is sometimes called "lowerCamelCase".

The way that ontologies and UML treat properties are essentially equivalent with one caveat. Within ontologies, properties are defined independently of classes and can be applied to multiple classes; within UML, properties only exist within the context of a specific class. For example, an ontology can define a property called "location" that indicates the current geographic coordinates of an entity and associate the property with multiple classes (e.g., Dog, Cat, Car) without having to redefine its meaning. If a designer wanted to define a similar property with a different definition or format (e.g., the location of a data file on a computer system), the designer would either need to assign a new name for the new property or create the property in a different namespace. By contrast, within UML, properties are always defined within the context of a class. Thus, within UML, the "location" of a Dog is technically a separate property than the "location" of a Cat, even if they have identical designs. The name of the property is located on the far side of the association line. Because ontologies define properties independently of classes, ODM allows all properties to be represented as their own class with a line drawn between the classes to show their association (Figure 43). This accurately models the capabilities of an ontology (i.e., the property is a separate entity that can be reused) but requires a significant amount of graphical space to depict and can quickly create complex diagrams when there are lots of properties. As an alternative, ODM allows all properties to be represented as names listed in a separate container within the class rectangle, underneath the class name (Figure 44).

A.4.1 Object properties

Within this document, object properties are typically shown using the association method, as shown in Figure 43. This clearly designates the other object as a class and conveys the concept that it can be a complex entity. However, the attribute method is also allowed.

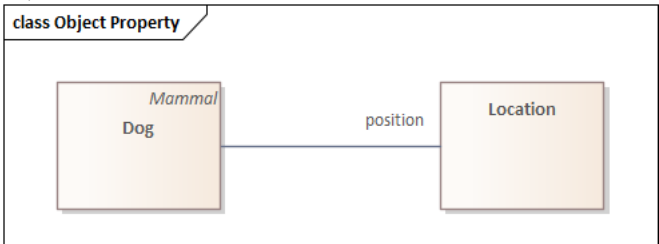


Figure 43—ODM property as an association

A.4.2 Data properties

Within this document, data properties are typically shown using the attribute method, as shown in Figure 44; however, the association method is also allowed. In this example, an Animal is defined to have a weight. This property is inherited by all subclasses of Animal (e.g., Dog).

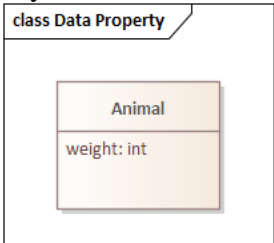


Figure 44—ODM property as an attribute

A.4.3 Multiplicity

A class can be defined to have multiple instances of the same property or can be defined to have optional properties. For example, a Dog can be defined to have two eyes and zero or more owners, as shown in Figure 45.

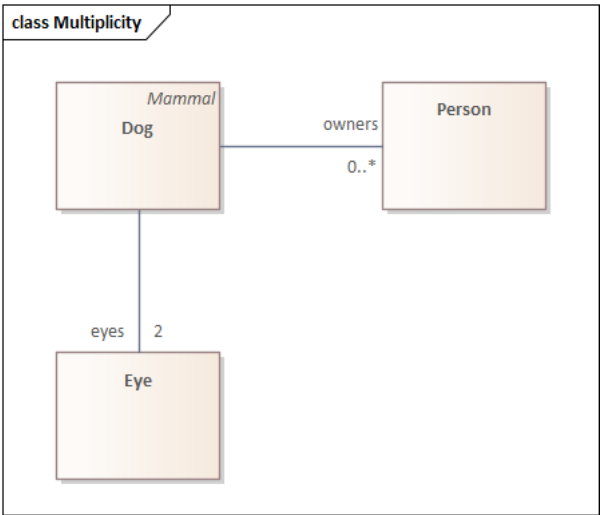


Figure 45—ODM multiplicity

A.4.4 Inverse properties

An object property (association) relationship indicates that one class can be viewed as a property of a second class. In some cases, the object property (association) relationship is bi-directional (i.e., the second class can be viewed as a property of the first class as well). An ontology captures this relationship by declaring that the two properties have an inverse relationship with each other. For example, a "Dog" class can defined to have a property called "wearing", which is associated with the "Collar" class, where the dog can wear zero or more collars. Likewise, the "Collar" class can be defined to have a property called "on", which is associated with the "Dog" class, where a collar can only be on one dog at a time. In reality, there is a single association, but the term used to describe the association is different depending on the direction that one views the association. By comparison, a "Dog" can be defined to have another property called "carrying", which indicates what the dog might be carrying with its mouth, which could be a Collar. However, in this case, the Collar is not considered to be on the Dog. These relationships are shown in Figure 46.

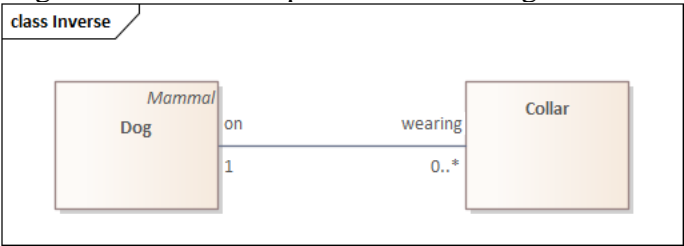


Figure 46—ODM inverse properties

A.5 Namespaces

In cases, classes and properties are imported from other namespaces. These can be shown within the model using the UML notation of the namespace name followed by a double colon followed by the entity name, as shown in Figure 47. Within this document, the namespaces are typically only included in class names. To save space, the diagrams typically do not show the namespace of property names unless there is a significant chance of ambiguity. Namespace names are always included in the formalization tables.

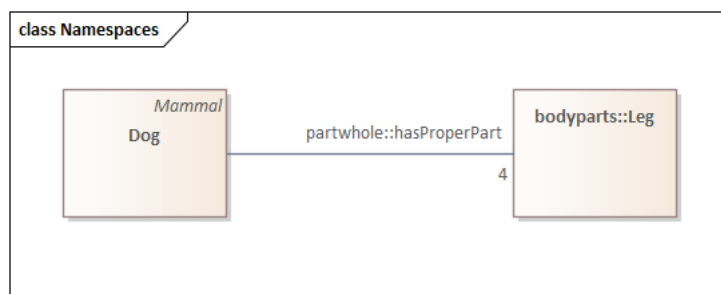


Figure 47—ODM namespaces

A.6 Diagrams versus model

Ontology patterns can become quite complex and difficult to understand if shown in a single diagram. To make diagrams more useful, ODM diagrams are allowed to depict a subset of information within the larger ontological model. Within this document, most of the ODM diagrams are presented from the point of view of a subject object and depicts the properties associated with that object along with direct generalized classes and direct specialized classes. Other information, including the names of inverse properties are omitted from these diagrams to improve the readability of the diagram. The complete normative definition of the pattern can be found in the formalization tables associated with the pattern, which may include details that are not depicted any diagrams.

Annex B
(informative)

Example use cases

Annex C (informative)

Relationship to existing standards

C.1 TC204 relevant standards

C.2 OpenDrive.org

<http://www.opendrive.org/project.html>

C.3 CityGML

C.3.1 Scope

CityGML is an XML-based standard for representing 3D city models. Target application areas identified include: “urban and landscape planning; architectural design; tourist and leisure activities; 3D cadastres; environmental simulations; mobile telecommunications; disaster management; homeland security; vehicle and pedestrian navigation; training simulators and mobile robotics.” It is intended to capture the data necessary to generate 3D portrayals in appropriate tools, providing not only geometry but data regarding surface characteristics and objects of interest (e.g. buildings, water bodies).

C.3.2 Relevance

C.3.3 Data Mappings

C.4 INSPIRE

C.4.1 Scope

The INSPIRE directive is aimed at supporting the sharing of and access to spatial data throughout the EU, particularly those that may have an impact on the environment. INSPIRE aims to create an infrastructure to achieve this, part of which includes the specification of data models in UML. These specifications are defined according to 34 data themes, ranging from Addresses, to Geology, to Human Health and Safety.

C.4.2 Relevance

C.4.3 Data Mappings

C.5 37166: Smart community infrastructures—Urban data integration framework for smart city planning (SCP)

C.5.1 Scope

37166 “focuses on the integration and application of heterogeneous data from urban infrastructure systems, e.g. water, transport, energy and waste etc., so as to support smart city planning. It builds a data framework that involves possible multi-source common data through standardized data integration and sharing mechanism.”

C.5.2 Relevance

37166 is a data framework that involves possible multi-source common data through standardized data integration and sharing mechanism. Although it refers to different categories of data, such as smart grid, transportation, and environment, it does not provide any explicit data models, hence of no relevance to 5087.

Annex D (informative)

Location of pattern implementations

The patterns defined in this document are implemented as OWL files, available online at the following locations:

Transportation Network Pattern:

Bibliography

- [1] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge acquisition*, vol. 5, no. 2, pp. 199-220, 1993.
- [2] M. Grüninger and M. S. Fox, "Methodology for the design and evaluation of ontologies," 1995.
- [3] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific american*, vol. 284, no. 5, pp. 28-37, 2001.
- [4] S. Cox, A. Cuthbert, R. Lake, and R. Martell, "Geography markup language (GML) 2.0," URL: <http://www.opengis.net/gml/01-029/GML2.html>, 2001.
- [5] Unified Modeling Language: Superstructure, Version 2.4.1. OMG Specification, Available at <https://www.omg.org/spec/UML/2.4.1>
- [6] Ontology Definition Metamodel, Version 1.1. OMG Specification, Available at <https://www.omg.org/spec/ODM/1.1>