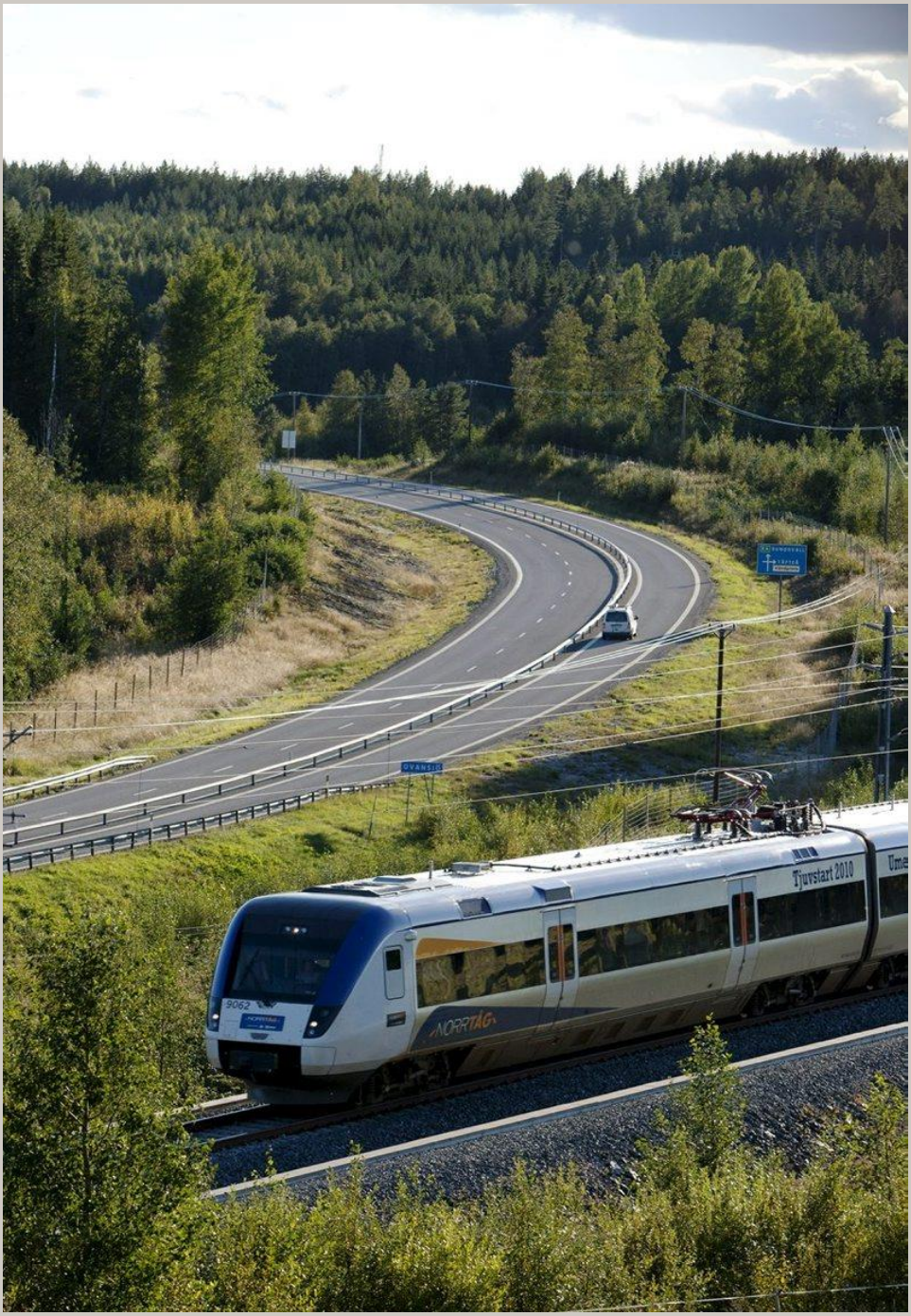
TRAFIKVERKET

REPORT

Information Model Transport Network - Road and Railway



Document title: Information model transport network – road and railway
Created by: Project New Information Models (Projekt Nya informationsmodeller)
Translated to English by: Marieann Sjögren, Rejlers AB, Lars Wikström Triona, Anna Johansson Jacques, Trafikverket
Document date: 2013-06-18
Document type: Report
Document ID: -
Case number: TRV2012/46563
Project number: -
Version: 1.0

Publishing date: 2012-10-30
Publisher: Trafikverket
Contact person: Anna Johansson Jacques
Assignment Manager: Anna Gabrielsson
Print:

Innehåll

1 Background and Purpose

The document describes a proposal for a new information model for the transport network to support the business of the Swedish Transport Administration (Trafikverket).

An information model describes the information required to describe a particular domain of interest for the stakeholders of that domain. The information model is also sometimes called a logical data model. Pertaining to an information system, the information model forms the basis of for example exchange models, components and database models as shown in the figure below.

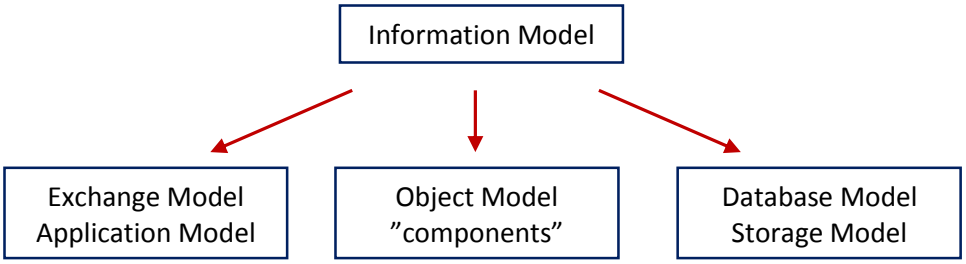


Figure 1 – Information model

The document does not describe the exchange model, components or database model but will be used as requirements definition, when such models should be described.

The information model that is described in this document is primarily used to manage the Swedish Transport Administration's internal needs. It is part of the Swedish Transport Administration’s common information architecture and thus is not a replacement for the existing exchange formats and agreements relating to the Swedish Transport Administration's data exchange with external parties. These exchange formats and agreements have instead been seen as requirements for the information model and which will continue to be met as long as the agreements are current.

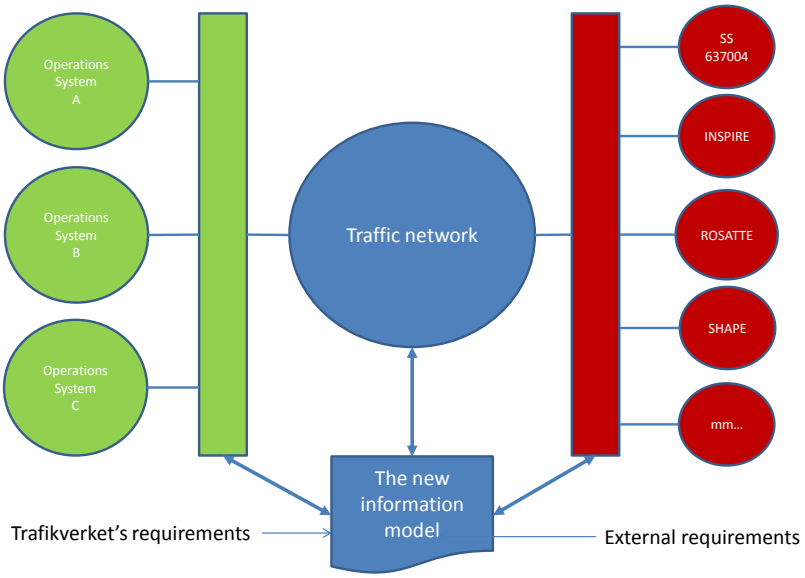


Figure 2 – The Information model in context

The Swedish Transport Administration's own requirements for the new data model is documented separately and are not included in this document. Regarding the external requirements, a mapping between this information model and corresponding definitions in (SIS/TK320 - SS 637004, 2009) and (INSPIRE DS TN) is provided in this document. It is important to note the relationship below between the various sets of requirements.

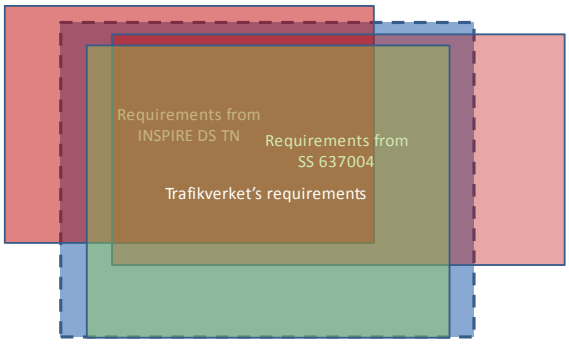


Figure 3 - Relation between the amounts of requirements

The extent of the new information model must meet:

- Mandatory requirements of The Swedish Transport Administration (documented separately).
- Requirements from INSPIRE regarding the road and rail network and intermodal transfer nodes. INSPIRE also set requirements for traffic types like aviation, maritime and cable networks (e.g. cable cars) that do not relate to The Swedish Transport Administration. Note that INSPIRE constitutes legal requirements that The Swedish Transport Administration SHALL meet.
- Requirements from SS 637004 in the parts that are used in today's data exchange with NVDB and also those parts that meet the new requirements from The

Swedish Transport Administration (e.g. intermodal transfer nodes and management of multiple levels of detail for the network).

The above provides an extent of the information model, which roughly is illustrated with the blue dotted rectangle above.

2 Summary

The purpose of the transport network as a reference system is to be able to, where appropriate, connect features (equipment, construction properties, states, events, etc.) to the network and thus provide the location, shape and topological relationships (e.g. borders, intersects, etc.). The features are connected to the reference system by relations with elements in the reference system, i.e. the network. This means that the feature itself need not contain any position or shape data, thus these are obtained from the reference system via the relations.

There are many advantages to using the transport network as a reference system. It makes it easy to find all the features that affect a certain part of the network. Features for which the network location and shape is a good enough representation, do not need to collect these data but rather re-use the geometry data of the reference systems via the relations. Data maintenance (such as improvements in quality of geometry, changes to road/rail designations, classifications etc.) automatically feed through to everyone that connect to the reference system. The reference system contains data that are fundamental to many business areas. For the reference system to serve its purpose, it must also be able to provide a number of services (functions) to support, for example, the following needs:

- Connecting features to the reference system
- Provision of basic data such as geometry (position/shape), track or road type
- Transformations between different reference systems (e.g. between a kilometer post system and the corresponding geometry in SWEREF99 or vice versa)
- Geographic selection (e.g. all network elements within a service area polygon)
- Selection based on, for example, road numbers, track portion, type of road or track type etc.
- Basic mapping services, for example in the form of WMS (web maps)

A service oriented architecture is fundamental because the lack of it usually leads to duplication of data storage, by the fact that several businesses get their own copy, or worse, each gets its own representation of the network and basic properties to be able to perform the above functions within their respective information system. At the same time duplicate software is also often created.

In light of the above, the transport network as a reference system is defined in an information model that contains the following elements:

- Reference network, road and rail. The concept of reference network describes:
    - A network where all the road and rail routes are included. Moreover, it is possible to describe how these road or rail routes are connected, i.e. how

to navigate through the network, and where the different parts of the network are located geographically
    - Networks should in themselves constitute reference systems to which it is possible to connect and reference other features (business data). This should be done in a consistent manner regardless of the type of feature.
    - The networks should be able to cover the different network types available in each mode of transport, e.g. car, bike and walk, rail, tram and metro. *Note: It is an issue of implementation what information the Swedish Transport Administration will actually maintain*.
- Generic model for provision and maintenance of other business data (in this document called features) that are connected to the reference system.
- Service Interfaces, which is included in the overall vision for a future architecture - not specified further in this document, although will be developed in the future implementation of the model. Examples of services are different types of search engines (for all or part of the network), network connection services, transformation services (for mapping between for example different linear referencing systems with other external reference models)

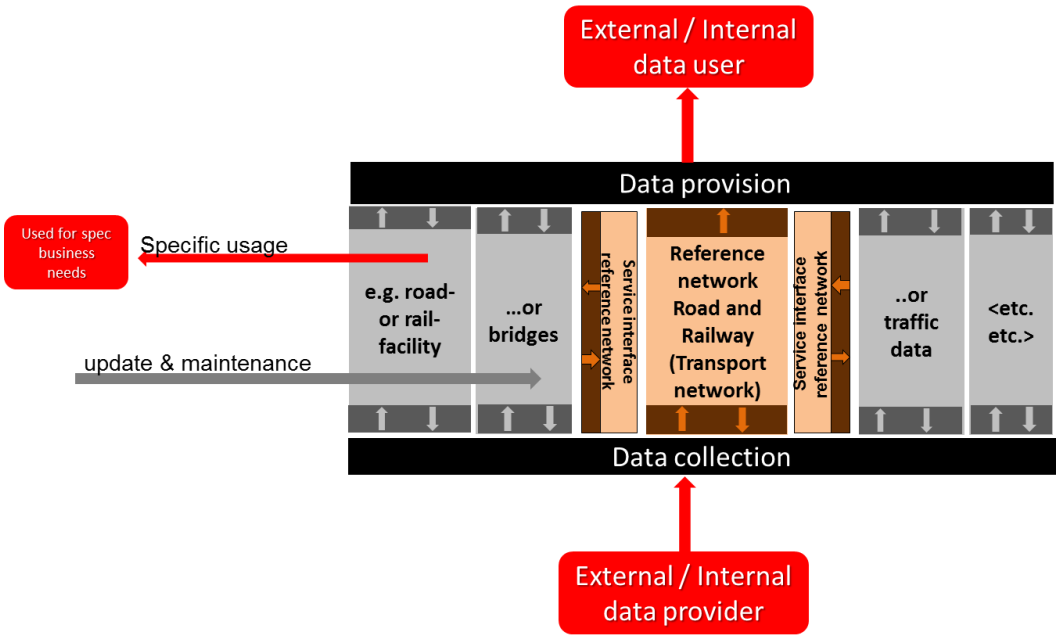


Figure 4 - Overview of the scope of the information model

The figure above shows an overview of the extent of the information model. In the center are the reference network and the basic mechanisms that exist to connect other features to this. Centrally and connected to the reference network there are, besides the reference network, a number of fundamental features that are required to support the various service interfaces which are exemplified above. With the generic feature model there is a possibility to collect and provide data in a uniform way, regardless of what type of feature it is. The mechanisms of the information model for connecting features to the reference network will also be used in cases when specific business applications and data are using the network as a common reference system. This information model does however not include these specific applications and data.

In addition to the above figure, a number of requirements on the topics listed above and some additional topics are made, such as requirements for incremental exchange of data, use of standards, etc.

It is important to understand the relations between the different parts of the model and especially the relationship between the network (road and rail) and the features that are connected to the network. The figure below describes this relationship on a general level:

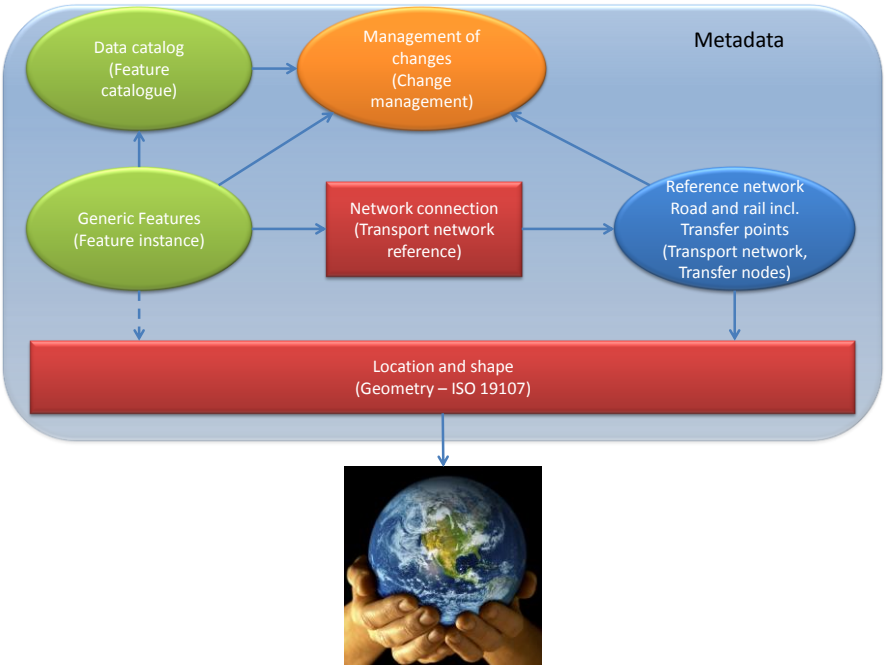


Figure 5 - Relationship between networks, operating data and geography

The reference network describes the actual network where the main properties are:

- Topology which describes how the various elements of the network are interconnected. Hence the topology describes how it is possible to travel through the network.
- Geometry that describes the location (relative to Earth) and the shape of the various elements of the network. Geometry is described in three dimensions.
- The time dimension that describes when in time the various elements of the network exists in reality. This can be seen as a fourth dimension that manages the time axis in addition to the room axes.
- Identity
    - The reference network will in itself be used as a reference system to which other features can be connected. This requires that the elements of the network, to which features are connected, have identities that are stable and are reliable over time.

Generic features can really be anything that has a logical or physical connection to the network. It may concern:

- Objects such as equipment associated with road or railway, for example, signals, signs, wildlife fences etc.
- Events that have occurred anywhere in the network, such as accidents or actions.
- Properties that characterize the elements of the network, such as road surface or railway switches/turnouts, or a construction as a whole.
- Administrative partitions such as railway line segments, road number or road manager.
- Regulations such as bearing capacity and speed limits.

To connect features with the network, coherent mechanisms are needed so that features, regardless of type, can be found in a uniform manner, for example to be able to regard different features and compare them against each other in relation to parts of the road and rail networks. This is referred to, in the figure above, as **network connection.** Features get, through its network connection, indirectly a connection to the geography as the network itself has geometry, i.e. this is a type of indirect positioning as previously described. Features may also, if necessary, be described by separate geometry to get a direct geographic positioning.

Change management is a topic which particularly affects descriptions of changes of data, i.e. network connected features and reference network, in terms of what changes occurred, when, why and by whom. The purpose of change management is primarily to provide traceability of data and also provide a basis for the exchange of incremental updates of the data. Exchanging data according to (SIS/TK320 - SS 637004, 2009) requires this functionality.

Metadata provides information on all data giving the internal and external users, without having to look at the data itself, the ability to find out what data exist and also to evaluate whether the data can be used in a certain context. Within INSPIRE this usage of metadata is termed **discovery** and **evaluation** and is essential to achieve optimal use of data and services.

3 Reading instructions

Chapter 4 contains definitions of terms and abbreviations that may be required to understand the contents of the rest of the document.

Chapter 5 contains the information model itself. Since the model contains many different components and shall constitute a relatively formal requirements specification, it is relatively extensive. The chapter begins with a brief description of the notation (UML) that is used for the information model. UML is used as a modeling language in SIS/TK 320 (Road and rail information) and SIS/TK 323 (Framework for Geodata) as well as within the ISO standardization, e.g. the areas of geographic information (ISO/TC 211) and ITS (ISO/TC 204). The information model utilizes additionally many parts mainly from ISO/TC 211 (whose standards are called ISO 191xx where ”xx” is enumerated from”01” and upwards).

To give a better overview of the information model, it is divided into a number of packages and Chapter 5 contains a subchapter per package. Knowledge of UML and a basic understanding of information modeling in general is required to understand the contents of these chapters.

The original report is written in Swedish but the names used in the actual information model are in English. The reason for this is compliance to international standards and to have an international interface for future system development and data exchange.

In this report all Swedish is translated to English while the names and terms in English have been left unchanged.

4 Terms and abbreviations

Business related terms and concepts are described in a separate appendix of terms. The following describes additional terms and abbreviations that are important for the understanding of this document.

Term	Description
Object	1. Single occurrence of a class. (SIS/TK323 - SS 637006, 2006) 2. Representation of an event/feature in the real world (SIS/TK323 - SS 637006, 2006) *Note:* *Object is a term with multiple but similar meanings. Often, the meaning is derived from the context. A case where the difference is particularly interesting is when you talk about data that describes the real object or metadata that describes the data object.* *An example of data is the date and time when a road or railway line became operational.* *An example of metadata is the date and time when the data describing the road or railway line was recorded in the database.*
Feature	Something concrete or abstract that exists, has existed or will exist (SIS/TK323 - SS 637006, 2006) *Note:* *In this document, the concept feature is used as a general concept, which refers to both the elements in the road and rail networks and also the information that is connected to the network. The features*

	connected to networks are called network connected features.
Class	Modeling concept in UML that is used to describe types of objects. The class is usually an abstraction of a real kind of feature for which one can describe the attributes, relationships and behavior.
Direct positioning	Positioning using coordinates via a coordinate system provides a direct position relative to the Earth.
Indirect positioning	Positioning via object which in turn has a direct or indirect position relative to the Earth. Examples of indirect positioning can be: - Address - Location code - Network reference, e.g. linear reference *Note:* *This document describes a model for the network references used to connect features to parts of the network.*
Linear reference	A type of indirectly position that according to (ISO/TC211 - ISO 19148) is described by: - A linear elements, where each position is unambiguously identified by a measurement value (e.g. length) - A position along the linear element - A description of the method used for linear referencing
Linear element	One dimensional object that functions as a linear axis along which linear referencing can be made. *Note:* *In the case of road- and rail networks, parts of the road and railway are used as linear elements along which one can do linear references.*

Abbreviation	**Description**

UML	Unified Modeling Language (OMG) http://www.uml.org/
OCL	Object Constraint Language (OMG) http://www.omg.org/spec/OCL/2.0/
OMG	Object Management Group http://www.omg.org/

5 Information Model

5.1 Notation

UML is used as the modeling language (OMG). UML is in itself very general and can be used for many types of modeling. In this information model UML is used according to standard (ISO/TC211 - ISO/TS 19103).

Below is a brief description of used notations.

5.1.1 Class

A class is a description of the characteristics of a set of objects that share attributes, relationships, operations and restrictions. In the UML a class is expressed with the symbol below.

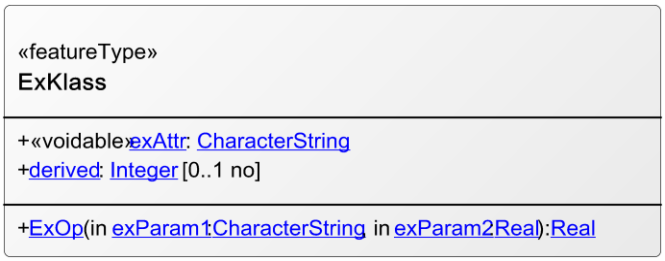


Figure 6 - Class

The class has three partitions that are separated by horizontal lines in the class symbol:

- At the top stereotype and name (see Stereotype below) is described. If the name is in italics it means that the class is abstract, i.e. there cannot be any objects of that type. This may be the case if the class is a base class for other classes (see Inheritance below). Then objects may exist from concrete subclasses.
- In the middle, attributes are described (see Attribute below)
- At the bottom operations are described (see Operations below)

5.1.2 Attribute

Descriptive properties of objects are handled with attributes in classes. The notation for this is:

”visibility <<stereotype>>name [min..max] : type = initial value”

visibility – Visibility of the attribute. In this model, there is only ”+” (normal attribute) or ”/” (derived attributes)

<<stereotype>> - Stereotype for the attribute

name - name of the attribute

[min..max] - Multiplicity of the attribute. This indicates how many occurrences of the attribute that are permitted for an object of the class. This is indicated by a lower (min) and upper limit (max). Lack of multiplicity implies that there should be one and only one occurrence of the attribute for an object of the class.

type - the attribute type (value domain), for example Integer, CharacterString or GM_Point

initial value – A value that the attribute is expected to have initially. If not specified, this attribute is initially undefined.

Example:

+ <<voidable>>length : *Number = 0*

5.1.3 Relations

Relations are stated between classes. There are a number of different types of relations as described below.

5.1.3.1 Inheritance

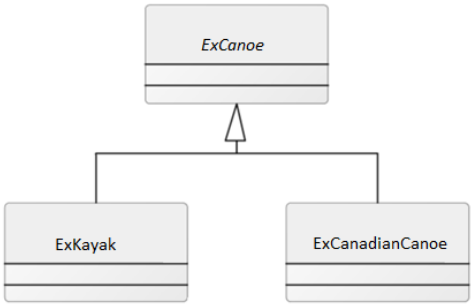


Figure 7 - Example of inheritance

Inheritance is indicated above by a white arrow. Inheritance should be read out as ”is a/an”. In the example above the model is read:

- ExKayak is an ExCanoe
- ExCanadianCanoe is an ExCanoe

ExCanoe is the base class (generalization) and ExKayak and ExCanadianCanoe are subclasses (specialization). The inheritance means in principle two things:

- All properties (including restrictions, attributes, and relationships) are inherited from the base class to subclass.
- Substitutability. Statements regarding the base class also apply to any and each one of its subclasses (but the reverse does not apply). For the above case, this means for example that associations to class ExCanoe in

practice allows associations to objects from an arbitrary subclass of ExCanoe. This follows logically from the semantic meaning of "is a/an".

In the example above, the name *ExCanoe* is written in italics, which means that the class is abstract. This means that that there cannot be occurrences that are only of the type *ExCanoe*. Concrete occurrences must be either *ExKayak* or *ExCanadianCanoe.*

In an information model the inheritance does not have to be exhaustive, i.e. that the quantity of subclasses cover all types of real occurrences. The information model treats the subset of the real world that is considered to be included in the problem domain. The problem domain is defined by its demarcation.

5.1.3.2 Association

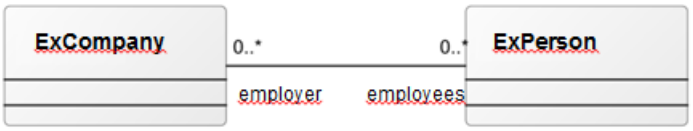


Figure 8 Examples of association

An association is indicated by a line between the classes that are associated. An arrow at either end means that the association is explicitly navigable in that direction, i.e. one can easily find one object of one class from an object of the other class.

The name and multiplicity can be specified to one end of an association, which describes the respective class role in the association. The example above should be read as:

- An occurrence of *ExCompany* may be associated with zero to many occurrences of *ExPerson* who has the role of employees in the association. In the same association *ExCompany* plays the role of employers. An *ExPerson* is allowed to have zero to many employers.

There is certainly information associated with the employment itself (the association) such as employee number, salary, etc. In this case you must define another class for describing the employment and remake the associations from the above case:

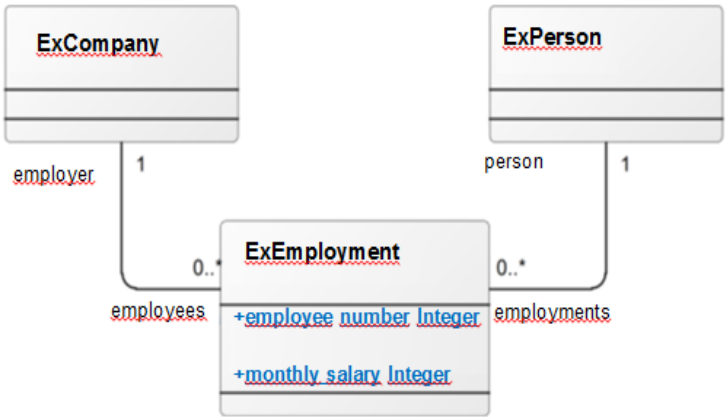


Figure 9 - Association, refined example

The example above can be refined further. The point is to show that from this aspect there is no difference to be modeling with UML compared to alternative notations such as entity-relationship. The models shall be normalized as desired, and the each thing must be described in the right place.

If multiplicity is not specified for association roles it is assumed that [1..1] applies.

It is not always necessary to specify a name for an association role. If it is not interesting to talk about the association in that direction, no name is needed either. However you should usually specify multiplicity at both ends.

5.1.3.3 Aggregation

Expresses a stronger type of association. This type of relationship is not used in this information model as the semantics is not completely clear.

5.1.3.4 Composition

Expresses the strongest type of relationship that is used for cases when you want to describe an entirety that is built up by its parts and where the parts share life span with the entirety.

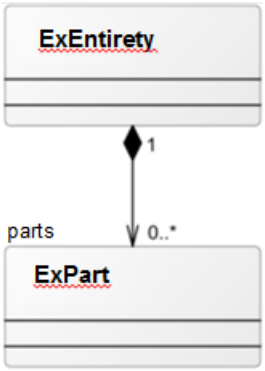


Figure 10 - Example of composition

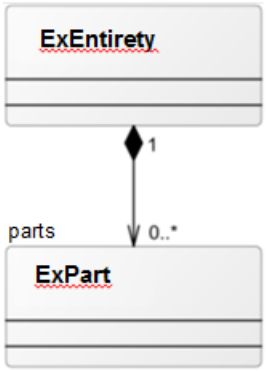
Above a composition between *ExEntirety* and *ExPart* is shown, which means that the entirety consists of or is built up by its parts. If an occurrence of the type *ExEntirety* is removed, all its occurrences of the *ExPart* should also disappear. This is shown by a filled diamond at the "entirety end " of the relation.

Otherwise, the same rules apply for describing the relational roles as for normal association.

Note:

> *In UML, attributes and composition (see below) are equivalent concepts. To describe a class ExEntirety with an attribute "parts" of the type ExPart is analogous with describing a composition between the classes ExEntirety and*

ExPart and name the composition to parts. The figure below is therefore



equivalent to

Figure 10 - above.

Figure 11 - Example of composition as attribute

5.1.4 Operations

Operations describe what the class is expected to be able to perform in terms of functions that can have parameters and return value types. Since this document describes an information model, operations are not so important. However, it may be interesting to model operations for the times when you want to be able to refer to them in connection with the definitions of restrictions or that the operation as such adds semantics and increased understanding, or clarifies requirements. An operation is expressed by the following notation:

"<<stereotype>>name(parameters) : returned_value_type"

<<stereotype>> - Stereotype for the operation

name - name for the operation

parameters – type and name for the parameters to be sent to the function

returned_value_type – type of value that the operation is expected to return

Example:

Length(Boolean useZ) : *Number*

5.1.5 Stereotypes

Stereotypes provide additional semantics to the rest of the modeling elements, and can be seen as a meta-model class. The stereotypes used in this information model follow (ISO/TC211 - ISO/TS 19103), (ISO/TC211 - ISO 19108) and (INSPIRE GCM) according to the table below:

Stereotype	Links to	Description
<<featureType>>	Class	Specifies that the class should be regarded as a feature according to (ISO/TC211 - ISO 19108)
<<type>>	Class	The class is a non-instantiable collection of attributes, relations and operations. The class should be seen as a template (specifying requirements) of concrete and instantiable classes that should be able to occur instead of the class itself in concrete cases.
<<datatype>>	Class	Specifies that the class only describes a data structure. Occurrences of the class are not independent objects, but should only be seen as more complex structures of attribute values.
<<voidable>>	Attribute, Relation role	Specifies that occurrences can omit the data for this attribute or this relation, but is expected to give a description of why the data is omitted. See detail in (INSPIRE GCM).
<<enumeration>>	Class	Specifies that the class is a data type where each occurrence can have a value from an exhaustive enumeration of allowed values in a list. The amounts of allowed values are shown in the class.
<<CodeList>>	Class	Similar to <<enumeration>> with the difference that the enumeration of values should only be seen as examples. A concrete list must be maintained and made available outside of this information model.

5.1.6 *Restrictions*

The OCL language, as described in (OMG), is used for restrictions. Restrictions complements the UML model by specifying ways formally allowed or possible to use the model where it is not clearly defined by the UML model itself.

Understanding of the OCL is not necessary for the conceptual understanding of the information model. However, the OCL-constraints brings further formalization and where applicable, detailing of the requirements set by the model.

OCL-constraints are shown in the document in a text box as follows:

Restriction

5.2 Purpose of the model

The model should cover the sets of requirements for the following areas:

- Reference network
- Network connection
- Data catalog
- Generic model for business data (features)
- Change management

The requirements for these areas are documented separately.

On a general level, the model shall provide a harmonized description in the first place of the road and rail networks out of the same basic model, and also be able to allow for transitions between different transport networks.

5.3 Model for road and rail networks

5.3.1 Overview

The figure below shows the packages that are included in the information model and their interdependencies. The purpose of the package structure is:

- A way to divide the model into a number of logical parts or topics, where each topic is covering a particular area of information and/or purpose
- A way to check the permitted dependencies between the components

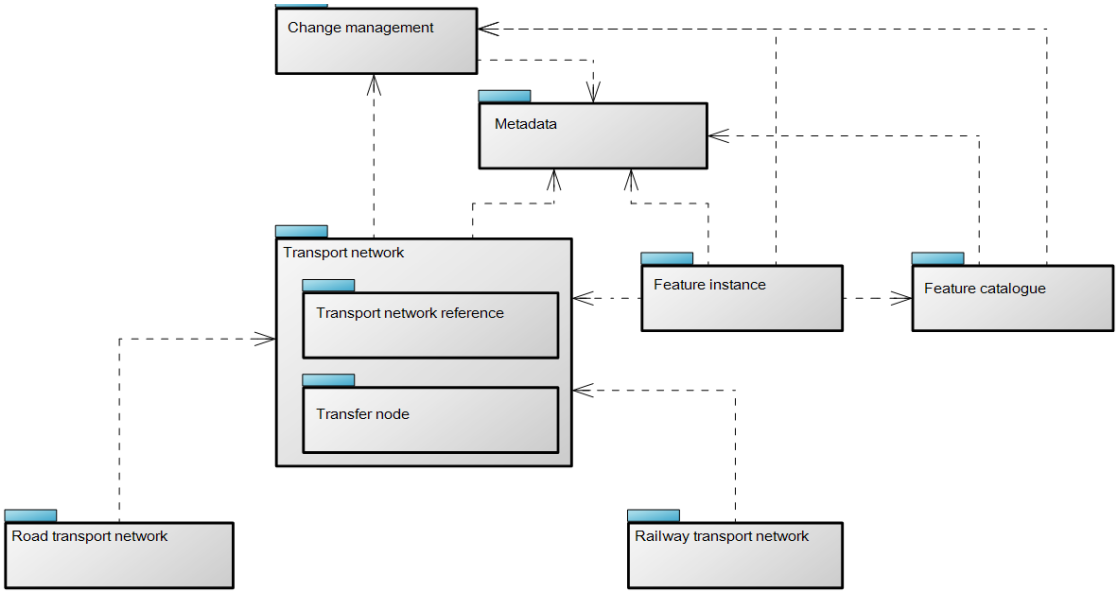


Figure 12 – Package Structure

The specified packages are as follows:

- **Transport network** whose purpose is a model to describe the general aspects of the network. This part also includes transfer nodes whose purpose is to describe transitions between different types of transport
- **Transport network reference** whose purpose is to provide a model for network connection
- **Feature instance** and **Feature catalogue** whose purpose is to describe a generic model for features that can be connected to the network
    - The Package **Feature catalogue** describes the structure of a catalog in which you define the types of features and their property types and relations. Please note that only a small number of feature types are defined in this document. It is the task of stakeholders to specify feature catalogs for their business domains, according to defined structure in this document, when having to exchange the business data with others
    - The Package **Feature instance** describes a generic model for feature instances linked to the types defined in the feature catalog
- **Road transport network** describe specific aspects of road networks based on the general model for networks
- **Railway transport network** describe specific aspects of railway networks based on the general model for networks

The packages above uses, as far as possible, the standards in this area, e.g.:

- SS 637004 (SIS/TK320 - SS 637004, 2009)
- SS 637006 (SIS/TK323 - SS 637006, 2006)

- INSPIRE DS TN (INSPIRE DS TN)
- ISO 19100 (ISO/TC211)
    - ISO 19103 – Conceptual schema language
    - ISO 19107 – Spatial schema
    - ISO 19108 – Temporal schema
    - ISO 19109 – Rules for application schema
    - ISO 19115 – Metadata
    - ISO 19137 – Core profile of the spatial schema
    - ISO 19147 – Transfer nodes
    - ISO 19148 – Linear referencing
    - ISO 19157 – Data quality

In cases where the definitions are retrieved from standards, this is shown by reference to the relevant standard/specification, e.g. (INSPIRE DS TN) to show that the definition is retrieved from the data specification for transport network in INSPIRE.

In addition to the above packages in the information model, there are also more technical packages, whose purpose is to describe data. These packages are:

- **Change management** that models the information needed to describe data updates. This part uses especially SS 637007 (SIS/TK323 - SS 637007, 2006).
- **Metadata** which currently points to the Swedish national metadata profile, describing other interesting properties of data, such as quality declarations.

Each package is illustrated by one or more figures in the form of UML diagrams. The purpose to make multiple diagrams is mainly to make the model easier to read. The same class can occur in several diagrams to connect the different diagrams to each other as a whole.

5.3.2 Package Transport network

5.3.2.1 Overview

The package contains classes for a basic network structure, regardless type of transport, which includes topological (= network graph), geometric and temporal aspects.

The model is based primarily on INSPIRE (INSPIRE DS TN) and the classes that are included in this package are generic and intended to be used regardless of type of transport. In this way, a harmonization between the road and rail networks is achieved. Extensive consideration is also given to Swedish standards (SIS/TK320 - SS 637004, 2009).

The basic principle of the model is to describe the network (*TransportNetwork*) as a set of interconnected linear elements (links/*TransportLink*). The links are interconnected via nodes (*TransportNode*). Geometric lines and points describe the position and shape of the elements in the network. One network describes the network for one type of transport. There can be several networks within the same type of transport. For example, you can choose to separate pedestrian and bicycle networks from the road network, or metro network from railway network even if they belong to the same type of transport.

For both road and rail networks there is a need to describe the network at different levels of detail. To deal with this there are two mechanisms in the model. First and foremost each element in a network describes which level of detail it represents. In addition to this there is an possibility to describe how elements of different levels of detail are interrelated and correspond to each other, for example a link which at one level describes an entire highway, and at another more detailed level corresponds several links representing each carriageway of the highway. The mechanism of this mapping between levels of detail has been retrieved from the Swedish standard (SIS/TK320 - SS 637004, 2009) and is called *TransportSystem*. The corresponding mechanism does not exist in INSPIRE.

To provide the ability to connect features to elements that are as stable as possible, there is also a linear element of higher order available which is called *TransportLinkSequence,* corresponding to the class *NW_RefLink* in Swedish Standard. Advantages of this are described in a separate chapter below. The different possibilities to link features to the network are described in a separate package (*Transport Network Reference*).

There is also a need to collect a coherent set of links, for example, all of the track links in a operation station, or all road links at a complex intersection, to a superior element to which you can connect features. The superior element in the model is called *LinkSet.*

The types of transport road and railway have both been modeled starting from a basic, transport type independent, model to harmonize the basic concepts. Specific parts of the various transport types are described in separate packages (*Road Transport Network* and *Railway Transport Network*).

Finally, since the model describes networks for several transport types, there is a need be able to connect the networks to each other in different ways. In this information model there are three different ways to do this:

- Via so called transfer nodes (*TransferNode*) where different types of types of transport and transport means are linked together. These are described in a separate chapter 5.3.4
- Via *InterNetworkConnection* where different networks within the same transport type are linked so that a transport mean can move between different networks. An example would be a bicycle path network that connects to a car road network so that you can describe how a cyclist first can travel along the bicycle path network and then into the road network.
- Via *InterNetworkConflict* where different networks (both within and between transport types) collide and are in conflict, without the possibility that any transport means can switch networks. An example might be intersections between road and rail networks.

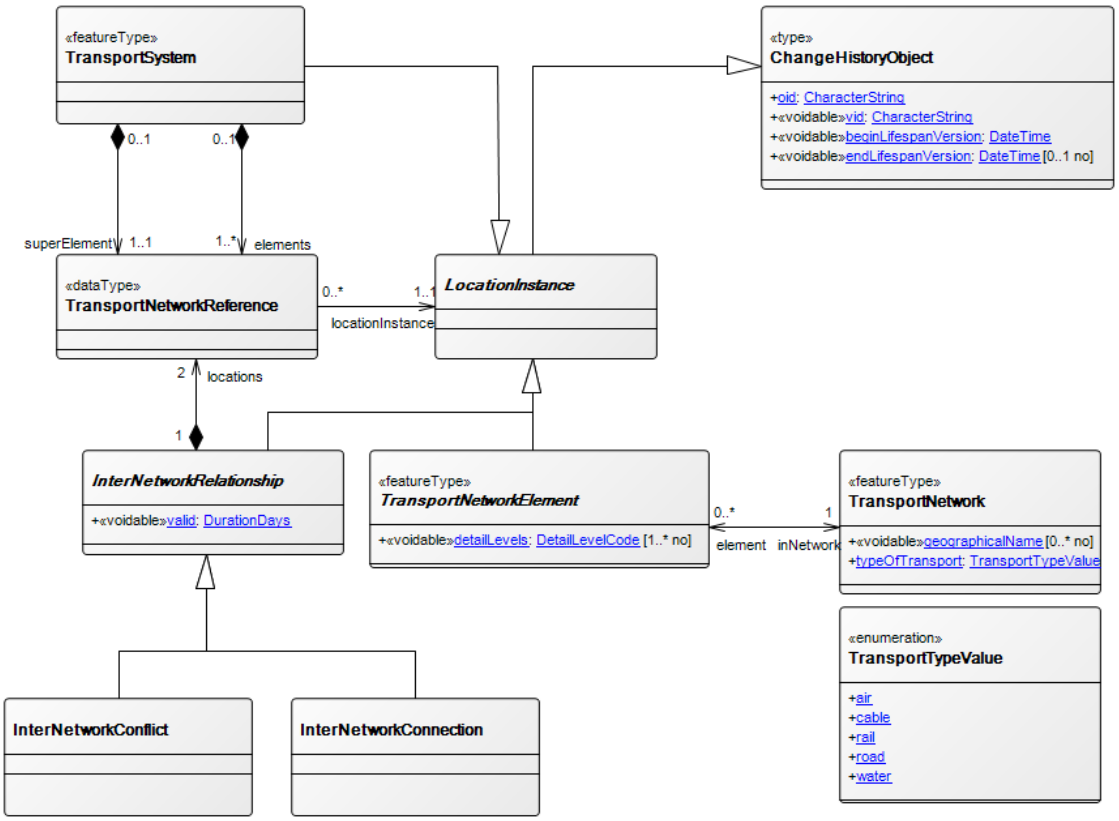


Figure 13 - Overall structure of traffic network

The figure above describes the classes that compose the basic structure of the network. As shown, the class *TransportNetworkElement* is abstract. It constitutes a base class for the different types of network elements that may be present in a network. The figure below shows the different types of network elements that may be present in a network.

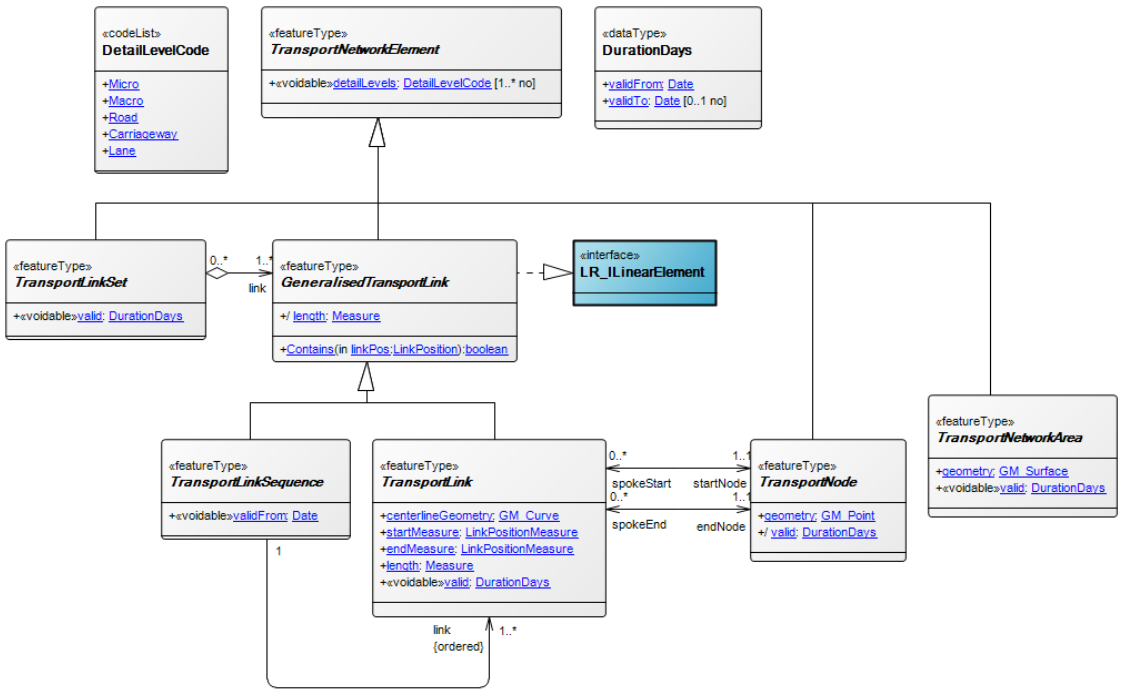


Figure 14 - The different types of network elements

Basically, there are links and nodes that are described by the classes *TransportLink* and *TransportNode.* Nodes primarily describe the places where it is possible to make route choices in a network, for example at junctions or in switches. The nodes also describe the locations where the network ends, such as road or trail ended. Finally, nodes can exist in other places where you, for practical reasons, have chosen to divide the net. These nodes are often called pseudo nodes. To describe how to travel in the network, the nodes are linked together with links where each link connects two nodes. Each node can theoretically have unlimited incoming and outgoing links, although in practice, e.g. in switches, there may be physical limitations.

To link network elements with geography, both links and nodes have geometry. Geometry's task is to describe the position and shape of the various network elements.

An important principle for the network is that it does not describe the business data as attributes directly in the various network element classes. This is primarily for the model to be flexible to handle all types of business data. If you, for example, consider the division of the wearing course type and the speed limit along the road network then these data have only in exceptional cases the same partition. If you want to fill up with another hundred types of data types which all have different partitions, then the network partition will become very fragmented since each element would need to be homogeneous with regards to its attributes. A fragmented network is difficult to keep up to date, analyze and present. In addition, problems arise when you later want to add even more types of data that perhaps do not have the same partition as the previous data already has.

A guideline in this context that we have tried to follow is that:

- The network shall not be affected by changes in business data

- Business data shall not be affected by changes in the network

For this reason, the model is designed so that the network itself has a small number of basic characteristics. This is supplemented by a number of mechanisms that shall be used to connect additional data to the network without the network partition itself being affected.

Many of the classes at this level in the model are abstract (i.e. cannot be instantiated). Concrete subclasses available in packages for road respectively rail networks are described in Chapters 5.3.5 Package – Road transport network and 5.3.6. Package – Railway transport network.

In order to be able to position features (business data) with linear references along linear elements, a number of classes are defined according to the diagram below. The reason that these classes are defined in this package is that the definitions are also needed for the definition of the linear elements in the network. We want to avoid circular dependencies between packages in the model.

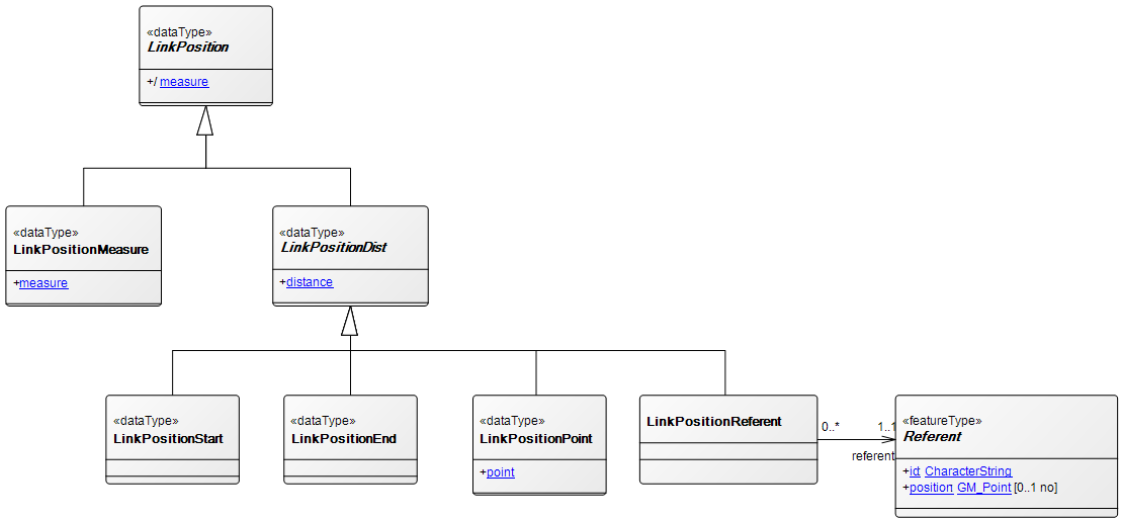


Figure 15 - Positions on linear elements

The following describes the different classes in the network model.

5.3.2.2 Class <<featureType>> TransportNetwork (INSPIRE DS TN)

5.3.2.2.1 Definition

The class represents a network consisting of a set of topologically interconnected network elements (*TransportNetworkElement*) for a type of transport.

Note:

It may for different reasons be multiple networks within the same transport types. You can for example choose to separate pedestrian and bicycle path network from the road network due to practical or organizational reasons.

5.3.2.2.2 Attribute

Name	Type	Description
geographicalName[0..*] (INSPIRE DS TN)	GeographicalName (INSPIRE DS TN)	Geographical name for network
typeOfTransport (INSPIRE DS TN)	TransportTypeValue (INSPIRE DS TN)	The type of transport that the network is serving.

5.3.2.2.3 Relations

Name	Type	Description
element[0..*] (INSPIRE DS TN)	TransportNetworkElement (INSPIRE DS TN)	The set of network elements included within the network.

5.3.2.3 Class LocationInstance (SIS/TK320 - SS 637004, 2009)

An abstract base class for position objects. I.e. the parts of the network that can be used for indirect positioning and constitutes parts to where you can connect features.

LocationInstance is a subclass of *ChangeHistoryObject* which means that it has identity and is subject to change management (see chapter 5.3.10)

5.3.2.4 Class <<featureType>> TransportNetworkElement (INSPIRE DS TN)

An abstract base class that represents an element in a network.

TransportNetworkElement is a subclass of *LocationInstance* which means that each element is a position object to which it is possible to connect features, and that the element is subject to change management.

5.3.2.4.1 Attribute

Name	Type	Description
detailLevels [1..*] (SIS/TK320 - SS 637004, 2009)	DetailLevelCode (SIS/TK320 - SS 637004, 2009)	The levels of detail *(Micro, Macro, Road, Carriageway, Lane – se DetailLevelCode)* that the network element represents. *Note:* *Swedish standard specifies a multiplicity of 0..*. In this model, we have considered that a network element*

		always belongs to at least one level of detail.

5.3.2.4.2 Relations

Name	Type	Description
inNetwork (INSPIRE DS TN)	TransportNetwork	The network that the element is included in.

5.3.2.5 Class InterNetworkRelationship (SIS/TK320 - SS 637004, 2009)

An abstract base class that describes a relationship between elements in different networks. It corresponds to *NW_ValidNetConnector* in (SIS/TK320 - SS 637004, 2009).

5.3.2.5.1 Attribute

Name	Type	Description
<<voidable>>valid (SIS/TK320 - SS 637004, 2009)	DurationDays	The period of time during which the relation is valid.

5.3.2.5.2 Relations

Name	Type	Description
locations [2] (SIS/TK320 - SS 637004, 2009)	TransportNetworkReference	The two network elements (or parts of the network elements) that are related. *Note:* *Swedish standard has multiplicity 2..n, but since the semantics of this becomes complex (which network elements are associated with which) we have here limited ourselves to multiplicity 2.*

5.3.2.6 Class InterNetworkConflict

A type of *InterNetworkRelationship* that describes a conflict between different networks (both within and between transport types), where means of transport cannot change networks.

Example:

Level crossing between the road and rail networks

5.3.2.7 Class InterNetworkConnection

A type of *InterNetworkRelationship* that describes a relationship between network elements in different networks within the same transport type, which allows the transport means to move between networks.

Example:

The coupling between a bicycle network and a road network with a bicycle lane, where a bicycle can move between the networks

Note:

To describe the connections where persons or goods can change between transport types, the model for the transfer nodes shall be used (Chapter 5.3.4).

Restriction:

The network references should reference elements in the same transport types

```
Context InterNetworkConnection inv:

location->at(1).locationInstance.oclAsType(TransportNetworkElement).inNetwork.typeOfTransport =
location->at(2).locationInstance.oclAsType(TransportNetworkElement).inNetwork.typeOfTransport
```

5.3.2.8 Class <<featureType>> TransportLinkSet (INSPIRE DS TN)

A type of network element that is a set of linear network elements (links and/or link sequences), where the set as a whole has a specific function in the network. *LinkSet* is an abstract base class.

Note:

The purpose of LinkSet is to describe sets of linear network elements where the collection in itself should be an identifiable element in the network. There are no restrictions on how the different links and/or link sequences can be grouped together other than each element may only appear once.

Example:

All links that constitutes a junction for road or track links in an operation station for the railway; can be described as a LinkSet. As a result of this, operation station as such, is an element in the network to which it is possible to connect other information. The set of links can vary over time.

Figure 16 - Example of use of LinkSet for road

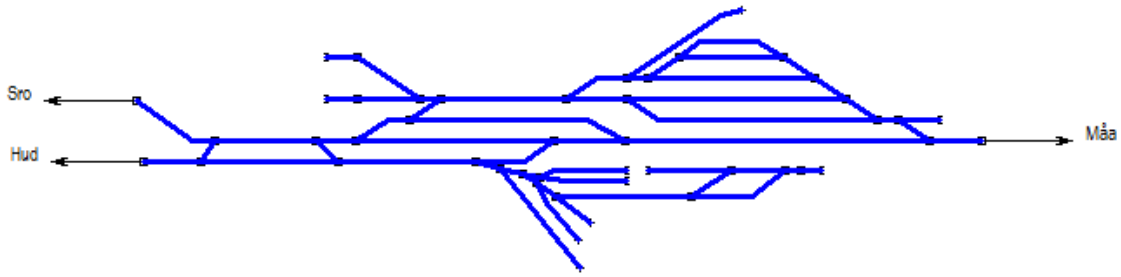


Figure 17 - Example of use of LinkSet for railway

5.3.2.8.1 Attribute

Name	Type	Description
<<voidable>> valid (INSPIRE DS TN)	DurationDays	Validity of the set as a whole. *Note:* *The validity of linear network elements can be allowed to vary. At any one time valid network elements are considered to be included in the set, provided*

		that the set itself is valid. A network element that never is valid at the same time as the set will never be included in the set.

5.3.2.8.2 Relations

Name	Type	Description
link [1..*] (INSPIRE DS TN)	GeneralisedTransportLink	The linear network elements included in the set. *Restriction:* *All included elements should belong to the same network, as the set itself.* Context LinkSet inv: link->forall(l | l.inNetwork = self.inNetwork)

5.3.2.9 Class <<featureType>> GeneralisedTransportLink (INSPIRE DS TN)

A linear network element (realizes interface LR_ILinearElement (ISO/TC211 - ISO 19148)) which can be used as target for linear references and shall therefore meet the requirement to describe a single continuous path in the network. Each position on the element should be identifiable with one and only one parameter, such as length.

GeneralisedTransportLink is an abstract base class.

Note:

This type of network element constitutes a type of linear element in the primary linear referencing system.

The existing options for network connection are described in the package Network reference as subclasses of the class NetworkReference. Other types of (secondary) linear referencing systems are not explicitly described in this document.

5.3.2.9.1 Attribute

Name	Type	Description
/length	Measure (ISO/TC211 - ISO/TS 19103)	Derived attribute that contains the length in meters for the linear element.

5.3.2.9.2 Operations

Name	Return Type	Description
Contains(linkPos : LinkPosition)	Boolean (ISO/TC211 - ISO/TS 19103)	Operation that returns true if the specified position is included in the linear element. Otherwise false is returned. *Note:* *This function is only used in order to specify requirements that the link positions associated with location data on linear elements actually are included in the linear element.*

5.3.2.10 Class <<featureType>> TransportLinkSequence (INSPIRE DS TN)

A type of linear transport network element, that is described by a sequence of transport links (to meet the requirement for linear elements).

TransportLinkSequence is an abstract base class.

Note:

TransportLinkSequence corresponds to NW_RefLink in (SIS/TK320 - SS 637004, 2009).

Note that a link sequence may pass through more than one node. Due to the restrictions that apply to linear network elements, each node shall have a maximum of two connecting links that are included in the sequence. If two links connect to the node, one link should be incoming and the other outgoing. Note that in the case of a loop (for example a turning area on a road), a link may have one and the same node as both the start and end node.

In (INSPIRE DS TN) *there is a class (DirectedLink) in a relation with TransportLinkSequence and its included parts. This provides the possibility that the respective part need not have the same direction as the sequence as a whole. This option has been removed in this model to make it simpler. We believe that it is not a restriction to require that the components must be of the same direction as the whole sequence.*

Example:

The figure below shows an example of a TransportLinkSequence. Note that the link sequence can keep the same geographical extent *despite new links connecting along the way. This makes it well suited for usage when connecting other information (features) to the network.*

Figure 18 - Example of TransportLinkSequence

5.3.2.10.1 Attribute

Name	Type	Description
<<voidable>> validFrom (INSPIRE DS TN)	Date	The point in time from when the link sequence takes effect. *Note:* *In the general case (and in INSPIRE) a link sequence has a complete validity period (with a start and end). In our case, we believe it is sufficient that the constituent links describe the validity on the lowest level. In its role as a linear element, a link sequence should always be valid in its entirety, i.e. you should always be able to measure along the whole object, even if parts have been removed from service.*

5.3.2.10.2 Relations

Name	Type	Description

link [1..*] {ordered} (INSPIRE DS TN)	TransportLink	The sequence of transport links that is included in the transport link sequence. *Restriction 1:* *For each link (L1) included in the sequence of links, there must be no overlaps with any other link (L2) within the same sequence* *Context TransportLinkSequence inv:* *link->forAll(l1, l2 | l2 <> l1 implies not ((l1.startMeasure >= l2.endMeasure) or (l1.endMeasure <= l2.startMeasure))* *Restriction 2:* *All and every link referred to, shall belong to the same network as the link sequence* *Context TransportLinkSequence inv:* *link->forAll(l | l.inNetwork = self.inNetwork)* *Restriction 3:* *The relative length measurement should start with 0 and end with 1 for a link sequence.* *Context TransportLinkSequence inv:* *link->first().startMeasure = 0* *link->last().endMeasure = 1* *Restriction 4:* *There must be no gaps in the relative length measurement*

		for a link sequence. The sum of the relative lengths for incoming links shall be 1 (seen in combination with Restriction 1 and 2) *Context TransportLinkSequence inv:* *link.(endMeasure-startMeasure)->sum()=1* *Restriction 5:* *Incoming links should be sorted in ascending startMeasure-order* *Context TransportLinkSequence inv:* *link->asSequence() = link->sortedBy(startMeasure)*

5.3.2.11 Class <<featureType>> TransportLink (INSPIRE DS TN)

A type of linear network element that connects two nodes. A *TransportLink* is an abstract base class. A link has a positive direction from the start node to the end node which should also be consistent with the direction of its geometry.

Although we do not in this document specify concrete requirements on geometry characteristics such as tolerance, it is a requirement that the link geometry within a given tolerance is consistent with the start and end nodes geometry.

Note:

A link corresponds to both a link (NW_ValidEdge) and a reference link part (NW_RefLinkPart) in SS 637004 (SIS/TK320 - SS 637004, 2009).

In this basic model, independent of transport type, there are no ports as a mechanism (these are instead located in the road specific package of the model due to backward compatibility reasons). In the absence of ports, the link itself needs to carry definition of the length-coordinates for the primary linear reference system. Compared to the model in SS 637004 (SIS/TK320 - SS 637004, 2009) *for road networks, this corresponds to the relative length for start and end ports. All positions along the link shall be interpolated between the specified start and end values.*

A link describes the path between two nodes. This means by definition that the links appear and disappear as soon as new links are connected to existing ones. This makes the link more volatile than the link sequence, and is therefore less suitable for use in order to connect features to the network.

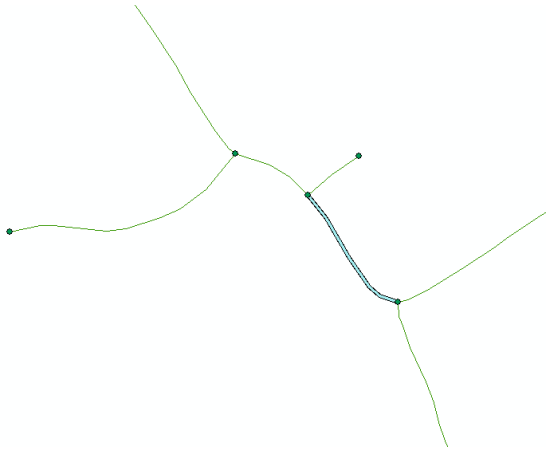


Figure 19 – Example of a link for road

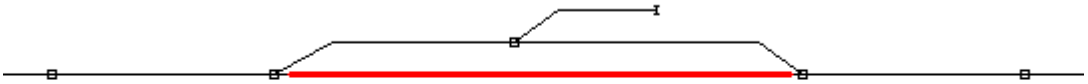


Figure 20 - Example of a link for railway

5.3.2.11.1 Attribute

Name	Type	Description
centrelineGeometry (INSPIRE DS TN)	GM_Curve (ISO/TC211 - ISO 19107)	The geometry which represents the center line of the real network element. *Restriction 1:* *Only the geometry of type Line String (i.e. breaking points with straight lines between (polyline)) is permitted.* *Context TransportLink inv:* *centrelineGeometry. segment->size() = 1 and centrelineGeometry. segment->forall(s | s. oclIsTypeOf(GM_LineString)*

		Restriction 2: *Coordinate dimension must be three-dimensional.* *Context TransportLink inv:* *centrelineGeometry. coordinateDimension() = 3*
startMeasure	LinkPositionMeasure	Start value for the primary linear referencing system for the current link in the case where the link position is indicated by *LinkPositionMeasure*. *Note:* *Replaces the entry of relative length that today is indicated by ports in the road network model.* *Restriction:* *Shall be a value between 0 and 1.* *Context TransportLink inv:* *startMeasure >= 0 and startMeasure <= 1*
endMeasure	LinkPositionMeasure	End value of the primary linear reference system for the current link, in the case where the link position is indicated by *LinkPositionMeasure*. *Note:* *Replaces the entry of relative length that today is indicated by ports in the road network model.*

		Restriction 1: *Shall be a value between 0 and 1.* *Context TransportLink inv:* *endMeasure >= 0 and endMeasure <= 1* *Restriction 2:* *endMeasure must be greater than startMeasure.* *Context TransportLink inv:* *endMeasure > startMeasure*
length	Measure (ISO/TC211 - ISO/TS 19103)	The agreed length of the link in meters (can be the geometric length). This length is the value to which absolute lengths (*LinkPositionDist)* in the network connections shall be related. *Note:* *This attribute exists primarily to avoid misunderstandings regarding the agreed length of a link. Even though the geometric length is used there may still be problems with the interpretation when it comes to how the z (height) is used in length calculation.*
<<voidable>> valid (INSPIRE DS TN)	DurationDays	The time period during which the link is valid.

5.3.2.11.2 Relations

Name	Type	Description

<<voidable>> startNode (INSPIRE DS TN)	TransportNode	The node where the link starts. *Restriction:* *The start node shall belong to the same network as the link.* *Context TransportLink inv:* *self.inNetwork = startNode.inNetwork*
<<voidable>> endNode (INSPIRE DS TN)	TransportNode	The node where the link ends. This node may be the same node as the start node. *Restriction:* *The end node shall belong to the same network as the start node (and the link).* *Context TransportLink inv:* *self.inNetwork = endNode.inNetwork*

5.3.2.12 Class <<featureType>> TransportNode (INSPIRE DS TN)

A type of network elements that represent a significant position in the network that always occurs at the beginning or end of a link. A node is an abstract base class.

5.3.2.12.1 Attribute

Name	Type	Description
geometry (INSPIRE DS TN)	GM_Point (ISO/TC211 - ISO 19107)	The node's position. *Restriction:* *Coordinate dimension must be three-dimensional.* *Context TransportNode inv:* *geometry. coordinateDimension() = 3*

<<voidable>> /valid (INSPIRE DS TN)	DurationDays	The time period when the node is valid. The value is derived from the validity in the connecting links. *Note:* *Talking about node validity is difficult from an business perspective. However, it may be practical from a usage perspective to have validity available. The following applies as a proposal for deriving the validity of nodes:* *For a node describing an intersection or an end, the from date equals the first date for which the node describes an intersection or an end, and the to date equals the last date for which the node describes an intersection or an end.* *A node that never describes an intersection or an end (i.e. always has exactly two connecting links) shall not be valid at any duration in time.*

5.3.2.12.2 Relations

Name	Type	Description
<<voidable>> spokeStart [0..*] (INSPIRE DS TN)	TransportLink	The links that start in this node. *Restriction:* *All connecting links shall belong to the same network as the node.*

		Context TransportNode inv: *spokeStart->forAll(l | l.inNetwork = self.inNetwork*
<<voidable>> spokeEnd [0..*] (INSPIRE DS TN)	TransportLink	The links that terminates in this node. *Restriction:* *All connecting links shall belong to the same network as the node.* *Context TransportNode inv:* *spokeEnd->forAll(l | l.inNetwork = self.inNetwork*

5.3.2.13 Class <<featureType>> TransportNetworkArea (INSPIRE DS TN)

Represents a two-dimensional network element. A TransportNetworkArea is an abstract base class.

Note:

The use of this class is as yet somewhat unclear. The class exists in INSPIRE and is meant to be used to describe two-dimensional objects with a function in the network. One can for example consider the use in those cases where definitions actually concerns areas, in contrast to a designated set of network elements. An example would be a densely built-up area, as exemplified below:

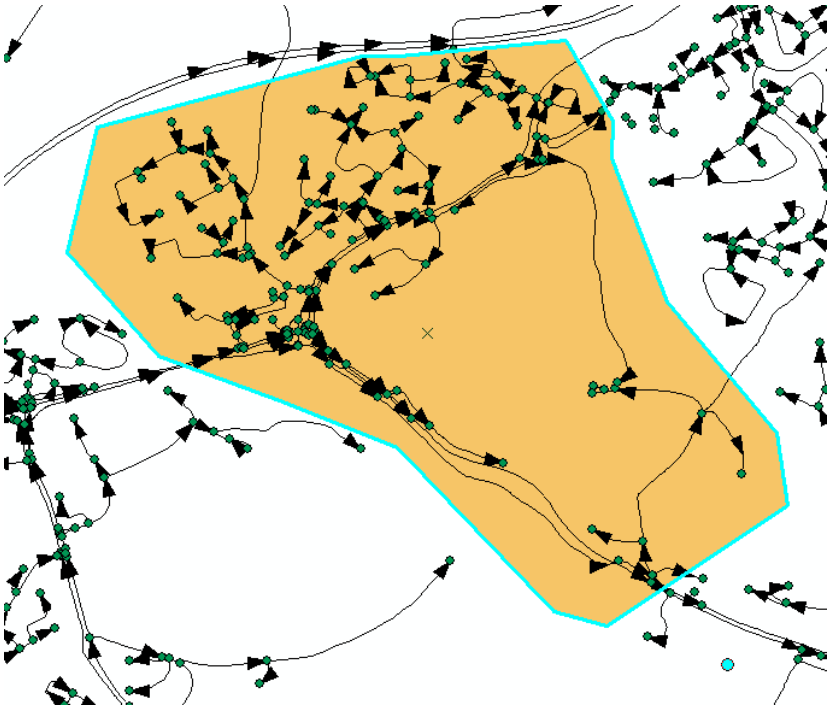


Figure 21 - Example of TransportNetworkArea

If we connect information such as traffic regulations to an area in this way, it is possible to specify a set of rules implying that additional network elements in the area automatically inherit the properties of the area if this is the desired behavior, which can be a big advantage.

5.3.2.13.1 Attribute

Name	Type	Description
geometry (INSPIRE DS TN)	GM_Surface (ISO/TC211 - ISO 19107)	Specifies the geometry of the network element as a two-dimensional surface. *Restriction 1:* *Only geometry of the type Polygon where the surface is only represented by their boundary lines is allowed.* *Context TransportNetworkArea inv:* *geometry.patch->size() = 1 and geometry.patch->forall(p | p.oclIsTypeOf(GM_Polygon)) and geometry.patch->forall(p | p.spanningSurface->size() = 0)*

		Restriction 2: *Coordinate dimension must be three-dimensional.* *Context TransportLink inv:* *geometry.coordinateDimension() = 3*
<<voidable>> valid (INSPIRE DS TN)	DurationDays	The time period during which the surface is valid

5.3.2.14 Class <<featureType>> TransportSystem (SIS/TK320 - SS 637004, 2009)

A grouping of elements which on a less detailed level is considered as one network element.

Note:

It corresponds to the class NW_System in (SIS/TK320 - SS 637004, 2009).

The reason that NetworkReference is used instead of referring directly to the network elements is to handle the case that a network element at a level of detail corresponds to a part of a network element on a different level. The example below shows such a case (LS2 and LS3 corresponds to a part of the LS1).

The use of NetworkReference also provides the ability to handle the case when the directions between the two levels are not consistent.

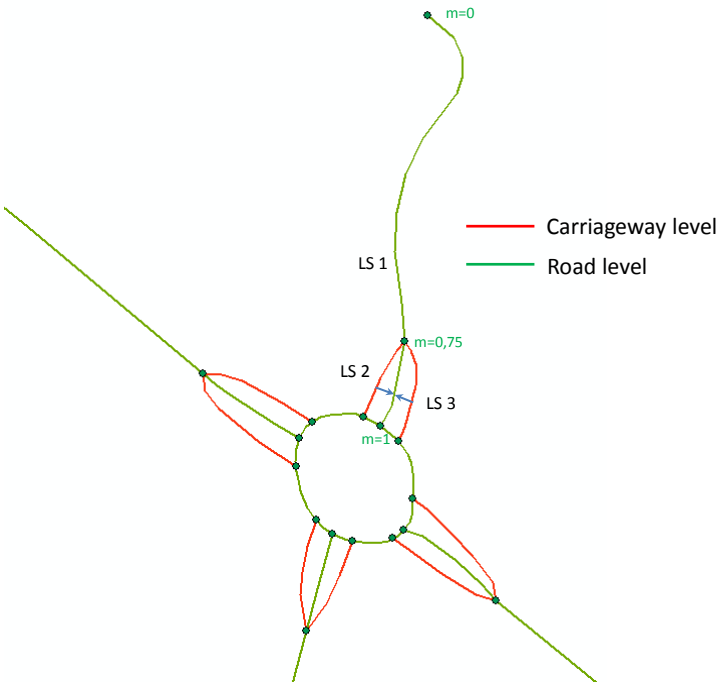


Figure 22 - Example of mapping between levels of detail

superElement	subElements

{
LocationInstance:"LS 1",
startPosition:0,75
endPosition:1
}
	[{
	locationInstance:"LS 2",
	startPosition:0
	endPosition:1
	},
	{
	locationInstance:"LS 3",
	startPosition:0
	endPosition:1
	}]

Restriction:

The elements shall belong to the same network regardless of if they are on a more or less detailed level.

```
Context TransportSystem inv:

subElements->forall(s |
s.locationInstance.oclAsType(TransportNetworkElement).inNetwork =
superElement.locationInstance.oclAsType(TransportNetworkElement).inNetwork
```

5.3.2.14.1 Relations

Name	Type	Description

superElement (SIS/TK320 - SS 637004, 2009)	TransportNetworkReference	The network element at the a less detailed level.
elements [1..*] (SIS/TK320 - SS 637004, 2009)	TransportNetworkReference	The network elements at the more detailed level.

5.3.2.15 Class <<dataType>> LinkPosition (SIS/TK320 - SS 637004, 2009)

An abstract base class that specifies a relative or absolute position along a linear element (*GeneralisedLink*).

5.3.2.15.1 Attribute

Name	Type	Description
/measure	Measure (ISO/TC211 - ISO/TS 19103)	Derived attribute specifying that all different subtypes for this class must be able to provide a value for the measure regardless of how the original measurement for the link position was recorded (e.g. km + m) *Note:* *The reason for this attribute is to easily compare the position for different data without having to recalculate between different types of link positions.*

5.3.2.16 Class <<dataType>> LinkPositionMeasure

Asubclass to *LinkPosition* that represents a position which is defined by the distance relative to the *startMeasure* and *endMeasure* for the linear element. Compare with *NW_LinkPositionRelDist* in (SIS/TK320 - SS 637004, 2009).

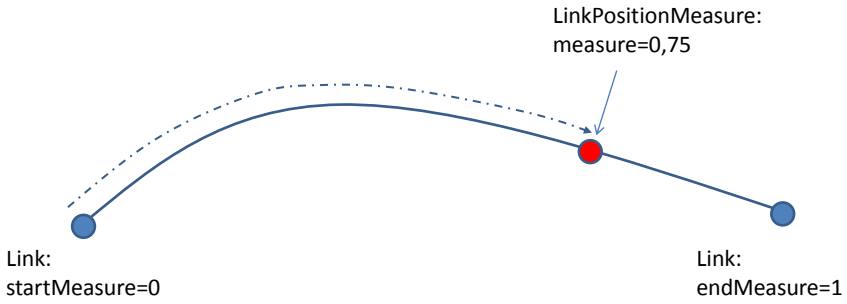


Figure 23 - LinkPositionMeasure

5.3.2.16.1 Attribute

Name	Type	Description
measure (SIS/TK320 - SS 637004, 2009)	Measure (ISO/TC211 - ISO/TS 19103)	Distance relative to the startMeasure and endMeasure for the linear element. *Restriction:* *Shall be a value between 0 and 1.* *Context LinkPositionMeasure inv:* *measure >= 0 and measure <= 1*

5.3.2.17 Class <<dataType>> LinkPositionDist (SIS/TK320 - SS 637004, 2009)

It is a subclass to *LinkPositionDist* that represents a position defined by the distance from the beginning or end of a linear element or from a defined point along the linear element. The class is an abstract base class.

5.3.2.17.1 Attribute

Name	Type	Description
distance (SIS/TK320 - SS 637004, 2009)	Measure (ISO/TC211 - ISO/TS 19103)	Distance in meters, in relation to *TransportLink.Length*. *Note:* *A distance in the direction of the linear element is indicated by a positive number (plus*

		distance). A distance against the direction of the linear element is indicated by a negative number (minus distance).

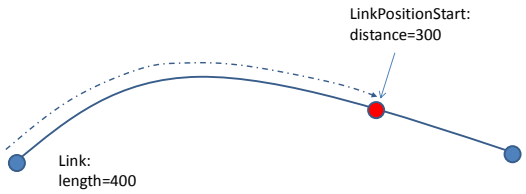
5.3.2.18 Class <<dataType>> LinkPositionStart (SIS/TK320 - SS 637004, 2009)

Represents a position defined by a distance from the start of a linear element.

Restriction:

Context LinkPositionStart inv:

self. distance >= 0.



5.3.2.19 Class <<dataType>> LinkPositionEnd (SIS/TK320 - SS 637004, 2009)

Represents a position defined by a distance from the end of a linear element.

Restriction:

Context LinkPositionEnd inv:

self. distance <= 0.

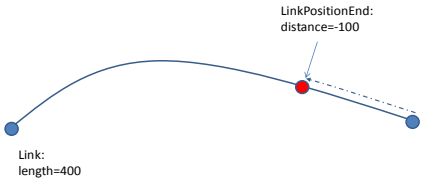


Figure 24 - LinkPositionEnd

5.3.2.20 Class <<dataType>> LinkPositionPoint (SIS/TK320 - SS 637004, 2009)

Represents a position defined by a distance from the anchor point on a link for a known point.

Note:

Anchor point is defined as the closest point on the link where there is a perpendicular line from the link (a normal to the link geometry) which is intersecting the designated point.

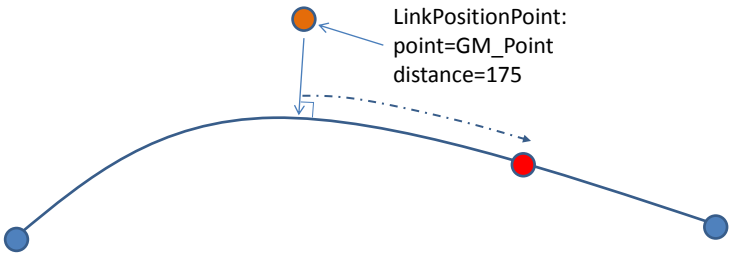


Figure 25 - LinkPositionPoint

5.3.2.20.1 Attribute

Name	Type	Description
point (SIS/TK320 - SS 637004, 2009)	GM_Point (ISO/TC211 - ISO 19107)	A point. *Note:* *The distance which is inherited from the class LinkPositionDist refers to the distance along the linear element up to the anchor point for the specified point.*

5.3.2.21 Class <<dataType>> LinkPositionReferent

Represents a position defined by a distance from a known referent.

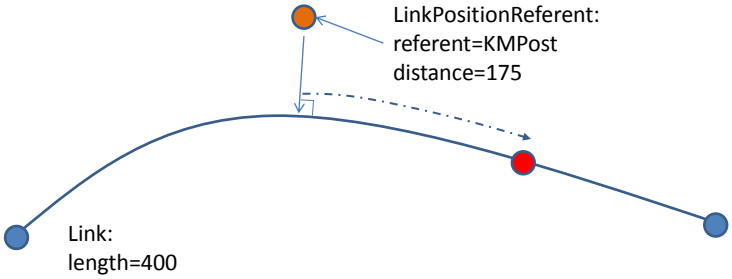


Figure 26 - LinkPositionReferent

5.3.2.21.1 Relations

Name	Type	Description
referent	Referent	A referent. *Note:* *The distance which is inherited from the class LinkPositionDist refers to the distance along the linear element up to the referent's position relative to the linear element.*

5.3.2.22 Class <<featureType>> Referent

It is an abstract base class that represents a referent that can be used for specifying linear references. The class can be considered as originating from the class *LR_Referent* in (ISO/TC211 - ISO 19148). A concrete type of referent in this document is *KMPost* (kilometer post) in the package RailwayTransportNetwork in Chapter 5.3.6.

5.3.2.22.1 Attribute

Name	Type	Description
id	CharacterString	Identity of the referent *Note:* *Corresponds to name in* (ISO/TC211 - ISO 19148)
position [0..1]	GM_Point (ISO/TC211 - ISO 19107)	Geographical position of the referent. *Note:* *Linear position for referents is defined in the respective concrete class.*

5.3.2.23 Class <<dataType>> TransportNetworkReference (INSPIRE DS TN)

A concrete base class for all types of network connections. It corresponds to NW_Extent in (SIS/TK320 - SS 637004, 2009).

5.3.2.23.1 Relations

Name	Type	Description

locationInstance (SIS/TK320 - SS 637004, 2009)	LocationInstance	The position of objects referred to in the network connection.

5.3.2.24 Class <<dataType>> DurationDays

A class that describes a data type for specifying the time period with resolution in days. The class is broken out because this type of time period is used for many classes in the model to describe validity.

5.3.2.24.1 Attribute

Name	Type	Description
validFrom	Date	The date from which the time period is valid.
validTo [0..1]	Date	The date from which the time period is no longer valid. If validTo is not specified the time period is assumed to be valid indefinitely.

5.3.2.25 Class <<enumeration>> TransportTypeValue

The class is an enumeration of possible transport types.

5.3.2.25.1 Permissible values

Name	Description
Air	Aviation
Cable	Cableway
Rail	Railway
Road	Road
Water	Shipping

5.3.2.26 Class <<codeList>> DetailLevelCode (SIS/TK320 - SS 637004, 2009)

An enumeration of the valid levels of detail.

5.3.2.26.1 Allowed values

Name	Description
Micro	Indicates micro level for tracks, i.e. tracks and switches.

Macro	Indicates macro level for tracks, i.e. the line and the operatinon station/operational site boundaries.
Road (SIS/TK320 - SS 637004, 2009)	Indicates road level. This represents one or two carriageways.
Carriageway (SIS/TK320 - SS 637004, 2009)	Refers to carriageway level. This represents one or more lanes.
Lane (SIS/TK320 - SS 637004, 2009)	Refers to lane level.

5.3.2.27 Other INSPIRE- requirements

INSPIRE places requirements on the following information regarding the traffic network at a general level. These will be treated as network related features at the Swedish Transport Administration and are therefore not included in the model.

Name	**Description**
MaintenanceAuthority	Principal fund manager
OwnerAuthority	Owner
VerticalPosition	Specifies whether an element of the network is above, below or on the ground.
TrafficFlowDirection	Specifies possible traffic direction for elements in the network: - BothDirections - InDirection - InOppositeDirection
ConditionOfFacility	Indicates the state of the elements in the network: - Disused - Functional - Projected - UnderConstruction
RestrictionForVehicles	Specifies restrictions regarding driving vehicles for elements in the network. - MaximumDoubleAxleWeight - MaximumHeight - MaximumLength - MaximumSingleAxleWeight - MaximumTotalWeight - MaximumTripleAxleWeight

	- MaximumWidth
AccessRestriction	Specifies special conditions regarding the access to the elements in the network: - ForbiddenLegally - PhysicallyImpossible - Private - PublicAccess - Seasonal - Toll

5.3.2.28 Other requirements on the network

This information model does not impose specific requirements on data beyond those specified above. Requirements such as timeliness, thematic accuracy, geometric accuracy/resolution, etc. are defined in separate data product specifications and can also differ between different types of business data and applications. Presented at the end of this document are the quality descriptive properties that at different levels shall or should be collected and handled.

As previously noted, this model for the network is completely free of data other than those deemed absolutely essential, i.e. geometry, topology, linear reference system and the time period for when each network element is open to traffic. Network connected features are not described within the frame of this information model. However, there is a generic feature model in Chapter 5.3.8 and 5.3.9 that can be used for all types of features.

There is information not included in the network model but has been considered to be closely related to the network. This information is required in order to support basic services that are widely used throughout the Swedish Transport Administration's operations. Typical for this information is that there may not be any specific designated organizational unit responsible for collection and maintenance.

This information shall be treated as network related features according to the model in Chapter 5.3.8 and 5.3.9 (i.e. separate features linked to the network). The table below shows these features.

Category of data/Function	**Selection Railway**	**Selection Road**
Identification	• Routes • Track Portion • Traffic Junction • Track number • UNE(UpDownSingle(Track), (*UppNedEnkel*)) • Track Switch/Track End (id)	• Road Number/Road Name • Street Name • Link Role • Intersection (type, name, number)
State	• Status (Planned, Opened for traffic, Turned Off, Disused, Demolished) • Planning Alternative	• Status (Planned, Opened for traffic, Turned Off, Demolished) • Planning Alternative

Overall classification	• Railway Track Type (Similar to Quality of operation and maintenance) • Main Track/Side Track	• Functional Road Class Quality of operation and maintenance
Road/Track Design	• Multi Tracks/Double Tracks/ Single Tracks	• Road Type (Form of way) • Number of Lanes
Ownership	• Owner • Infrastructure Manager • Infrastructure Manager Type (state, municipal, private)	• Owner • Road Manager • Road Manager Type (state, municipal, private)
Organization	• Operation station • Maintenance Area • Region	• Maintenance Area • Region
Geography	• Country • County • Municipality • National Border	• Country • County • Municipality • National Border

Regarding history the model is already handling past, present and future. Past and present time has occurred and can be described unambiguously in the model. As for the future, there may be different alternatives. Alternative future solutions are described through the various options being indicated and held together by a network connected feature "Planning Alternative".

5.3.2.29 Secondary linear referencing system

The model described above describes a reference system (reference network) to which other data can be connected. The reference system is based on linear elements, nodes and some additional types of elements. There are additional linear referencing systems used today, none of which is being described explicitly in any one model. One such example is road numbers where you frequently refer to locations in the road network using road numbers and distances along the road number. This document expresses currently no model for description of additional (secondary) linear referencing systems. To avoid the need to build applications for specific handling of any possible additional linear referencing systems, a general approach to this should be described if or when the need for this occurs. Starting from the basic principles of a linear referencing system (see (ISO/TC211 - ISO 19147)) the following information is required:

- Linear elements must be identifiable unambiguously
- Positions along linear elements must be unambiguously identifiable with the support of a description of the method used for linear referencing.

It is usually a requirement that it is possible to transform positions between all existing reference systems. It means that additional secondary linear referencing systems should be possible to map towards the primary reference system described in this document. This means in practice:

- A description must be provided for how other linear elements are mapped to the linear elements that exist in this model

- A description must be provided for how positions on other linear elements are mapped to positions on the linear element which exists in this model

This can be achieved by describing a network connected feature that constitutes the linear elements of a secondary linear referencing system. The extent of the features on the linear elements in the reference network constitutes simultaneously a mapping between the secondary and primary reference systems.

5.3.3 Package – Transport network reference

5.3.3.1 Overview

The package describes the specialized classes that will be used to connect data to the reference network. The concrete classes in this package are used to describe locations of the features relative to the network. See (SIS/TK320 - SS 637004, 2009).

The different methods for network connection are specified as follows:

- Connection to arbitrary position objects (*LocationInstance*)
- Connection along the whole/part of linear element (*GeneralisedTransportLink*)
    - Distance
    - Point
- Connection to an identified set of linear elements (*LinkSet*)
- Connection to node (*TransportNode*)
    - Connection to the node
    - Connection to a turn through the node
- Connection to a maneuver
    - A sequence of turns
- Connection to surface (*TransportNetworkArea)*
- Connection to network connection, i.e. network transition or network conflict (*InterNetworkConnection/InterNetworkConflict)*

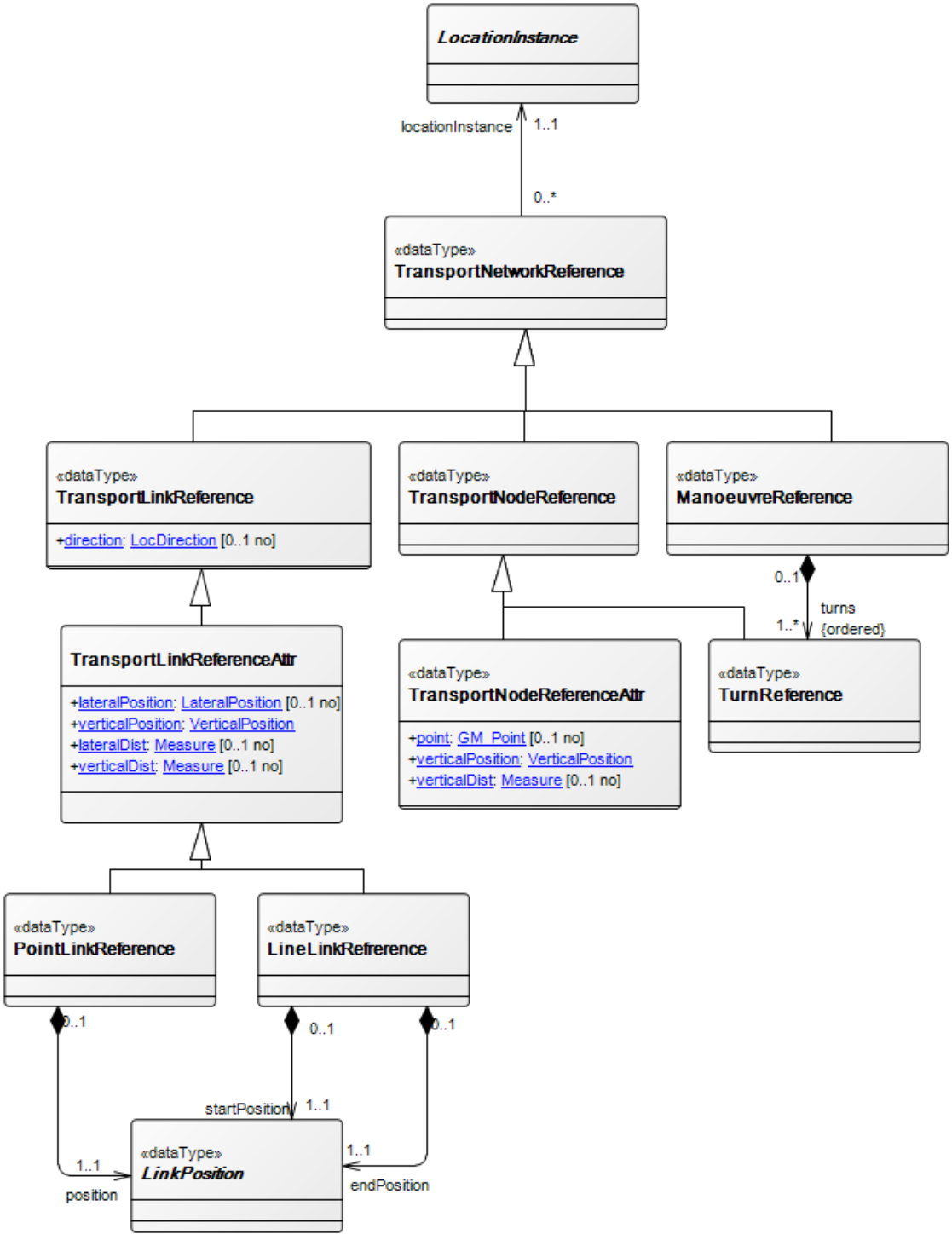


Figure 27 – Transport network reference, basic types

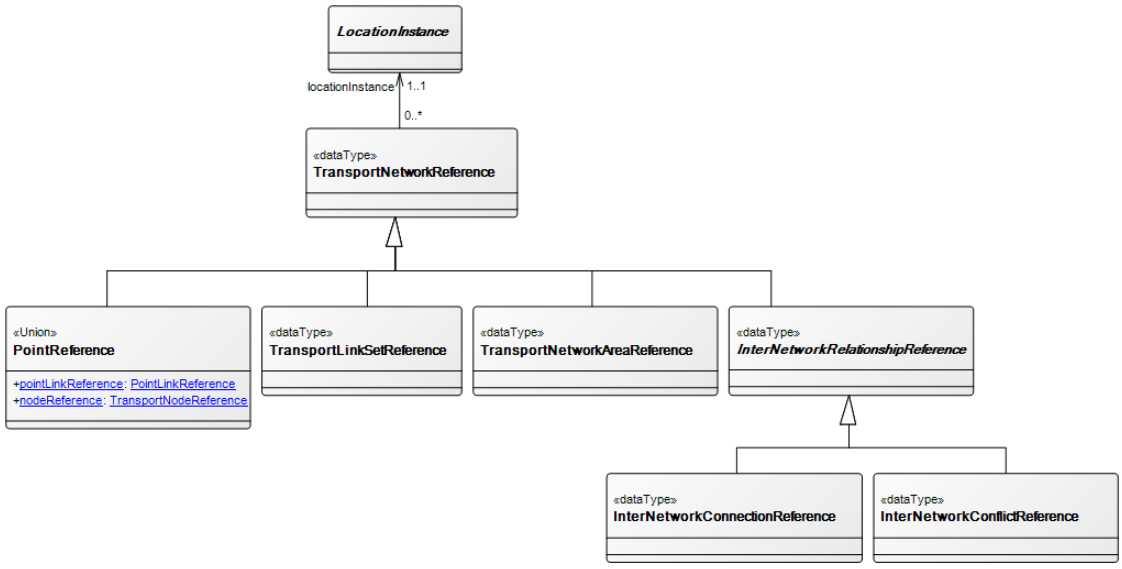


Figure 28 - Transport network reference other types

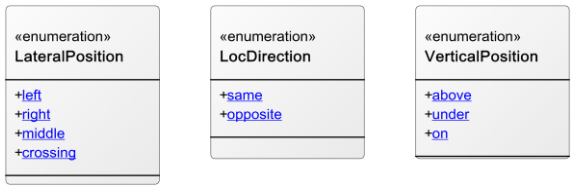


Figure 29 - Enumerationer and data types

5.3.3.2 Class <<dataType>> TransportLinkReference (SIS/TK320 - SS 637004, 2009)/ (INSPIRE DS TN)

A base class for network connection to linear elements in the network (*GeneralisedTransportLink*). Please note that currently there is no class for connecting exclusively towards any of the subclasses to the *GeneralisedTransportLink.*

Restriction:

Referenced position object should be of type GeneralisedTransportLink or any of its subclasses.

```
Context TransportLinkReference inv:

locationInstance.oclIsKindOf(GeneralisedTransportLink)
```

Example:

To describe a location for any type of feature in relation to a reference network according to Transport Link Reference only the following information is required:

- *Identity of the linear element (mandatory data)*
- *Direction relative to the element. Optional attribute that is used if the data is dependent on direction.*

This can be illustrated in the table below:

Feature	***LocationInstance (Id for linear element)***	***Direction***

Abc123	*111:222*	*Same*
Xxxyyy	*222:333*	*Opposite*

5.3.3.2.1 Attribute

Name	Type	Description
direction [0..1] (SIS/TK320 - SS 637004, 2009)	LocDirection	Direction *(same, opposite – see LocDirection)* relative to the positive direction of the linear element. *Note:* *Definition of turns requires the directional attribute to be at this level (see* Class <<dataType>> TurnReference *below).*

5.3.3.3 Class <<dataType>> TransportLinkReferenceAttr

A base class for network connection to linear elements in the network (*GeneralisedTransportLink*).

5.3.3.3.1 Attribute

Name	Type	Description
lateralPosition [0..1] (SIS/TK320 - SS 637004, 2009)	LateralPosition	Lateral position *(left, right, middle, crossing – see LateralPosition)* relative to the positive direction of the linear element.
verticalPosition [0..1] (SIS/TK320 - SS 637004, 2009)	VerticalPosition	Vertical position *(above, under, on – see VerticalPosition)* relative to the linear element. *Note:* *Is named heightPosition in Swedish standard but is here renamed for better "symmetry" with verticalDist.*

lateralDist [0..1] (SIS/TK320 - SS 637004, 2009)	Measure (ISO/TC211 - ISO/TS 19103)	Lateral distance in meters relative to the linear element. *Restriction:* *This is given as an absolute value and should be read together with the lateralPosition above.* *If lateralPosition has the value "**middle**" the lateral distance is 0 and lateralDist is not specified.* *Context TransportLinkReferenceAttr inv:* *lateralDist ->size() = 1 implies lateralPosition->size = 1 and lateralDist > 0* *lateralPosition = middle implies lateralDist->size() = 0*
verticalDist [0..1] (SIS/TK320 - SS 637004, 2009)	Measure (ISO/TC211 - ISO/TS 19103)	Vertical distance in meters relative to the linear element. *Restriction:* *Is always specified with a positive number. Above respective below is specified with the attribute verticalPosition. If verticalPosition has the value **"on"** the height difference is 0 and verticalDist is not specified.* *Context TransportLinkReferenceAttr inv:* *verticalDist ->size() = 1 implies verticallPosition->size = 1 and verticalDist > 0* *verticalPosition = on implies verticalDist->size() = 0*

5.3.3.4 Class <<dataType>> PointLinkReference (SIS/TK320 - SS 637004, 2009)/ (INSPIRE DS TN)

A subclass to *TransportLinkReferenceAttr* indicating location in the form of a point on linear elements in the network (*GeneralisedTransportLink*).

Figure 30 - Example of point *PointLinkReference*

The figure above shows an example of a point location. The different variants to describe the position along linear elements have been specified previously in the document in separate chapters (*LinkPosition* and subtypes).

Restriction:

The specified position shall be included in the referenced linear element.

```
Context PointLinkReference inv:

locationInstance->oclAsType(GeneralisedTransportLink).Contains(position)
```

Example:

To describe a location for any type of feature relative to the reference network according to PointLinkReference the following information is used:

- *Identity for the linear element (mandatory data)*
- *Position along the linear element (mandatory data)*
  - *Can be described with relatively or absolute measurement value or distance from known point or referent*
- *Direction relative to the element. Optional attribute that is used if the data is dependent on direction.*
- *Lateral position. Optional attribute that is used if the data is side dependent.*
- *Lateral distance. Optional attribute that is used if the lateral distance relative to the linear element is of interest*
- *Vertical position. Optional attribute that is used if the data is height dependent.*
- *Vertical distance. Optional attribute that is used if the height distance relative to the linear element is of interest*
-

This can be illustrated in the table below (shaded columns are optional):

Feature	*Location-Instance (Id for linear element)*	*Position*	*Direction*	*Lateral Position*	*Lateral Distance*	*Vertical Position*	*Vertical Distance*
Abc123	*111:222*	*0.43*	*Same*	*right*	*5*	*above*	*3*
Xxxyyy	*222:333*	*0.57*	*Opposite*	*left*	*7*	*on*	*0*

Note that the above example uses the relative length measurement as position, i.e. a value between 0 and 1. The attribute can vary dependent on the type of link position indication (LinkPosition), e.g. length or coordinate.

5.3.3.4.1 Relations

Name	Type	Description
position (SIS/TK320 - SS 637004, 2009)	LinkPosition	Linear position for point

5.3.3.5 Class <<dataType>> LineLinkReference (SIS/TK320 - SS 637004, 2009)/ (INSPIRE DS TN)

Subclass to *LinkReference* indicating position in the form of a distance in linear elements in the network (*GeneralisedTransportLink*).

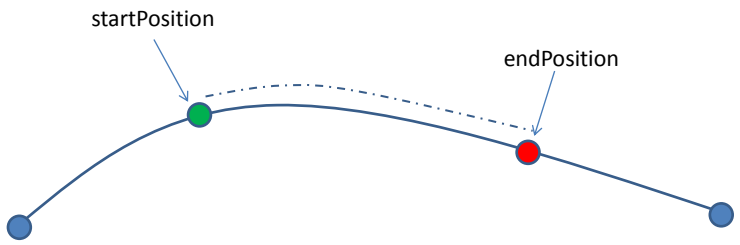


Figure 31 - Example of LineLinkReference

The figure above shows an example of a *LineLinkReference*. The different variants to describe the position along linear elements have been reported previously in the document in separate chapters (*LinkPosition* and sub types).

Restriction:

The stated positions shall be included in the referenced linear element.

Context PointLinkReference inv:

locationInstance->oclAsType(GeneralisedTransportLink).Contains(startPosition) and locationInstance->oclAsType(GeneralisedTransportLink).Contains(endPosition)

Example:

To describe a location for any type of feature relative to the reference network according to LineLinkReference the following information is used:

- *Identity for the linear element (mandatory data)*
- *Start position and end position along the linear element (mandatory data)*
  - *Can be described with relatively or absolute measurement value or distance from known point or referent*
- *Direction relative to the element. Optional attribute that is used if the data is dependent on direction.*
- *Lateral position. Optional attribute that is used if the data is side dependent.*
- *Lateral distance. Optional attribute that is used if the lateral distance relative to the linear element is of interest*
- *Vertical position. Optional attribute that is used if the data is height dependent.*
- *Vertical distance. Optional attribute that is used if the vertical distance relative to the linear element is of interest*
-

This can be illustrated in the table below. For the sake of simplicity, an example with no data for side splitting, lateral distance, height splitting and height distance is shown:

Feature	***LocationInstance (Id for linear element)***	***startPosition***	***endPosition***	***Direction***
Abc123	*111:222*	*0.1*	*0.3*	*Same*
Xxxyyy	*222:333*	*0*	*1*	*Opposite*

5.3.3.5.1 Relations

Name	**Type**	**Description**
startPosition (SIS/TK320 - SS 637004, 2009)	LinkPosition	Starting Position
endPosition (SIS/TK320 - SS 637004, 2009)	LinkPosition	End position

5.3.3.6 Class <<dataType>> TransportNodeReference (SIS/TK320 - SS 637004, 2009)

A base class to specify the location of a node in the network. It is a subclass to *TransportNetworkReference*.

Restriction:

Referenced position object should be of the type TransportNode:

```
Context TransportNodeReference inv:

locationInstance.oclIsKindOf(TransportNode)
```

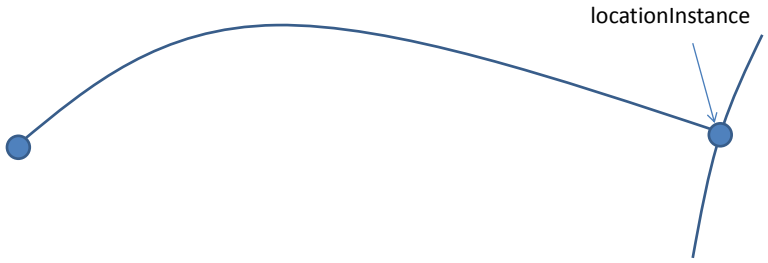


Figure 32 - Example of TransportNodeReference

5.3.3.7 Class <<dataType>> TransportNodeReferenceAttr (SIS/TK320 - SS 637004, 2009)

A subclass to *TransportNodeReference* to indicate the position of the node in the network that is not a turn. The class is used to connect a feature to a node, and further specify the location.

Example:

To describe a location for any type of business data relative to the reference network according to TransportNodeReferenceAttr the following information is used:

- *Identity for the node (mandatory attribute)*
- *Exact position (optional attribute)*
- *Vertical position. Optional attribute that is used if the vertical position in relation to the node is of interest.*
- *Vertical distance. Optional attribute that is used if the vertical distance relative to the node is of interest*
-

This can be illustrated in the table below (shaded columns are optional):

Feature	***LocationInstance (Id for node)***	***Point x, y, z***	***Vertical Position***	***Vertical Distance***
Abc123	*333:444*	*Point*	*above*	*3*

Xxxyyy	*555:666*	*Point*	*on*	*0*

5.3.3.7.1 Attribute

Name	Type	Description
point [0..1] (SIS/TK320 - SS 637004, 2009)	GM_Point (ISO/TC211 - ISO 19107)	Measured start position for the extent of the feature. If this attribute value is missing, verticalPosition and verticalDist is indicated relative to the node's position.
verticalPosition [0..1] (SIS/TK320 - SS 637004, 2009)	VerticalPosition	Specifies the vertical position of the feature *(above, under, on – se VerticalPosition)* relative to the node, or to the point if stated. *Note:* *Is named heightPosition in the Swedish standard but is here renamed for better symmetry with verticalDist.*
verticalDist [0..1] (SIS/TK320 - SS 637004, 2009)	Measure (ISO/TC211 - ISO/TS 19103)	Specifies the vertical distance to the node, or to the point if stated. *Restriction:* *Is always specified with a positive number. Above respective below is specified with the attribute verticalPosition. If verticalPosition has the value **on** the vertical distance is 0 and verticalDist is not specified.*

Context
TransportNodeReferenceAttr inv:

verticalDist ->size() = 1 implies verticallPosition->size = 1 and verticalDist > 0

verticalPosition = on implies verticalDist->size() = 0

5.3.3.8 Class <<dataType>> TurnReference (SIS/TK320 - SS 637004, 2009)

The class defines a turn through a node.

The model for this is shown in the figure below.

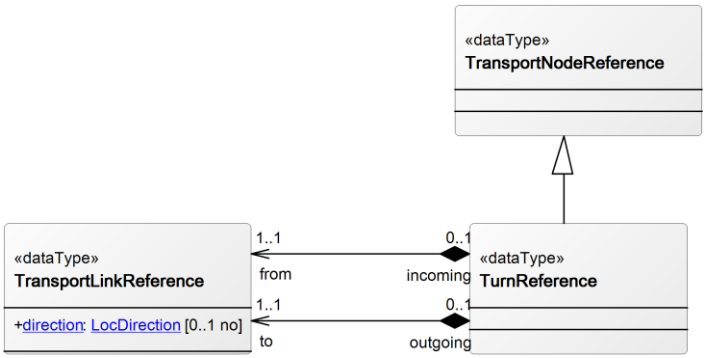


Figure 33 – Turn Reference

Example:

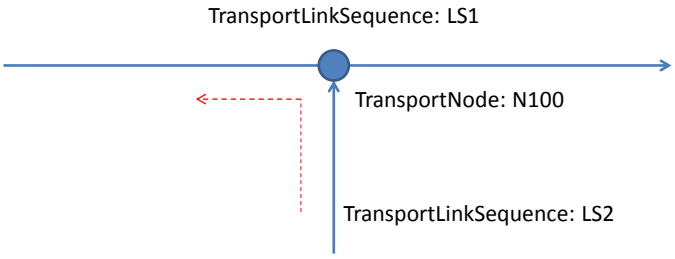


Figure 34 - Example of *TurnReference*

To describe a location for any type of feature relative to the reference network according to TurnReference the following information is used:

- *The identity of the node that the turn passes (mandatory attribute)*
- *The linear element that the turn starts from (mandatory attribute)*
  - *Identity*
  - *Direction relative to the linear element*
- *The linear element that the turn goes against (mandatory attribute)*
  - *Identity*
  - *Direction relative to the linear element*

The example in the figure above can be illustrated in the table below:

Feature	***Location-Instance (Id for node)***	***From LocationInstance (Id for linear element)***	***From direction***	***To LocationInstance (Id for linear element)***	***To direction***
Abc123	*N100*	*LS2*	*same*	*LS1*	*opposite*

Since the turn is defined between two link sequences (where link sequence "LS1" is split in half) it requires an indication of direction to be able to separate right and left turn. This is not required if links (Transport Link) are used for the turn definition, because then the turn is always unambiguously defined. However the advantage still applies that the use of link sequences is less volatile because the link sequences from the example above are stable, and still applies even if new intersections are added, for example, before or after the "N100" which is not the case with the corresponding links.

Restrictions:

When referencing to link sequence the direction must be indicated.

Context TurnReference inv:

from.locationInstance.oclIsKindOf(TransportLinkSequence) implies
from.direction->size() = 1 and from.direction = ("same" or "opposite")

to.locationInstance.oclIsKindOf(TransportLinkSequence) implies
to.direction->size() = 1 and to.direction = ("same" or "opposite")

5.3.3.8.1 Relations

Name	**Type**	**Description**
from (SIS/TK320 - SS 637004, 2009)	TransportLinkReference	Specifies from which network element you come and direction relative to that network element.
to (SIS/TK320 - SS 637004, 2009)	TransportLinkReference	Specifies which network element you continue on and direction relative to that network element.

5.3.3.9 Class <<dataType>> ManoeuvreReference (SIS/TK320 - SS 637004, 2009)

The class defines a maneuver.

Note:

A maneuver is a sequence of turns where each turn has a to-position on the same network element as the from-position for the following turn. An example of the use of maneuver is a restriction for a non-allowed u turn.

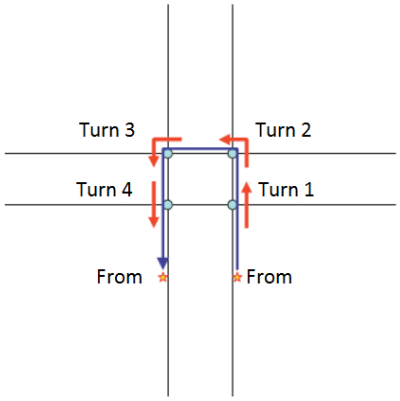


Figure 35 - Example of ManoeuvreReference

5.3.3.9.1 Relations

Name	Type	Description
turns [1..*] (SIS/TK320 - SS 637004, 2009)	TurnReference	Specifies the turns that are included in the maneuver.

5.3.3.10 Class <<Union>> PointReference

Class for network connections of point-shaped objects to the network either to a point on a linear element or to a node.

5.3.3.10.1 Attribute

Name	Type	Description
pointLinkReference	PointLinkReference	A reference to a point on the linear element
nodeReference	TransportNodeReference	A reference to a node

Note that because the class is of type <<Union>> only one of the attributes is used for each instance, i.e. either point on the linear element or node.

5.3.3.11 Class <<dataType>> TransportNetworkAreaReference

A class for network connection to the surface elements in the network (*TransportNetworkArea*).

Restriction:

LocationInstance shall be of type TransportNetworkArea:

```
Context TransportNetworkAreaReference inv:

locationInstance.oclIsKindOf(TransportNetworkArea)
```

5.3.3.12 Class <<dataType>> TransportLinkSetReference

A class for network connection to LinkSet-element in the network (*TransportLinkSet*).

Restriction:

LocationInstance shall be of type TransportLinkSet:

```
Context TransportLinkSetReference inv:

locationInstance.oclIsKindOf(TransportLinkSet)
```

5.3.3.13 Class <<dataType>> InterNetworkRelationshipReference

An abstract base class for network connection to InterNetworkRelationship-element in the network.

Restriction:

LocationInstance shall be of type InterNetworkRelationship:

```
Context InterNetworkRelationshipReference inv:

locationInstance.oclIsKindOf(InterNetworkRelationship)
```

5.3.3.14 Class <<dataType>> InterNetworkConnectionReference

A class for network connection to InterNetworkConnection-element in the network.

Restriction:

LocationInstance shall be of type InterNetworkConnection:

```
Context InterNetworkConnectionReference inv:

locationInstance.oclIsKindOf(InterNetworkConnection)
```

5.3.3.15 Class <<dataType>> InterNetworkConflictReference

A class for network connection to InterNetworkConflict-element in the network.

Restriction:

LocationInstance shall be of type InterNetworkConflict:

```
Context InterNetworkConflictReference inv:

locationInstance.oclIsKindOf(InterNetworkConflict)
```

5.3.3.16 Class <<enumeration>> LateralPosition (SIS/TK320 - SS 637004, 2009)

The class specifies an enumeration of the possible values of lateral position relative to the referenced linear element's positive direction.

5.3.3.16.1 Allowed values

Name	Description
left (SIS/TK320 - SS 637004, 2009)	Left side
right (SIS/TK320 - SS 637004, 2009)	Right side
middle (SIS/TK320 - SS 637004, 2009)	Middle
crossing (SIS/TK320 - SS 637004, 2009)	Intersecting/crossing

5.3.3.17 Class <<enumeration>> LocDirection (SIS/TK320 - SS 637004, 2009)

The class specifies an enumeration of the possible values of indicated direction relative to the referenced linear element's positive direction.

5.3.3.17.1 Allowed values

Name	Description
same (SIS/TK320 - SS 637004, 2009)	Same direction
opposite (SIS/TK320 - SS 637004, 2009)	Opposite direction

5.3.3.18 Class <<enumeration>> VerticalPosition (SIS/TK320 - SS 637004, 2009)

The class specifies an enumeration of the possible values of vertical position relative to the referenced network element.

5.3.3.18.1 Allowed values

Name	Description

above (SIS/TK320 - SS 637004, 2009)	Above
under (SIS/TK320 - SS 637004, 2009)	Under
on (SIS/TK320 - SS 637004, 2009)	On

5.3.4 Package – Transfer node

5.3.4.1 Overview

The package describes the classes that are used to link different transport types and transport means via transfer nodes. The basis of the model is taken from (ISO/TC211 - ISO 19147). Note that ISO 19147 so far resides on CD-level (Committee Draft) which means changes may occur.

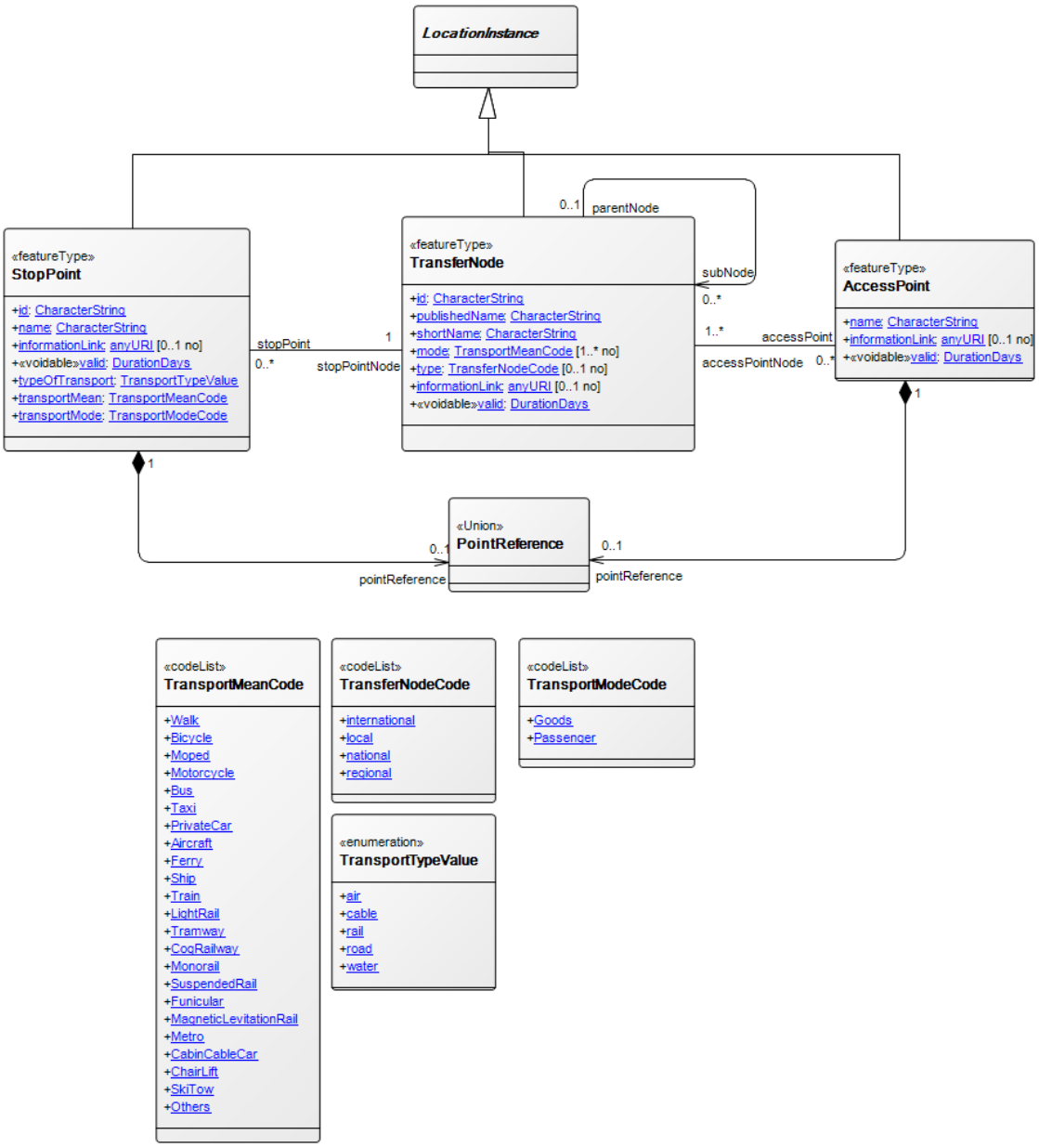


Figure 36 - Overview transfer node

The principal for a central register of transfer nodes, is to preferably only contain a minimum of information, since there naturally are several stakeholders involved. Each stakeholder should be able to connect their information in a clear manner to a central registry for it to be offered to different users in a uniform manner. In this document we deliberately keep the information content to a minimum, for the same reasons as for the respective transport types, and instead offer some simple connection mechanisms which can be used to allow various stakeholders to connect their information to the transfer nodes.

The different classes that are used to describe transfer nodes inherit from the class *LocationInstance* which means that they are position objects and thereby possible to use as targets when connecting features to the network. This also means that we have specifically added some classes to be able to specify specific network references to these particular classes.

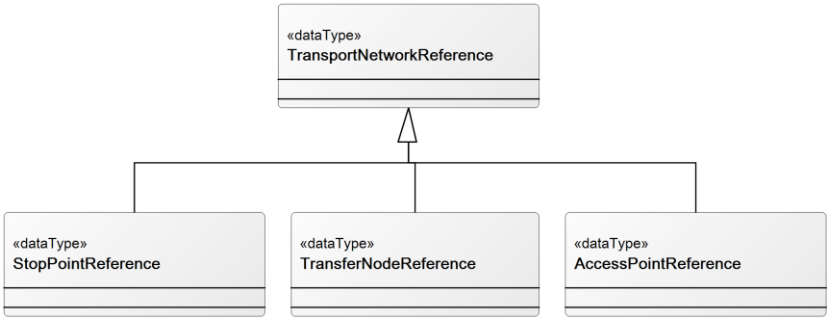


Figure 37 - Classes for specific network references towards transfer nodes

5.3.4.2 Class <<featureType>> TransferNode (ISO/TC211 - ISO 19147)

A *TransferNode* is a place that enables exchange between different transport means and transport types. It may include facilities and services for users of the node, such as information services, ticket sales, etc.Information on restrictions (e.g. for access) and abnormalities (i.e. a service is not operational) can be linked to a transfer node. A transfer node can have one or many access points (*AccessPoint*).

A transfer node can consist of other transfer nodes. This means that there may be hierarchies of transfer nodes and a transfer node can be related to one or several transport types or transport means. A transfer node contains at least one stop point where each stop point only belongs to one transport mean. Stop points are the locations where vehicles can stop and where people can step on/off or cargo can be loaded on/off. The physical location of a transfer node and its stop points may overlap with other transfer nodes and other stop points.

5.3.4.2.1 Attribute

Name	Type	Description
id (ISO/TC211 - ISO 19147)	CharacterString (ISO/TC211 - ISO/TS 19103)	Identity of the transfer node
publishedName (ISO/TC211 - ISO 19147)	CharacterString (ISO/TC211 - ISO/TS 19103)	Official name for the transfer node, e.g. "Arlanda airport”
shortName (ISO/TC211 - ISO 19147)	CharacterString (ISO/TC211 - ISO/TS 19103)	Short name for the transfer node, e.g. "ARN”.
mode [1..*] (ISO/TC211 - ISO 19147)	TransportModeCode (ISO/TC211 - ISO 19147)	Indicates the transport modes that are served by the transfer node
type [0..1] (ISO/TC211 - ISO 19147)	TransferNodeCode	Specifies the type of transfer node

informationLink [0..1] (ISO/TC211 - ISO 19147)	anyURI	Indicates a link to additional information about the transfer node
valid	DurationDays	Specifies the time period when the transfer node is valid

5.3.4.2.2 Relations

Name	Type	Description
stopPoint [0..*] (ISO/TC211 - ISO 19147)	StopPoint	The set of stop points for the transfer node
accessPoint [0..*] (ISO/TC211 - ISO 19147)	AccessPoint	The set of access points for the transfer node *Note:* *The standard requires at least one access point for each transfer node. From a reality perspective it is logical. Within the frame of this information model the focus does not lie on access points, which means that we have removed this requirement and allow transfer nodes without access points.*
subNode [0..*] (ISO/TC211 - ISO 19147)	TransferNode	Sub nodes to the current transfer node. *Note:* *Provides an opportunity for description of hierarchies.* *Example:* *Arlanda Airport consists of the sub nodes terminal 1-5 and Arlanda C (train station)*

		which in turn are transfer nodes.
parentNode [0..1] (ISO/TC211 - ISO 19147)	TransferNode	Inverse to the subNode above i.e. specifies an eventual superior transfer node in which the current transfer node is included.

5.3.4.3 Class <<featureType>> StopPoint (ISO/TC211 - ISO 19147)

A *StopPoint* describes a location at a transfer node where vehicles can stop. A stop point can only belong to one transfer node and one transport mean. The physical location of a stop point may overlap with other stop points. It is only transfer nodes with stop points that can serve transport means.

5.3.4.3.1 Attribute

Name	Type	Description
id (ISO/TC211 - ISO 19147)	CharacterString (ISO/TC211 - ISO/TS 19103)	Identity of the stop point
name (ISO/TC211 - ISO 19147)	CharacterString (ISO/TC211 - ISO/TS 19103)	Name of the stop point
informationLink [0..1] (ISO/TC211 - ISO 19147)	anyURI	Indicates a link to additional information on the stop point
typeOfTransport	TransportTypeValue (INSPIRE DS TN)	Specifies the transport type that the stop point belongs to
transportMean	TransportMeanCode (ISO/TC211 - ISO 19147) – TransportModeCode	Specifies the transport mean that is served by the stop point *Note:* *This is not included in the standard right now, but if you read the definition, a stop point should only serve one transport mean, which*

		indicates that the data should be included here. *In the Standard the corresponding property is called mode and is of the type TransportModeCode. We are of the opinion that the class actually describes transport mean and not transport mode.*
transportMode	TransportModeCode	Specifies the transport mode (goods or persons) that is served by the stop point.
valid	DurationDays	Specifies the period of time during which the stop point is valid

5.3.4.3.2 Relations

Name	Type	Description
pointReference [1..1]	PointReference	The network location for the stop point

5.3.4.4 Class <<featureType>> AccessPoint (ISO/TC211 - ISO 19147)

An *AccessPoint* (access point) describes a location that provides access to one or more transfer nodes.

5.3.4.4.1 Attribute

Name	Type	Description
name (ISO/TC211 - ISO 19147)	CharacterString (ISO/TC211 - ISO/TS 19103)	Name of the access point
informationLink [0..1] (ISO/TC211 - ISO 19147)	anyURI	Indicates a link to further information about the access point
valid	DurationDays	Specifies the period of time during which the access point is valid

5.3.4.4.2 Relations

Name	Type	Description
pointReference [1..1]	PointReference	The network location for the access point

5.3.4.5 Class <<dataType>> StopPointReference

A reference to a stop point (*StopPoint*).

Restrictions:

locationInstance shall be of type StopPoint.

```
Context StopPointReference inv:

locationInstance.oclIsKindOf(StopPoint)
```

The transport types indication of the stop point must comply with the transport types indication of the network element, for which the stop point's network reference refers to.

```
Context StopPointReference inv:

pointReference.locationInstance.oclAsType(TransportNetworkElement).inNetwork.typeOfTransport =
self.typeOfTransport
```

5.3.4.6 Class <<dataType>> TransferNodeReference

A reference to a transfer node (*TransferNode*).

Restriction:

locationInstance shall be of type TransferNode.

```
Context TransferNodeReference inv:

locationInstance.oclIsKindOf(TransferNode)
```

5.3.4.7 Class <<dataType>> AccessPointReference

A reference to an access point (*AccessPoint*).

Restriction:

locationInstance shall be of type AccessPoint.

```
Context AccessPointReference inv:

locationInstance.oclIsKindOf(AccessPoint)
```

5.3.4.8 Class <<codeList>> TransferNodeCode (ISO/TC211 - ISO 19147)

An enumeration of the possible types of transfer nodes available with respect to the type of transition or exchange that is possible.

5.3.4.8.1 Allowed values

Name	Description
international (ISO/TC211 - ISO 19147)	The transfer node offers transitions or exchanges towardsinternational routes
national (ISO/TC211 - ISO 19147)	The transfer node offers transitions or exchanges between national routes
regional (ISO/TC211 - ISO 19147)	The transfer node offers transitions or exchanges between regional routes
local (ISO/TC211 - ISO 19147)	The transfer node offers transitions or exchanges between local routes

5.3.4.9 Class <<codeList>> TransportMeanCode (ISO/TC211 - ISO 19147)

The class has an enumeration of the possible means of transportation available.

Note:

It is called TransportModeCode in (ISO/TC211 - ISO 19147)*. The standard is still under development which could entail changes.*

5.3.4.9.1 Allowed values

Name	Description
Walk (ISO/TC211 - ISO 19147)	Pedestrian
Bicycle (ISO/TC211 - ISO 19147)	Bicycle
Moped (ISO/TC211 - ISO 19147)	Moped/scooter
Motorcycle (ISO/TC211 - ISO 19147)	Motorcycle
Bus (ISO/TC211 - ISO 19147)	Bus
Taxi (ISO/TC211 - ISO 19147)	Taxi

PrivateCar (ISO/TC211 - ISO 19147)	Private car
Aircraft (ISO/TC211 - ISO 19147)	Means of transport for air transport
Ferry (ISO/TC211 - ISO 19147)	A vehicle for water transport of passengers or roll-on roll-off cargo such as automobiles
Ship (ISO/TC211 - ISO 19147)	A vehicle for water transport of passengers and cargo
Train (ISO/TC211 - ISO 19147)	Train for rail transport of goods or persons
LightRail (ISO/TC211 - ISO 19147)	Train similar means of transport that is limited to a rail network within a defined area
Tramway (ISO/TC211 - ISO 19147)	A railway transportation system in urban areas, which are usually found at street level and share the street area with other motorized traffic and pedestrians (tramway).
CogRailway (ISO/TC211 - ISO 19147)	A type of rail transport which allows the vehicle to operate in large inclines. Normally consists of toothed rack in the grooves on to which the gears of the vehicle can grip.
Monorail (ISO/TC211 - ISO 19147)	A type of rail transport that uses only one rail that fills the function of support and guidance.
SuspendedRail (ISO/TC211 - ISO 19147)	A type of rail transport over land where the vehicle is hanging in the rail (often only one rail).
Funicular (ISO/TC211 - ISO 19147)	A type of rail transport on rails where the vehicles are pulled

	up and down by the means of a cable.
MagneticLevitationRail (ISO/TC211 - ISO 19147)	A type of rail transport with only a rail that serves as guidance and support through magnetic levitation.
Metro (ISO/TC211 - ISO 19147)	A railway transportation system in urban areas and that are separated from other systems and often goes underground (subway/metro)
CabinCableCar (ISO/TC211 - ISO 19147)	A cable transport where the vehicle consists of a suspended cabin for carrying passengers or goods from one place to another
ChairLift (ISO/TC211 - ISO 19147)	A cable transport consisting of hanging chairs that carry passengers from one place to another
SkiTow (ISO/TC211 - ISO 19147)	A type of cable transport for pulling skiers and snowboarders up a mountain(ski tow/lift)
Others (ISO/TC211 - ISO 19147)	Other means of transport

5.3.4.10 Class <<codeList>> TransportModeCode

An enumeration of the possible transport modes that is available.

5.3.4.10.1 Allowed values

Name	Description
Goods	Goods
Passenger	Persons

5.3.5 Package – Road transport network

Since the basic model of the network and the network references also cover the basic needs for road networks, only a few classes are specified in this package that specialize the general classes that are already defined with regard to the transport type road. Most classes add

nothing beyond the type itself. These allow, for example, possibilities to define road specific restrictions in the model in a simple way and in the right location.

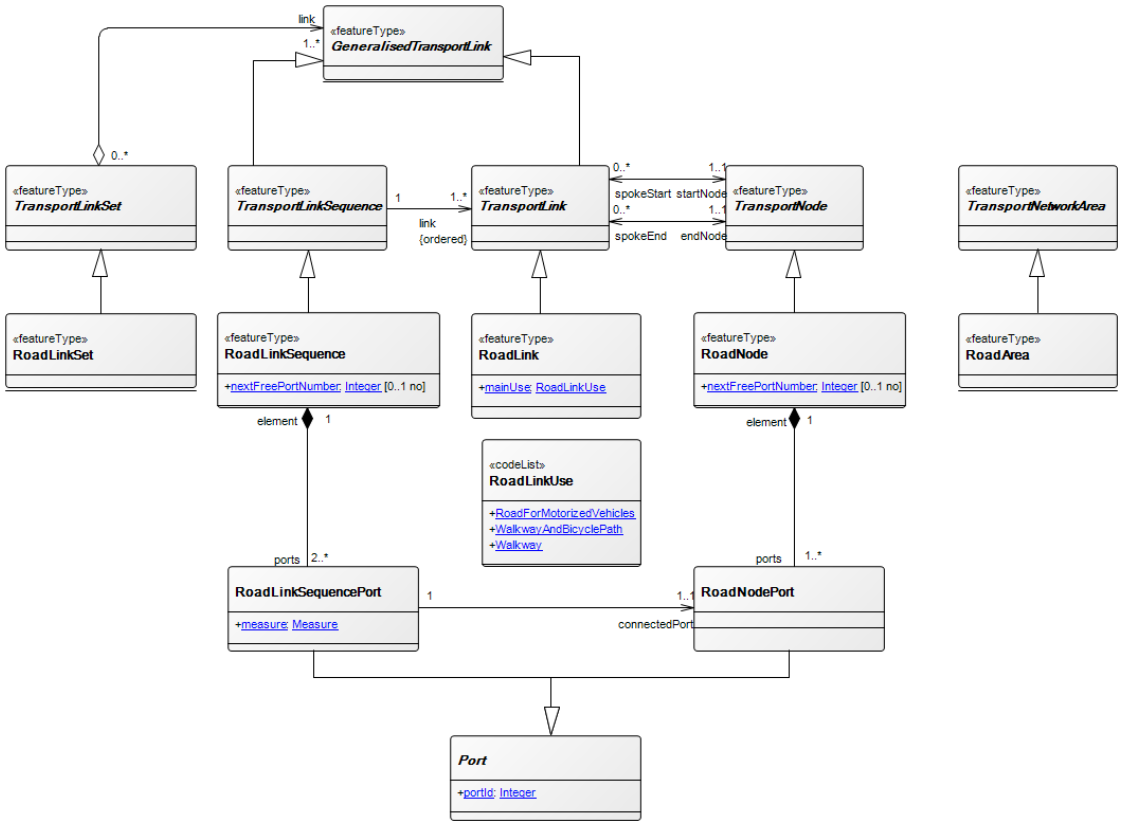


Figure 38 - Road transport network overview

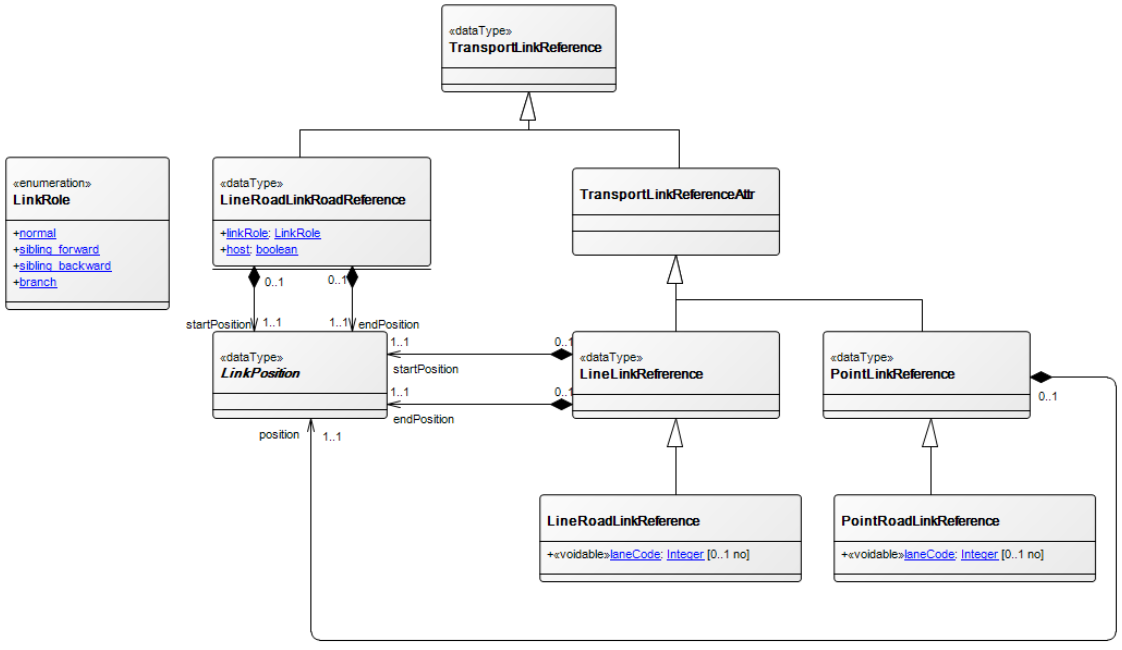


Figure 39 – Road specific network connection types

5.3.5.1 Class RoadLinkSet

Sub class to *TransportLinkSet* which describes a coherent and identifiable set of road links and road link sequences with a function in the road network.

Restriction:

The network in which the road link set is included shall have the transport type road.

```
Context RoadLinkSet inv:
inNetwork.typeOfTransport = road
```

5.3.5.2 Class RoadLinkSequence (INSPIRE DS TN)

The class is a subtype of TransportLinkSequence intended for road.

Note:

This corresponds to NW_RefLink i (SIS/TK320 - SS 637004, 2009).

Restrictions:

The network in which the road link sequence is included shall have the transport type road.

```
Context RoadLinkSequence inv:
inNetwork.typeOfTransport = road
```

5.3.5.2.1 Attribute

Name	Type	Description
nextFreePortNumber [0..1] (SIS/TK320 - SS 637004, 2009)	Integer	Sequential number that indicates the next available port number for current road link sequence. *Note:* *Available for compatibility reasons towards* (SIS/TK320 - SS 637004, 2009) *but serves no funciton otherwise.*

5.3.5.2.2 Relations

Name	Type	Description
ports [2..*] (SIS/TK320 - SS	RoadLinkSequencePort	The ports available for road link sequence (the reference link).

637004, 2009)		*Note:* *Available for compatibility reasons towards* (SIS/TK320 - SS 637004, 2009) *but serves nofunction otherwise. Note that the position must comply with the corresponding entry (startMeasure/endMeasure) for incoming road links.*

5.3.5.3 Class RoadLink (INSPIRE DS TN)

The class is a subtype of TransportLink intended for road.

Note:

This corresponds to today's reference link's part.

Restrictions:

The network in which the road link is included shall have the transport type road

```
Context RoadLink inv:
inNetwork.typeOfTransport = road
```

5.3.5.3.1 Attribute

Name	Type	Description
mainUse	RoadLinkUse	Specifies the main applications of the road link (road with motorized vehicles, pedestrians and cyclists or pedestrians).

5.3.5.4 Class RoadNode (INSPIRE DS TN)

The class is a subtype of TransportNode intended for road.

Restrictions:

The network in which the road node is included, shall have the transport type road

```
Context RoadNode inv:
inNetwork.typeOfTransport = road
```

5.3.5.4.1 Attribute

Name	Type	Description
nextFreePortNumber [0..1] (SIS/TK320 - SS 637004, 2009)	Integer	Sequential number that indicates the next available port number for the current road node. *Note:* *Available for compatibility reasons towards* (SIS/TK320 - SS 637004, 2009) *but serves no function otherwise.*

5.3.5.4.2 Relations

Name	Type	Description
ports [1..*] (SIS/TK320 - SS 637004, 2009)	RoadNodePort	The ports available for the road node. *Note:* *Available for backwards compatibility reasons, but serves no function otherwise.*

5.3.5.5 Class RoadArea (INSPIRE DS TN)

Sub class to *TransportNetworkArea* describing a surface with a function in the road network.

Restrictions:

The network in which the road surface is included, shall have the transport type road

```
Context RoadArea inv:
inNetwork.typeOfTransport = road
```

5.3.5.6 Class Port (SIS/TK320 - SS 637004, 2009)

An abstract base class that describes a coupling between the elements of a road network.

Note:

The class exists for backwards compatibility reasons with (SIS/TK320 - SS 637004, 2009).

5.3.5.6.1 Attribute

Name	Type	Description
portId (SIS/TK320 - SS 637004, 2009)	Integer	The port identity

5.3.5.7 Class RoadLinkSequencePort (SIS/TK320 - SS 637004, 2009)

A class describing a type of port that belongs to a road link sequence and is coupled together with a road node port.

Note:

The class exists for backwards compatibility reasons with (SIS/TK320 - SS 637004, 2009).

5.3.5.7.1 Attribute

Name	Type	Description
measure (SIS/TK320 - SS 637004, 2009)	Measure	The port location along the road link sequence must comply with the corresponding entry (startMeasure/endMeasure) for referenced road links *Restriction:* *Should be a value between 0 and 1.* *Context RoadLinkSequencePort inv:* *measure >= 0 and measure <= 1*

5.3.5.7.2 Relations

Name	Type	Description
connectedPort (SIS/TK320 - SS 637004, 2009)	RoadNodePort	Reference to the connected road node port. The node shall comply with the corresponding start or end node of the corresponding road link.

5.3.5.8 Class RoadNodePort (SIS/TK320 - SS 637004, 2009)

A class describing a type of port that belongs to a road node and to be connected to a port belonging to a road link sequence.

Note:

The class exists for backwards compatibility reasons with (SIS/TK320 - SS 637004, 2009).

5.3.5.9 Class <<dataType>> LineRoadLinkRoadReference (SIS/TK320 - SS 637004, 2009)

Specifies a linear location on a linear element in the road network (*GeneralisedTransportLink*) supplemented with the necessary attribute for road names/road numbers.

5.3.5.9.1 Attribute

Name	Type	Description
linkRole (SIS/TK320 - SS 637004, 2009)	LinkRole	Describes a road section role as a part of a road name/road number
host (SIS/TK320 - SS 637004, 2009)	Boolean	Specifies if the extent belongs to the host road for this road section

5.3.5.9.2 Relations

Name	Type	Description
startPosition (SIS/TK320 - SS 637004, 2009)	LinkPosition	Starting position
endPosition (SIS/TK320 - SS 637004, 2009)	LinkPosition	End position

5.3.5.10 Class <<enumeration>> LinkRole (SIS/TK320 - SS 637004, 2009)

The class specifies an enumeration of the possible values of the link role.

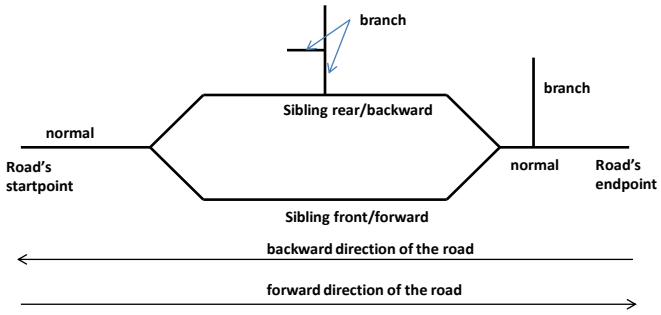


Figure 40 - Example of using the link role

Note:

The link role has its origin in the need to have an unambiguous description of a road's forward respectively backward direction. In calculating such lengths and amounts along the road's forward direction, all sections labeled "branch" and "sibling_back" can be neglected.

5.3.5.10.1 Allowed values

Name	Description
normal (SIS/TK320 - SS 637004, 2009)	Normal link
sibling_forward (SIS/TK320 - SS 637004, 2009)	Sibling forward
sibling_backward (SIS/TK320 - SS 637004, 2009)	Sibling backward
Branch (SIS/TK320 - SS 637004, 2009)	Branch

5.3.5.11 Class <<dataType>> PointRoadLinkReference (SIS/TK320 - SS 637004, 2009)

Indicates the point shaped location on a linear element in the network (*GeneralisedTransportLink*) supplemented with attribute for lane association.

5.3.5.11.1 Attribute

Name	Type	Description

<<voidable>> laneCode [0..1] (SIS/TK320 - SS 637004, 2009)	Integer	Specifies with a positive integer, the lane for which the connection is valid. *Restriction 1 :* *Lateral position shall be indicated together with laneCode.* *Context PointRoadLinkReference inv:* *laneCode->size() = 1 implies lateralPosition->size() = 1* *Restriction 2 :* *laneCode is a positive number* *Context PointRoadLinkReference inv:* *laneCode->size() = 1 implies laneCode > 0*

5.3.5.12 Class <<dataType>> LineRoadLinkReference (SIS/TK320 - SS 637004, 2009)

Indicates the distance shaped position on a linear element in the network (*GeneralisedTransportLink*) supplemented with attribute for lane association.

5.3.5.12.1 Attribute

Name	Type	Description
<<voidable>> laneCode [0..1] (SIS/TK320 - SS 637004, 2009)	Integer	Specifies the lane code for which the connection is valid. *Restriction 1 :* *Lateral position shall be indicated together with laneCode.* *Context LineRoadLinkReference inv:* *laneCode->size() = 1 implies lateralPosition->size() = 1* *Restriction 2 :*

		laneCode is a positive number *Context LineRoadLinkReference inv:* *laneCode->size() = 1 implies laneCode > 0*

5.3.5.13 Identification of lanes

In the classes PointRoadLinkReference and LineRoadLinkReference shown above, a lane can be identified. The purpose of this is to give the option to specify a specific lane when connecting features to road networks. This should be viewed in the same way as, for example, direction where data feature in its network reference can specify that it is only valid for a specified direction.

For lane identification, the rule applies that the lanes are numbered from one and upwards from the reference line (the linear element) and outward. To distinguish between the different sides of the road, the lane identification should be classified with an indication of the side (lateralPosition - left/right). This principle works regardless of the type of road. See the example below.

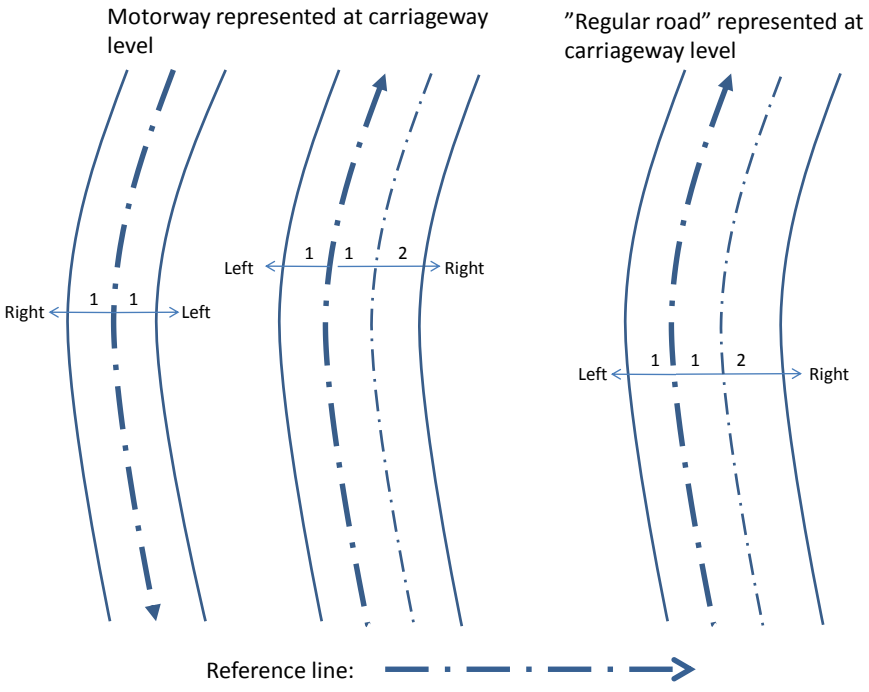


Figure 41 – Traffic lanes identification

5.3.5.14 Class <<codeList>> RoadLinkUse

The class specifies the possible values for the primary usage for road links.

Note:

Note that this is a code list which may be expanded with more entries, such as snowmobile trails.

5.3.5.14.1 Allowed values

Name	Description
RoadForMotorizedVehicles	Normal car road
WalkwayAndBicyclePath	Pedestrian and Cycle road
Walkway	Walkway

5.3.5.15 Other INSPIRE-requirements

INSPIRE has requirements on the following information when it comes to road networks. These should be treated as network connected features in the Swedish Transport Administration.

Name	Description
FormOfWay	Classification based on the road's physical characteristics: - BicycleRoad - DualCarriageway - EnclosedTrafficArea - EntranceOrExitCarPark - EntranceOrExitService - Freeway - Motorway - PedestrianZone - Roundabout - ServiceRoad - SingleCarriageway - SlipRoad - Tractor - TrafficSquare - Walkway
RoadWidth	Road width as an average value as well as which part of the road referred to.
SpeedLimit	Speed limit including specification of time, vehicles etc.
NumberOfLanes	Specifies the number of lanes
RoadName	Official name for the road, defined by the responsible authority.
RoadServiceType	Description of the type of RoadServiceArea: - BusStation - Parking - RestArea

	- Toll
RoadSurfaceCategory	Indicates whether a road is paved or not. - Paved - Unpaved
FunctionalRoadClass	A classification relating to the road's "importance" in the road network: - MainRoad - FirstClass - SecondClass - ThirdClass - FourthClass - FifthClass - SixthClass - SeventhClass - EighthClass - NinthClass

5.3.6 Package – Railway transport network

Since the basic model of the network and the network connection also cover the basic needs for the railway network, only a few classes are specified in this package that specialize the general classes that have are already been defined with regard to the transport type railway.

The most important additions to the railway networks in relation to a general transport network is the ability to use the kilometer posts and distance from those for positioning as well as the requirement to describe the possible ways to travel through nodes (switches). There is the possibility that in addition to the "normal" geometry inherited from the traffic network classes, to also describe schematic geometry on node and link level.

The schematic geometry is used primarily to make readable graphical representations of complex networks, such as operation station. The example below shows a simple example of this:

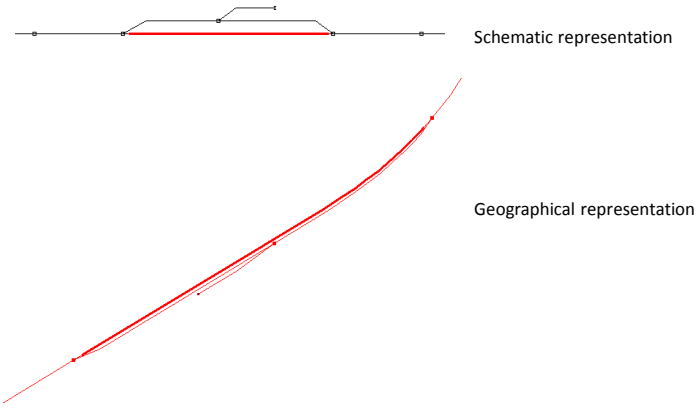


Figure 42 - Examples of schematic and geographical representation of the same network

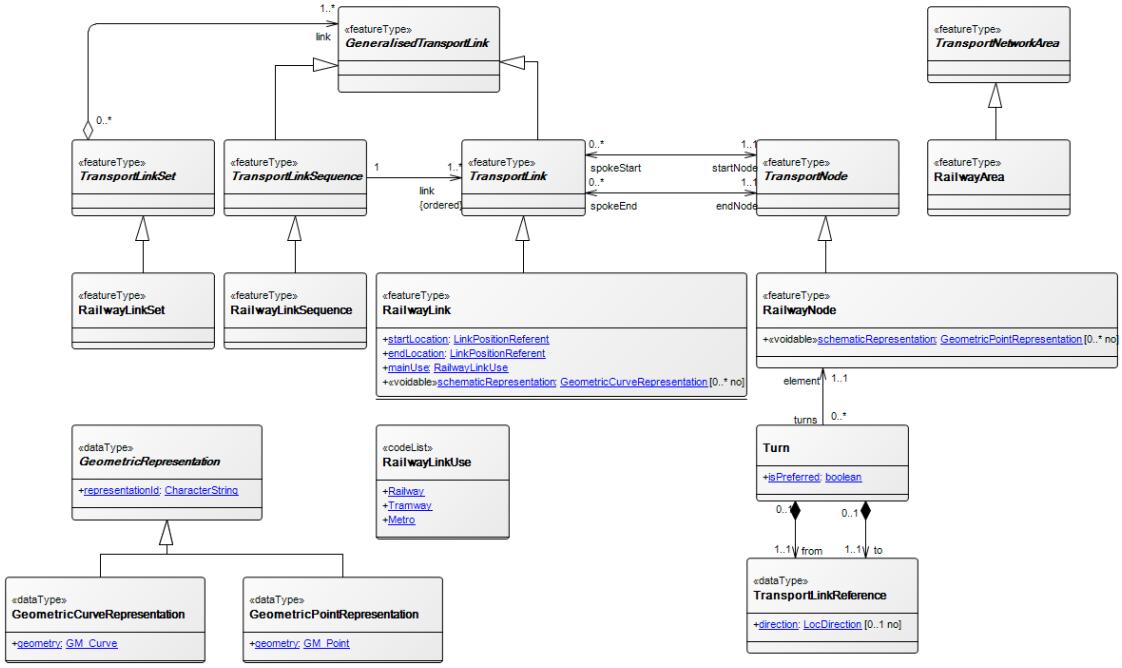


Figure 43 - Railway transport network overview

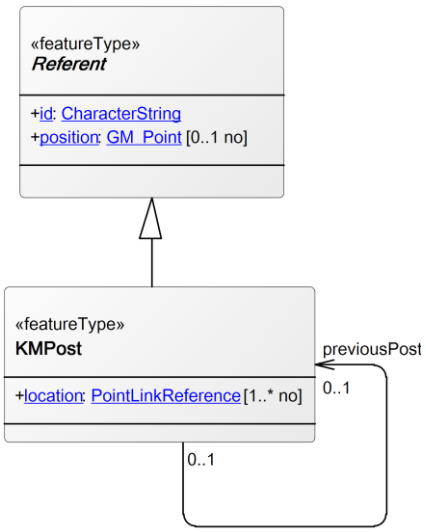


Figure 44 - KMPost - type of referent for kilometer posts

5.3.6.1 Class RailwayLinkSet

A subclass to *TransportLinkSet* which describes a coherent and identifiable set of rail links and rail link sequences with a function in the railway network.

Restrictions:

The network in which the railway link set is included shall have the transport type railway

```
Context RailwayLinkSet inv:
inNetwork.typeOfTransport = rail
```

5.3.6.2 Class RailwayLinkSequence (INSPIRE DS TN)

The class is a subtype of TransportLinkSequence intended for railway.

Restrictions:

The network in which the rail link sequence is included shall have the transport type railway

```
Context RailwayLinkSequence inv:
inNetwork.typeOfTransport = rail
```

5.3.6.3 Class RailwayLink (INSPIRE DS TN)

The class is a subtype of TransportLink intended for railway.

Restrictions:

The network in which railway link is included should have the transport type railway

```
Context RailwayLink inv:
inNetwork.typeOfTransport = rail
```

5.3.6.3.1 Attribute

Name	Type	Description
startLocation	LinkPositionReferent	Describes the railway link's starting position in relation to the kilometer system
endLocation	LinkPositionReferent	Describes the railway link's end position in relation to the kilometer system
mainUse	RailwayLinkUse	Specifies the primary usage for the link (railway, tramway or metro).
<<voidable>> schematicRepresentation [0..*]	GeometricCurveRepresentation	Describes schematic representations of the railway link

		intended for the presentation of such as operation station

5.3.6.4 Class RailwayNode (INSPIRE DS TN)

The class is a subtype of TransportNode intended for railway. Note that a railway node by default is NOT open for navigation. Each possibility to travel through the node is specified by a turn option, see the class *Turn* below.

Restrictions:

The network in which the railway node is included should have the transport type railway

```
Context RailwayNode inv:
inNetwork.typeOfTransport = rail
```

5.3.6.4.1 Attribute

Name	Type	Description
<<voidable>> schematicRepresentation [0..*]	GeometricPointRepresentation	Describes schematic representations of a railway node intended for the presentation of such as operation station

5.3.6.5 Class <<dataType>> GeometricRepresentation

An abstract base class for different types of geometrical representations.

5.3.6.5.1 Attribute

Name	Type	Description
representationId	CharacterString	Identifies a particular representation. Used to merge the representation of multiple objects to one image.

5.3.6.6 Class <<dataType>> GeometricCurveRepresentation

A class for geometrical line representation. It is a subclass to *GeometricRepresentation*.

5.3.6.6.1 Attribute

Name	Type	Description
Geometry	GM_Curve (ISO/TC211 - ISO 19107)	The geometry of the representation. *Restriction:* *GM_Curve allows many different types of geometries. Currently, we only allow geometry of the type LineString (i.e. breakpoints with straight lines between/polyline).* *Context GeometricCurveRepresentation inv:* *centrelineGeometry. segment->size() = 1 and centrelineGeometry. segment->forall(s | s. oclIsTypeOf(GM_LineString)*

5.3.6.7 Class <<dataType>> GeometricPointRepresentation

A class for geometrical point representation. It is a subclass to *GeometricRepresentation*.

5.3.6.7.1 Attribute

Name	Type	Description
Geometry	GM_Point (ISO/TC211 - ISO 19107)	The geometry of the representation

5.3.6.8 Class RailwayArea (INSPIRE DS TN)

A subclass to *TransportNetworkArea* describing an area with a function in railway network.

Restriction:

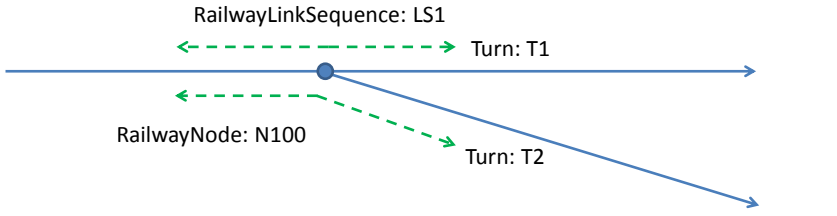
The network in which the railway surface is included should have the transport type railway

Context RailwayArea inv:
inNetwork.typeOfTransport = rail

5.3.6.9 Class Turn

The class describes a turn option for the railway, i.e. an possibility to travel through a railway switch. We always define a turn in a certain direction (from-via-to, just like turn extent, see chapter 5.3.3.8). Despite this, in this case, the turn is always bidirectional, i.e. it always applies both to the defined direction and in the reverse direction, because for possible travel through the switch there is no difference in the directions.

Below is an example of usage of the class *Turn* for a railway switch. Note that the only routes possible for traveling is according to turn T1 (which also is the preferred one) as well as turn T2. Any other way through the switch is not possible.



Turn : T1
{
element: N100,
from:
{
locationInstance: LS1,
direction: same
},
to:
{
locationInstance: LS1,
direction: same
},
isPreferred: true

Turn : T2
{
element: N100,
from:
{
locationInstance: LS1,
direction: same
},
to:
{
locationInstance: LS2,
direction: same
},
isPreferred: false

Figure 45 - Example of definition of turn (-possibility) in switches

5.3.6.9.1 Attribute

Name	Type	Description
isPreferred	Boolean	Indicates if the current turn is preferred for this railway node. *Restriction:* *The node shall only have one preferable turn .* *Context Turn inv:* *element.turns->select(t | t.isPreferred)->size() = 1*

5.3.6.9.2 Relations

Name	Type	Description
element	RailwayNode	Describes the railway node through which the turn goes
from	TransportLinkReference	Specifies from which network element you come, as well as direction in relation to that network element. *Restriction:* *Referenced position objects should belong to the railway network* *Context Turn inv:* *from.locationInstance.oclAsType(TransportNetworkElement).inNetwork.typeOfTransport = rail*
to	TransportLinkReference	Specifies to which network element you continue on, as well as the direction in relation to that network element. *Restriction:* *Referenced position objects should belong to the railway network* *Context Turn inv:* *to.locationInstance.oclAsType(TransportNetworkElement).inNetwork.typeOfTransport = rail*

5.3.6.10 Class <<featureType>> KMPost

The class describes a type of *Referent* corresponding to kilometer posts on railways. Objects of this class can be used as referents in connection with the location data with *LinkPositionReferent* to indicate the position relative to the kilometer reference system.

Kilometer posts are intended to be used together with the linear elements in the network (*RailwayLink/RailwayLinkSequence)* in the following way:

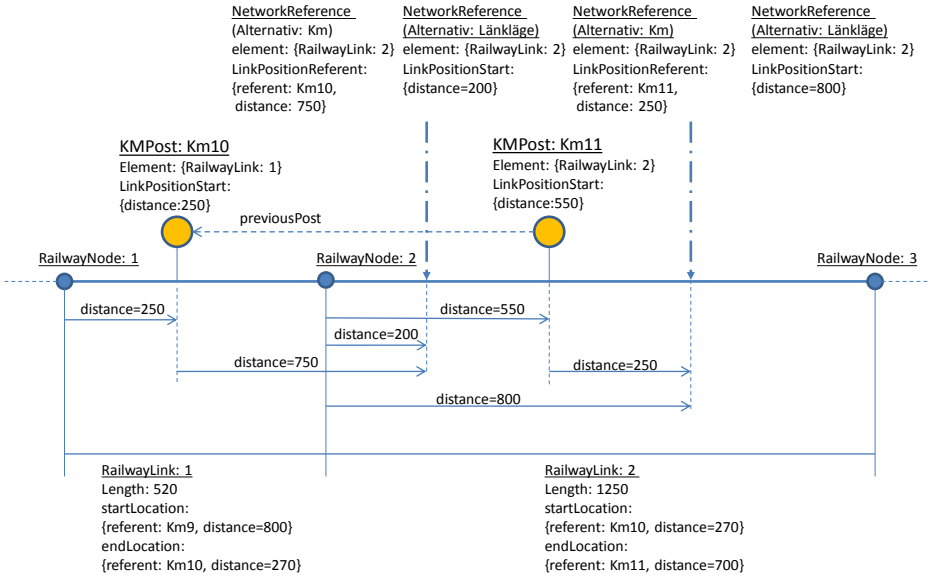


Figure 46 - Examples of using the kilometer reference system

- *The figure includes 2 links => RailwayLink: 1 (length=520m), RailwayLink: 2 (length=1250m)*
  - *The start and end positions of the links are described in relation to kilometer posts using LinkPositionReferent, i.e. reference to the kilometer post + distance.*
- *The figure contains 2 (+an invisible – Km9) kilometer posts => (Km9), Km10 and Km11. Handled as instance of the class KMPost*
  - *The kilometer posts' link positions are described by PointLinkReference (normal point reference), i.e. indication of the network element (link) as well as LinkPositionDist that indicates distance from the start node on the link where kilometer post exist.* ***Please note that each kilometer post has point reference against each link where it can be used (parallel tracks), hence a KMPost can have 1..* locations.***
- *The figure also contains examples of two different linear references (positions). These are shown by dotted arrows. These positions are indicated as both km and link-position located at the top of the figure*
- *To keep track of which type of position that is the origin of a position indication, it is required that you store the correct type of LinkPosition. You can always transform between the* ***three*** *existing possibilities (Link + distance, Link + measure, Km + distance). It may be beneficial to always store, for example, link + measure for all features to simplify analysis and compilations from a data storage perspective.*

5.3.6.10.1 Attribute

Name	Type	Description

location [1..*]	PointLinkReference	Location indications for the kilometer post in relation to the links that are connected to the current kilometer post. *Note:* *The reason why you can have multiple point locations is the need for one point location per track, in the case there are parallel tracks.*

5.3.6.10.2 Relations

Name	Type	Description
previousPost [0..1]	KMPost	Reference to previous kilometer post. *Note:* *If there is no previous kilometer post, this is excluded.*

5.3.6.11 Class <<codeList>> RailwayLinkUse

The class is an enumeration of the possible values for the primary usage of the track.

5.3.6.11.1 Allowed values

Name	Description
Railway	Railway
Tramway	Tramway
Metro	Metro/Subway

5.3.6.12 Other INSPIRE-requirements

INSPIRE requires the following information with regard to railway network. These or similar information should be treated as features at the Swedish Transport Administration.

Name	Description

NominalTrackGauge	The nominal distance between the two outer rails (gauge) of a railway track. - Measure - CategoryValue {broad, standard, narrow, notApplicable}
RailwayUse	The current use of the railway. - Cargo - carShuttle - mixed - passengers
DesignSpeed	The specification of the maximum speed to which a railway line is designed for (Velocity)
NumberOfTracks	The number of tracks for a railway distance.
RailwayStationCode	The unique code assigned to a railway station.
RailwayType	The type of railway transport the line is designed for. - cogRailway - funicular - magneticLevitation - metro - monorail - suspendedRail - train - tramway
RailwayElectrification	Indication whether the railway is provided with an electric system to power vehicles moving along it.

5.3.7 Management of network connected features (business data)

Network connected features are defined as all information that can be connected to the reference network, for example:

- Objects such as equipment associated with roads or railways, for example, signals, signs, animal fencing etc.
- Events that occurred somewhere in the network, such as accidents or events

- Characteristics that describe all or part of elements of the network, such as wearing course or railway switches/turnouts
- Administrative partitions such as railway line segments, road number or road manager
- Regulations such as bearing capacity and speed limits

The relationship between network connected features, feature catalog and reference network is illustrated at a less detailed level in Figure 5 Relationship between networks, operating data and geography.

The requirements for this information model for business data that is connected to the reference network, specifically express that the model for this should be "generic to provide ability to easily add, delete and change the types of features without any system development" and that "the model should be able to handle objects, events and partitions/attributes in the same way." With that as a starting point, the model in SS 637006 (SIS/TK323 - SS 637006, 2006) is governing when used for modeling of network linked features. In short, this means that the way the features are described in this model is generic and type independent. This is the way that many features have been handled traditionally both for road and rail network within the Swedish Transport Administration. This view is also consistent with the scope of the information model where explicit modeling of business specific features is not included. Therefore, the generic and type independent principles are also used in the proposed model in this report.

The following explains the basic principles of this view. More information can also be found, for example, in SS 637006 (SIS/TK323 - SS 637006, 2006).

5.3.7.1 Type independent feature model

In a type independent feature model, features of a specific type are not explicitly modeled; instead the same model is used regardless of what kind of feature is concerned. Structurally, it is, according to this model, no difference between a "speed limit" and a "railway switch". The advantage of this is that no systems or databases, that implements the generic feature model, need to be changed when new types of features are being added. This makes the model sustainable to changes of feature types and when adding future feature types, within a business. For example, the same set of tools for updating, presentation and analysis and processing may be used regardless of type of feature.

A model for features is reported in the chapter 5.3.9

5.3.7.2 Model for the feature catalog

The type independent model in itself is not sufficient to be able to interpret the data. The interpretation of the data requires a feature catalog that describes the various types of features that exist and that which is distinctive and descriptive for each feature type. Since the feature instance refers to the corresponding feature type in a feature catalog, the data is given a meaning. Note the following example:

Feature	**Feature catalog**

Feature, identity = abc123	Type = ”Speed limit” Description =”Traffic regulation concerning the maximum speed the vehicle may be driven, according to regulation.”
Attribute value = 30	Type = ” Maximum permitted speed” Unit = ”km/h” Data type = Enumeration with the following valid values : {20, 30 , 40, 50, 60, 70, 80, 90, 100, 110, 120}
Feature, identity = def456	Type = ” Road manager” Description = ” Whoever is responsible for road maintenance”
Attribute value = 2	Type = ” Road manager type” Data type = Enumeration with the following valid values : {1= governmental, 2= municipal, 3= private}
Attribute value = ”Borlänge Municipality”	Type = ” Road manager name” Data type = Character string
Feature, identity = 6870	Type = ”Signal” Description = ” Signal that provides optical message to drivers”
Attribute value = Hno	Type = ” Location signature” Data type = Character string
Attribute value = 1/6	Type = ”Signal no” Data type = Character string
Attribute value = >	Type = ”Direction validity” Description = ” With- Against- or Bi-directional signal, along the track length measurement” Data type = Enumeration with the following valid values :

	{< = Against direction, > = With direction, >< = Bi directional}
Attribute value = HS	Type = ”Type of signal” Data type = Enumeration with the following valid values: {FS = Pre signal (*Försignal*), HD = Main dwarf signal (*Huvuddvärgsignal*), HF = Combined main and pre signal (*Kombinerad Huvud & Försignal*), HS = Main signal (*Huvudsignal*), RFS = Repeater pre signal (*Repeterförsignal*), etc…}
Attribute value = M	Type = ”Type of main signal” Data type = Enumeration with the following valid values: {INF = Entrance signal (*Infartsignal*), M = Middle signal (*Mellansignal*), MBL = Middle block signal (*Mellanblocksignal*), UBL = Exit block signal (*Utfartsblocksignal*), etc…}
Attribute value = J	Type = ”Included in ATC” Data type = Enumeration with the following valid values : {J = Included in ATC, N = Not included in ATC }
Feature, identity = 60001	Type = ”Railway switch” Description = ” Device that allows for switching between tracks”
Attribute value = Hno	Type = ” Location signature” Data type = Character string
Attribute value = 1	Type = ”Switch no” Data type = Character string
Attribute value = EV-BV50-600-1:13	Type = ”Switch type” Data type = Enumeration with the following valid values: {EV-BV50-600-1:13, EVR-UIC60-2500-1:26,5, osv…}
Attribute value = H	Type = ”L/R-switch (*V/H-växel*)”

	Data type = Enumeration with the following valid values : {H = Right switch (*Högerväxel*), S = Symmetrical switch (*Symmetrisk växel*), V = Left switch (*Vänsterväxel*)}
Attribute value = 70	Type = ”STH Divergent” Description = ” Speed on diverging tracks” Data type = Numerical value, Min = 20, Max = 140

The left column may represent a number of features that are handled type independently. To interpret the data only based on this, one can only guess what the features actually represent. There is no principled distinction between the features because they all have identity and attribute values. By adding information from a feature catalog (right column) the data is given a meaning and can be interpreted. The point of doing it this way is that the structure of both features and feature catalog is uniform and can be reused for any number of feature types. Thus the feature catalog provides the knowledge and the possibility to reuse and combine existing feature types for new requirements.

A model for a feature catalog is reported in Chapter 5.3.8.

5.3.8 Package – Feature catalogue

The package contains a model for feature catalogs. A feature catalog contains definitions of feature types, attribute types, association types and value domains for a defined area of interest.

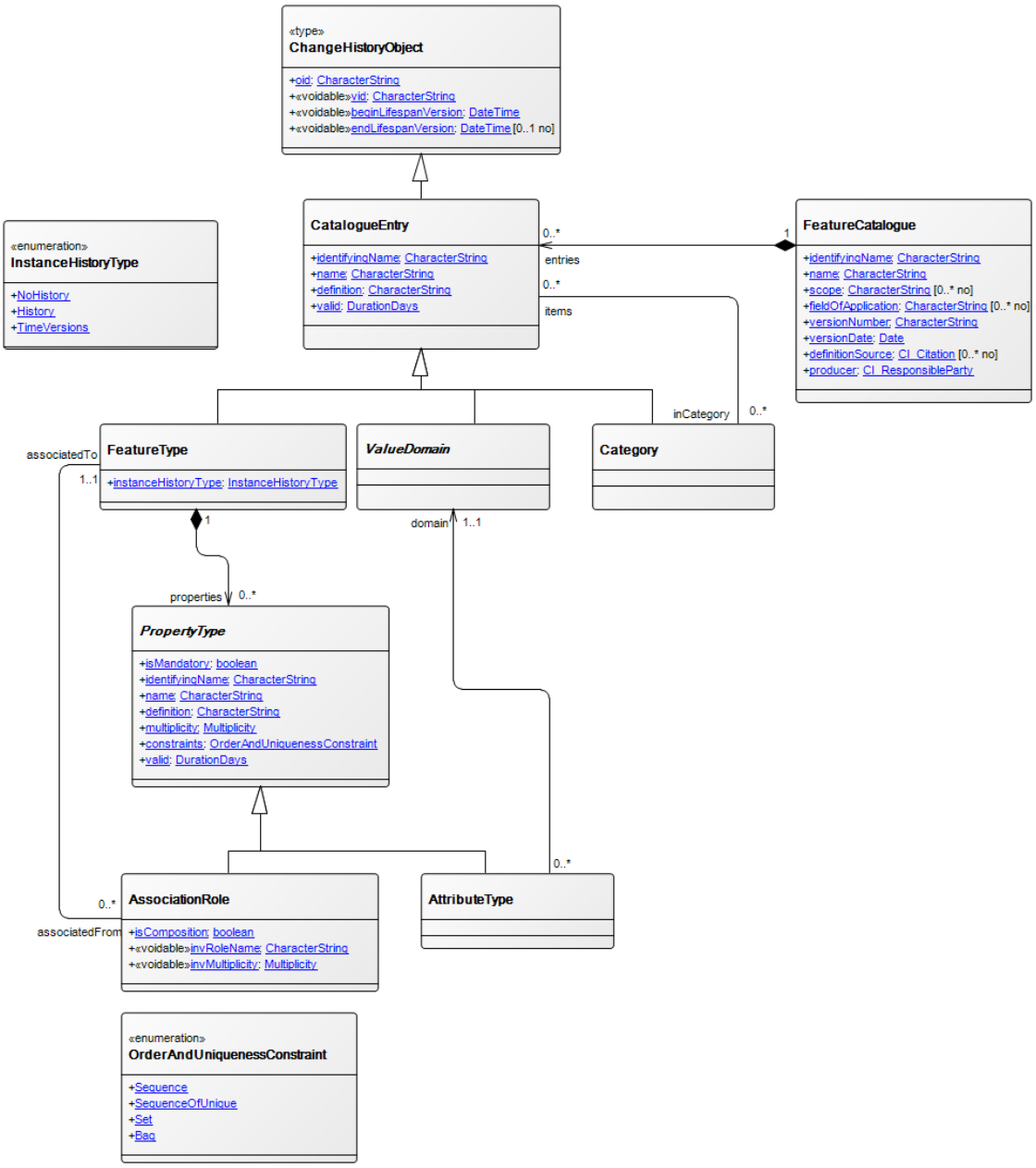


Figure 47 – Feature catalog overview

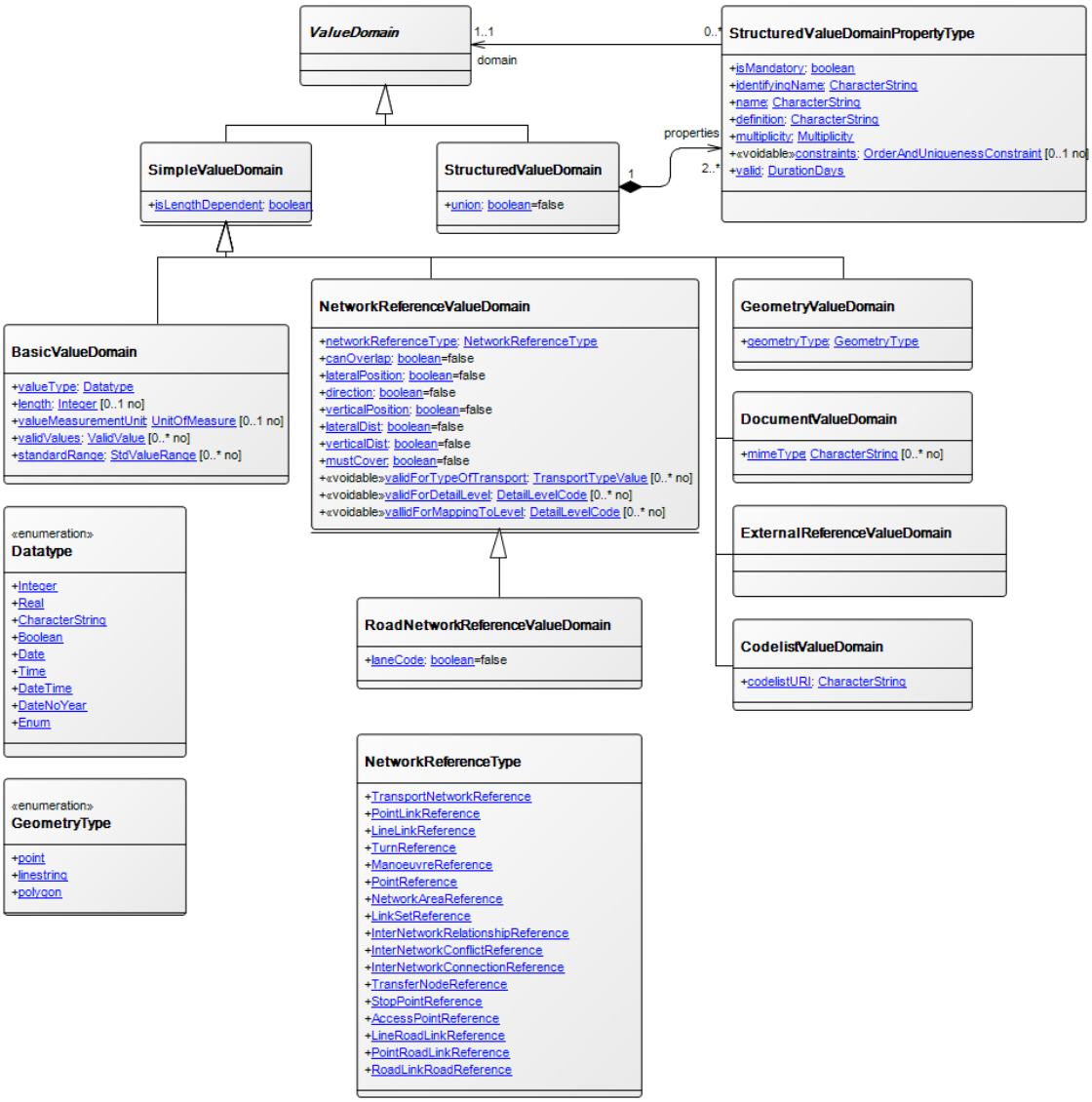


Figure 48 - ValueDomain overview

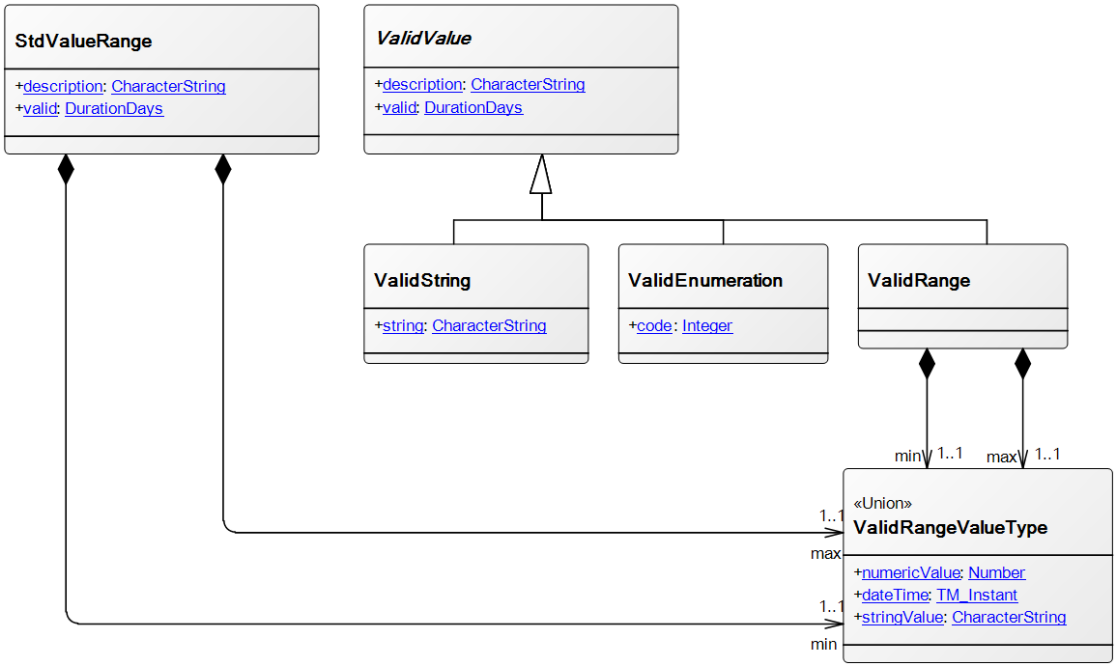


Figure 49 - Valid values and standard intervals

5.3.8.1 Class FeatureCatalogue (SIS/TK323 - SS 637006, 2006)

List of feature types, descriptive attribute types, connection types, and value domains within a defined area of interest.

Note:

Corresponds to FT_FeatureCatalogue i (SIS/TK323 - SS 637006, 2006)

5.3.8.1.1 Attribute

Name	Type	Description
identifyingName	CharacterString	Identifying name. *Note:* *The name should be set and used in such a way that the data catalog uniquely can be identified, and that the name can be directly used in technical systems.*
name	CharacterString	Descriptive name of the feature catalog
scope [0..*] (SIS/TK323 - SS 637006, 2006)	CharacterString	Area of interest for the feature types defined in the directory.

fieldOfApplication [0..*] (SIS/TK323 - SS 637006, 2006)	CharacterString	Describes what the feature catalog is intended to be used for
versionNumber (SIS/TK323 - SS 637006, 2006)	CharacterString	Version number for the feature catalog
versionDate (SIS/TK323 - SS 637006, 2006)	Date	Date for the validity of the feature catalog
definitionSource [0..*] (SIS/TK323 - SS 637006, 2006)	CI_Citation (ISO/TC211 - ISO/TS 19115)	Bibliographic reference to published external definition source for information in the feature catalog
Producer (SIS/TK323 - SS 637006, 2006)	CI_ResponsibleParty (ISO/TC211 - ISO/TS 19115)	Name and address of the person or organization that has primary responsibility for the contents of the feature catalog.

5.3.8.1.2 Relations

Name	Type	Description
entries [0..*] (SIS/TK323 - SS 637006, 2006)	CatalogueEntry	The set of entries in the feature catalog.

5.3.8.2 Class CatalogueEntry (SIS/TK323 - SS 637006, 2006)

An abstract base class. It represents an entry in the feature catalog.

Note:

Corresponds to FT_CatalogueEntry i (SIS/TK323 - SS 637006, 2006)

5.3.8.2.1 Attribute

Name	Type	Description
identifyingName	CharacterString	Identifying name. *Note:*

		The name should be set and used in such a way that the catalog entry is uniquely identified and that the name can be directly used in technical systems.
name	CharacterString	Descriptive name of the catalog entry.
definition (SIS/TK323 - SS 637006, 2006)	CharacterString	Definition or description of the catalog entry.
valid	DurationDays	The period of time during which the definition is valid.

5.3.8.2.2 Relations

Name	Type	Description
inCategory [0..*] (SIS/TK323 - SS 637006, 2006)	Category	The category or categories in which the catalog entry is included.

5.3.8.3 Class FeatureType (SIS/TK323 - SS 637006, 2006)

A subclass to CatalogueEntry. Feature type.

Note:

Corresponds to FT_FeatureType in (SIS/TK323 - SS 637006, 2006)

SS 637006 includes a possibility to specify inheritance relationships between feature types. This option is not currently included in this model because there is no clear requirement for that functionality. The assessment is that the inheritance relationship, even if it is a simple mechanism in terms of this model, adds a substantial complexity in the implementation of the model.

5.3.8.3.1 Attribute

Name	Type	Description
instanceHistory	InstanceHistoryType	Specifies which type of history that applies to instances of the feature type.

5.3.8.3.2 Relations

Name	Type	Description
properties [0..*] (SIS/TK323 - SS 637006, 2006)	PropertyType	Property types of the Feature type.
associatedFrom [0..*] (SIS/TK323 - SS 637006, 2006)	AssociationRole	The association roles from which this type is associated.

5.3.8.4 Class ValueDomain (SIS/TK323 - SS 637006, 2006)

A subclass to CatalogueEntry and an abstract base class. It is a catalog entry describing a value domain.

Note:

Corresponds to FT_ValueDomain in (SIS/TK323 - SS 637006, 2006)

5.3.8.5 Class Category (SIS/TK323 - SS 637006, 2006)

A subclass to CatalogueEntry. It is a catalog entry forming a grouping/categorization of other catalog entries.

Note:

Corresponds to FT_Category in (SIS/TK323 - SS 637006, 2006)

5.3.8.5.1 Relations

Name	Type	Description
items [0..*] (SIS/TK323 - SS 637006, 2006)	CatalogueEntry	The catalog entries included in the category.

5.3.8.6 Class PropertyType (SIS/TK323 - SS 637006, 2006)

An abstract base class that describes a type of property (attribute or association type) for a feature type.

Note:

Corresponds to FT_PropertyType i (SIS/TK323 - SS 637006, 2006)

Even in SS 637006 the FT_PropertyType is an abstract base class for attribute types and association types. However, there are some differences in the relationship to FT_FeatureType / Transport Property Type. In (SIS/TK323 - SS 637 006, 2006) the ownership relationship of the association types is unclear since neither feature types or data directory as a whole owns associations. In this model, it is the feature type that owns all property types, including

associations. Implications of this are further described in the chapter about associations, see 5.3.8.7.

5.3.8.6.1 Attribute

Name	Type	Description
isMandatory (SIS/TK323 - SS 637006, 2006)	Boolean	Indicates if the property type is mandatory for every instance of a particular feature type
identifyingName	CharacterString	Identifying name. *Note:* *The name should be set and used in such a way that the property type is uniquely identified and that the name can be directly used in technical systems.*
name (SIS/TK323 - SS 637006, 2006)	CharacterString	The name of the Property type
definition (SIS/TK323 - SS 637006, 2006)	CharacterString	The definition of the Property type
multiplicity (SIS/TK323 - SS 637006, 2006)	Multiplicity	Specifies the minimum and maximum number of occurrences of the property for an instance
constraints (SIS/TK323 - SS 637006, 2006)	OrderAndUniquenessConstraint	Specifies constraints regarding sequencing, or duplicates within the set of values (if more than one) of the property for an instance

valid	DurationDays	Specifies the time period during which the property type is valid and can be used

5.3.8.7 Class AssociationRole (SIS/TK323 - SS 637006, 2006)

It is a type of property type that describes an association with a feature type.

Note:

Corresponds to FT_AssociationType i (SIS/TK323 - SS 637006, 2006)

In (SIS/TK323 - SS 637 006, 2006) the ownership of the associations is unclear. The association is an independent bi-directional entity without a designated owner. In this model, each association is owned by a feature type. This means that for each association it needs to be decided which direction is the natural one for the association. The feature type, from which the association starts, shall be the owner of the association. This is also why the standard FT_AssociationType in this model is called Association Role, as the class actually describes the role that the associated feature type plays in relation to the feature type that owns the association.

An alternative approach is to model BOTH the association as a whole AND the association's roles in the respective end. In that case, the association as a whole would be owned by the feature catalog. The reason for not selecting this option is that it brings additional complexity to the model.

5.3.8.7.1 Attribute

Name	Type	Description
isComposition	Boolean	Specifies whether the association should be regarded as a composition, i.e. instances of the associated feature types shall share life span. *Note:* *The Swedish standard also uses aggregate as a type of association. In this model, for the sake of simplicity, we removed that possibility due to unclear semantics.*

<<voidable>> invRoleName [0..1] (SIS/TK323 - SS 637006, 2006)	CharacterString	Name of the association role in the inverted direction.
<<voidable>> invMultiplicity (SIS/TK323 - SS 637006, 2006)	Multiplicity	Indicates multiplicity in the inverted direction.

5.3.8.7.2 Relations

Name	Type	Description
associatedTo	TransportPropertyType	Refers to the associated feature type

5.3.8.8 Class AttributeType (SIS/TK323 - SS 637006, 2006)

A type of property type that describes an attribute for a particular feature type.

Note:

Corresponds to FT_AttributeType i (SIS/TK323 - SS 637006, 2006)

The difference between the association role and the attribute type is very subtle. The association role type describes a property type that forms an association with a different feature type, i.e. the instances of the feature type must contain a value that describes the association to other instances of the kind provided by the association type (e.g., a pointer or identity). The attribute type describes a property whose type is a value domain. The value domains possible to use are described by the class Value Domain and its subtypes.

5.3.8.9 Class <<enumeration>> OrderAndUniquenessConstraint (SIS/TK323 - SS 637006, 2006)

Enumeration of the possible restrictions regarding the sequencing and duplicates in the set of instances of property types (association types and attribute types).

Note:

The sequence only signifies that it is significant and has a meaning, and should be maintained. The definition or representation of the sequence is not described here, rather is defined case by case.

5.3.8.9.1 Attribute

Name	Description
Sequence	Sequence in which duplicates are allowed

SequenceOfUnique	Sequence in which duplicates are not allowed.
Bag	Unordered set in which duplicates are allowed
Set	Unordered set in which duplicates are not allowed

5.3.8.10 Class <<enumeration>> InstanceHistoryType (SIS/TK323 - SS 637006, 2006)

Defines an enumeration of the existing possibilities to specify restrictions regarding the managing history for instances of a particular feature type.

5.3.8.10.1 Attribute

Name	Description
NoHistory	The feature type instances describe no history. *Note:* *Means that the instances must be of the type FeatureInstanceWithoutHistory, see 5.3.9.2*
History	The feature type instances describe validity *Note:* *Means that the instances must be of the FeatureInstanceWithHistory, see 5.3.9.3*
TimeVersions	The feature type instances describe validity using time versions *Note:* *Means that the instances must be of the FeatureInstanceWithTimeVersions, see 5.3.9.4*

5.3.8.11 Class SimpleValueDomain (SIS/TK323 - SS 637006, 2006)

A type of value domain whose valid attribute values are simple and not composed of multiple attribute values. It is an abstract base class.

Note:

Corresponds to FT_SimpleValueDomain in (SIS/TK323 - SS 637006, 2006)

The subtypes of FT_SimpleValueDomain in (SIS/TK323 - SS 637006, 2006) which are *NOT handled explicitly in this model is FT_LocationalValueDomain, FT_MetaDataValueDomain and FT_TemporalValueDomain. The reason for this is that they add complexity to the model when*

defining a feature catalog, and that they may be represented through use of other types of value domains.

5.3.8.11.1 Attribute

Name	Type	Description
isLengthDependent	Boolean	Specifies whether the value of the domain is length dependent. *Note:* *Length Dependent value domains cannot easily be used in, for example, segmentations of the network (further breakdown of network elements with respect to different attribute values) because they relate to a specific distance. See the example below.*

Examples of the length-dependent attribute values:

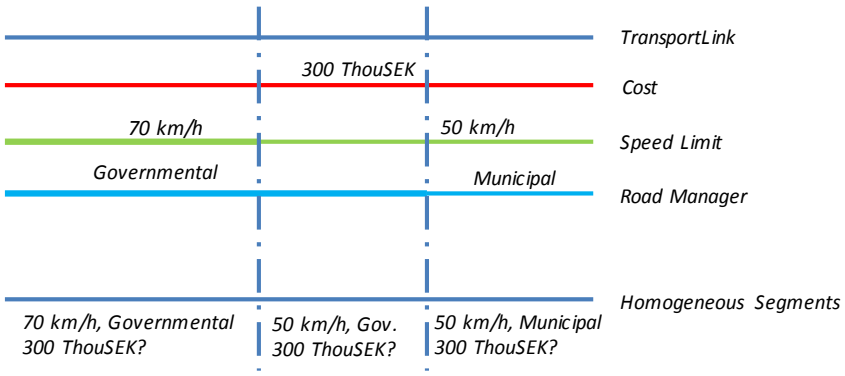


Figure 50 - Examples of the length-dependent attribute values

In the above example a link is described as divided into homogeneous segments based on three different attribute values. For speed limits and road managers it is not a problem that the data is applied for each sub-segment. However, it is certainly not a desirable result that the cost 300 thousand Swedish crowns is placed on each of the sub-segments. One might then be led to believe that the total cost of the route is 900 thousand Swedish crowns, which is not the case. A

more correct result is probably to make an intelligent breakdown of the cost per section of the route. The information model does not provide support for in which way such a division shall take place, but only gives an indication that the attribute is of the length-dependent type.

5.3.8.12 Class BasicValueDomain

Type of a simple value domain whose valid attribute values has a basic data type (possible data types are described by the enumeration Datatype).

Note:

It corresponds to FT_ThematicValueDomain in (SIS/TK323 - SS 637006, 2006).

5.3.8.12.1 Attribute

Name	Type	Description
valueType	Datatype	Specifies the data type of the value domain.
length[0..1]	Integer	Specifies the string length in the case that the data type is a character string (valueType=CharacterString)
valueMeasurementUnit[0..1]	UnitOfMeasure (ISO/TC211 - ISO/TS 19103)	Unit for attribute value, such as "km/h" or "m"
validValues [0..*]	ValidValue	Permitted set of values in the value domain
standardRange [0..*]	StdValueRange	Named ranges of values within the value domain.

5.3.8.13 Class NetworkReferenceValueDomain

Type of a simple value domain whose valid attributes values are of a type of network reference (*NetworkReference*).

Note:

Method to define a feature type's valid network reference type. It corresponds most closely to NW_ExtentValueDomain in (SIS/TK320 - SS 637004, 2009).

5.3.8.13.1 Attribute

Name	Type	Description
networkReferenceType	NetworkReferenceType	Specifies the type of network reference.

canOverlap (SIS/TK320 - SS 637004, 2009)	Boolean=False	Specifies the concurrency of the feature type, i.e. if the network references of feature instances (of current type) are allowed to overlap with network references of this or other feature instances (of the same type) in time and space. *See separate description regarding concurrency below.*
lateralPosition (SIS/TK320 - SS 637004, 2009)	Boolean=false	Specifies whether the lateral position of the network reference shall be stated
direction (SIS/TK320 - SS 637004, 2009)	Boolean=false	Specifies whether the network references of the feature instances are direction dependent (the direction shall be specified for network reference)
verticalPosition (SIS/TK320 - SS 637004, 2009)	Boolean=false	Specifies whether vertical position shall be specified for network references
lateralDist (SIS/TK320 - SS 637004, 2009)	Boolean=false	Specifies whether lateral distance shall be specified for network references
verticalDist (SIS/TK320 - SS 637004, 2009)	Boolean=false	Specifies whether vertical distance shall be specified for network references

mustCover (SIS/TK320 - SS 637004, 2009)	Boolean=false	Specifies whether the network references of the feature type, in total shall be comprehensive, i.e. cover all parts of a given network.
<<voidable>> validForTypeOfTransport [0..*]	TransportTypeValue	Specifies restrictions regarding the transport types for the network elements that the network references are allowed to be connected to. If no restriction is specified, the network references can be connected regardless of transport type
<<voidable>> validForDetailLevel [0..*]	DetailLevelCode	Specifies restrictions on the levels of detail of the network elements that network references are allowed to be connected to. If no restriction is specified, the network references can be connected regardless of level of detail.
<<voidable>> validForMappingToLevel [0..*]	DetailLevelCode	Specifies restrictions regarding which other levels of detail the network references may be reported on. If no restriction is specified, network references may be reported at all levels

		of detail that has a defined mapping to current level. *Note:* *This means that the information in class Transport Systems is being used (see 5.3.2.14) which describes how network elements at different levels of detail relate to each other, to account for the location of a feature instance on a different level than the level at which the location has been defined.* *This means that a feature may have a defined position in relation to a link, for example carriageway level. Through the information in the TransportSystem, the same feature is reported in relation to the corresponding location on the link that corresponds to the carriageway link at road level.*

For two network connected features to be considered to be overlapping (controlled by attribute *canOverlap*), the following applies:

- The spatial location of the network references should overlap completely or partially
    - The spatial location of the network references is described by

        - Referenced network elements (two network references on different network elements do not overlap)
        - Position on the network element (tolerance/resolution must be defined)
        - Direction in relation to the network element (two network references on the same part of the network but in different directions do not overlap)
        - Lateral position in relation to the network element (two network references on the same part of the network but on different sides do not overlap)
        - Lateral distance in relation to the network element (two network references on the same part of the network but at different lateral distances do not overlap)
        - Vertical position in relation to the network element (two network references on the same part of the network but at different vertical position do not overlap)
        - Vertical distance from the network element (two network reference on the same part of the network but with different vertical distance does not overlap)
        - Lanes (two network references on the same part of the network but in different lanes do not overlap)

    - The network references positions in time shall overlap completely or partially (they should co-exist at some point in time)
        - The validity must overlap, i.e. given two network references u1 and u2 then the following holds (note that the validity of the network references are retrieved from the feature instance whose location is described by the network references and that validTo is not included in the specified time interval)
            - *not(u1.validTo <= u2.validFrom or u1.validFrom >= u2.validTo)*

Example:

The figure below shows a number of examples of concurrent and non-concurrent network references. The network references are shown as dashed red lines and the network element (the link) is the solid blue line. The example uses spatial location, direction and time, but the same principle would apply to lateral position, lateral distance, lane, etc.

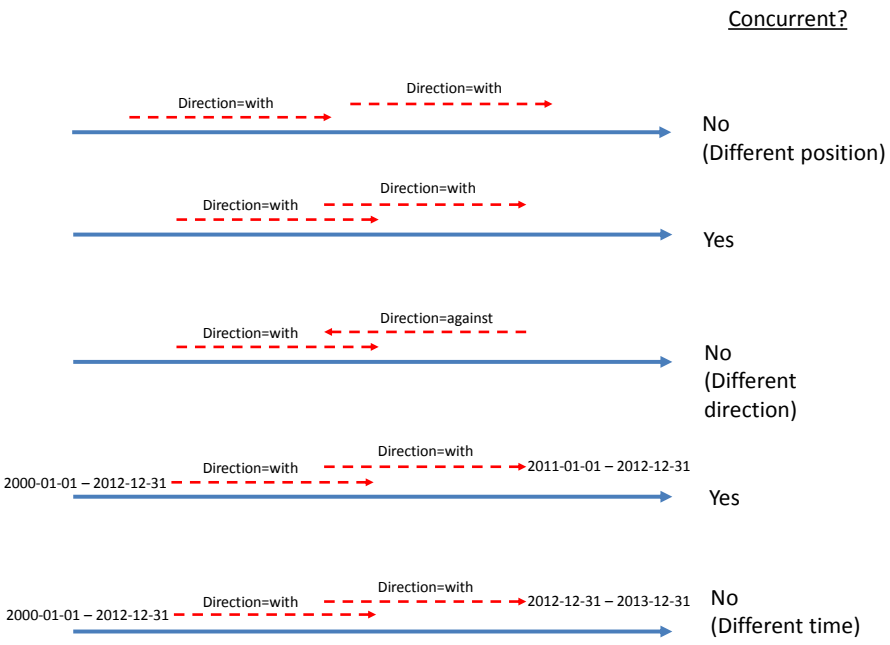


Figure 51 - Examples of concurrency

5.3.8.14 Class RoadNetworkReferenceValueDomain

Subtype to *NetworkReferenceValueDomain* for description of network reference type for road. Adds the ability to specify whether the network references could be connected to lanes or not.

5.3.8.14.1 Attribute

Name	Type	Description
laneCode (SIS/TK320 - SS 637004, 2009)	Boolean=false	Specifies whether the lane code shall be provided for network references

5.3.8.15 Class <<enumeration>> NetworkReferenceType

It is a class that enumerates valid network reference types.

5.3.8.15.1 Attribute

Name	Description
TransportNetworkReference	Network references can be of arbitrary type
PointLinkReference	The network references shall be of the type *PointLinkReference*
LineLinkReference	The network references shall be of the type *LineLinkReference*

TurnReference	The network references shall be of the type *TurnReference*
ManoeuvreReference	The network references shall be of the type *ManoeuvreReference*
PointReference	The network references shall be of the type *PointReference*
NetworkAreaReference	The network references shall be of the type *NetworkAreaReference*
LinkSetReference	The network references shall be of the type *LinkSetReference*
InterNetworkRelationshipReference	Network references can be of arbitrary subtype of *InterNetworkRelationship*
InterNetworkConflictReference	The network references shall be of the type *InterNetworkConflictReference*
InterNetworkConnectionReference	The network references shall be of the type *InterNetworkConnectionReference*
TransferNodeReference	The network references shall be of the type *TransferNodeReference*
StopPointReference	The network references shall be of the type *StopPointReference*
AccessPointReference	The network references shall be of the type *AccessPointReference*
PointRoadLinkReference	The network references shall be of the type *PointRoadLinkReference*
LineRoadLinkReference	The network references shall be of the type *LineRoadLinkReference*

RoadLinkRoadReference	The network references shall be of the type *RoadLinkRoadReference*

5.3.8.16 Class GeometryValueDomain

Type of a simple value domain whose valid attributes values are of a geometric type.

Note:

Corresponds most closely to FT_SpatialValueDomain in (SIS/TK323 - SS 637006, 2006)

5.3.8.16.1 Attribute

Name	Type	Description
geometryType	GeometryType	Specifies type of geometry.

5.3.8.17 Class <<enumeration>> GeometryType

It is a class with enumeration of valid geometry types.

5.3.8.17.1 Attribute

Name	Description
Point	The geometry shall be of type GM_*Point*
Linestring	The geometry shall be of type *GM_Curve (GM_Linestring)*
Polygon	The geometry shall be of type *GM_Surface (GM_Polygon)*

5.3.8.18 Class DocumentValueDomain

Type of a simple value domain whose values is of a document type.

5.3.8.18.1 Attribute

Name	Type	Description
mimeType [0..*]	CharacterString	Specifies restrictions on the types of documents that are allowed. Should (unless otherwise stated) follow IANA MIME Media Types (http://www.iana.org/assignments/media-types/index.html) *Example:*

		.docx – "application/vnd.openxmlformats-officedocument.wordprocessingml.document" *.jpg – "image/jpeg*

5.3.8.19 Class ExternalReferenceValueDomain

Type of simple value domain whose valid attribute values are of the type external reference, typically a reference to a resource that is available on the Web.

5.3.8.20 Class CodelistValueDomain

Type of simple value domain whose value valid attribute values are defined by an external code list.

5.3.8.20.1 Attribute

Name	Type	Description
codelistURI	CharacterString	A URI (Uniform Resource Identifier) which unambiguously identify the valid code list

5.3.8.21 Class StdValueRange (SIS/TK323 - SS 637006, 2006)

A class that defines a range of names of valid attribute values within a value domain.

Note:

Corresponds to FT_StdValueRange in (SIS/TK323 - SS 637006, 2006)

5.3.8.21.1 Attribute

Name	Type	Description
description (SIS/TK323 - SS 637006, 2006)	CharacterString	Describes/names the range
valid	DurationDays	Specifies the time period during which the definition is valid

5.3.8.21.2 Relations

Name	Type	Description
min (SIS/TK323 - SS 637006, 2006)	ValidRangeValueType	The lowest valid value within a standard range

max (SIS/TK323 - SS 637006, 2006)	ValidRangeValueType	The highest valid value within a standard range

5.3.8.22 Class ValidValue (SIS/TK323 - SS 637006, 2006)

An abstract base class that defines the allowable values in a value domain.

Note:

Corresponds to FT_ValidValue in (SIS/TK323 - SS 637006, 2006)

5.3.8.22.1 Attribute

Name	Type	Description
Description (SIS/TK323 - SS 637006, 2006)	CharacterString	Describes/names the meaning of the value
valid	DurationDays	Specifies the time period during which the definition is valid

5.3.8.23 Class ValidString (SIS/TK323 - SS 637006, 2006)

A type of *ValidValue* defining the permitted string value within a value domain.

Note:

Corresponds to FT_ValidString in (SIS/TK323 - SS 637006, 2006)

5.3.8.23.1 Attribute

Name	Type	Description
string (SIS/TK323 - SS 637006, 2006)	CharacterString	The valid character string

5.3.8.24 Class ValidEnumeration (SIS/TK323 - SS 637006, 2006)

A type of *ValidValue* defining the permitted value of the type integer within a value domain.

Note:

Corresponds to FT_ValidEnumeration in (SIS/TK323 - SS 637006, 2006)

5.3.8.24.1 Attribute

Name	Type	Description
code (SIS/TK323 - SS 637006, 2006)	Integer	The valid integer

5.3.8.25 Class ValidRange (SIS/TK323 - SS 637006, 2006)

A type of *ValidValue* defining the permitted value range in a value domain.

Note:

Corresponds to FT_ValidRange in (SIS/TK323 - SS 637006, 2006)

5.3.8.25.1 Relations

Name	Type	Description
min (SIS/TK323 - SS 637006, 2006)	ValidRangeValueType	The lowest value within the range
max (SIS/TK323 - SS 637006, 2006)	ValidRangeValueType	The highest value within the range

5.3.8.26 Class <<union>> ValidRangeValueType (SIS/TK323 - SS 637006, 2006)

A value type of the values in a valid range or standard range.

Note:

Corresponds to FT_ValidRangeValueType in (SIS/TK323 - SS 637006, 2006)

5.3.8.26.1 Attribute

Name	Type	Description
numericValue (SIS/TK323 - SS 637006, 2006)	Number	numeric value
dateTime (SIS/TK323 - SS 637006, 2006)	TM_Instant (ISO/TC211 - ISO 19108)	Time entry

stringValue (SIS/TK323 - SS 637006, 2006)	CharacterString	Character string

5.3.8.27 Class StructuredValueDomain (SIS/TK323 - SS 637006, 2006)

A value domain that is composed of other value domains.

Note:

Corresponds to FT_StructuredValueDomain in (SIS/TK323 - SS 637006, 2006)

5.3.8.27.1 Attribute

Name	Type	Description
union (SIS/TK323 - SS 637006, 2006)	Boolean=false	Specifies whether only one of the structure's values is allowed to be specified for each single instance.

5.3.8.27.2 Relations

Name	Type	Description
members (SIS/TK323 - SS 637006, 2006)	StructuredValueDomainPropertyType	Value domains included in a structured value domain.

5.3.8.28 Class StructuredValueDomainPropertyType

A single property type within a structured value domain.

Note:

Corresponds to FT_StructuredValueDomainMember in (SIS/TK323 - SS 637006, 2006) *but since the concept corresponds to Property Type, this name is instead also used here.*

5.3.8.28.1 Attribute

Name	Type	Description
isMandatory (SIS/TK323 - SS 637006, 2006)	Boolean	Indicates if the property type is mandatory.
identifyingName	CharacterString	Identifying name. *Note:*

		The name should be set and used in such a way that the directory entry can be uniquely identified within the frame of the FT_StructuredValueDomain where the property type is included, and that the name can be directly used in technical systems.
name (SIS/TK323 - SS 637006, 2006)	CharacterString	The name of the property type
definition (SIS/TK323 - SS 637006, 2006)	CharacterString	The definition of the property type
multiplicity (SIS/TK323 - SS 637006, 2006)	Multiplicity	Specifies the minimum and maximum number of occurrences of the property
<<voidable>> constraints [0..1] (SIS/TK323 - SS 637006, 2006)	OrderAndUniquenessConstraint	Restricts multiple occurrences (regarding sequencing or duplicates) within the set of occurrences of the property of a instance
valid	DurationDays	Specifies the time period during which the property type is valid and can be used

5.3.8.28.2 Relations

Name	Type	Description
domain (SIS/TK323 - SS 637006, 2006)	ValueDomain	Value Domain for a property type for a structured value domain

5.3.9 Package – Feature Instance

The package describes a model for the feature instances, i.e. occurrences of the feature types described in a feature catalog.

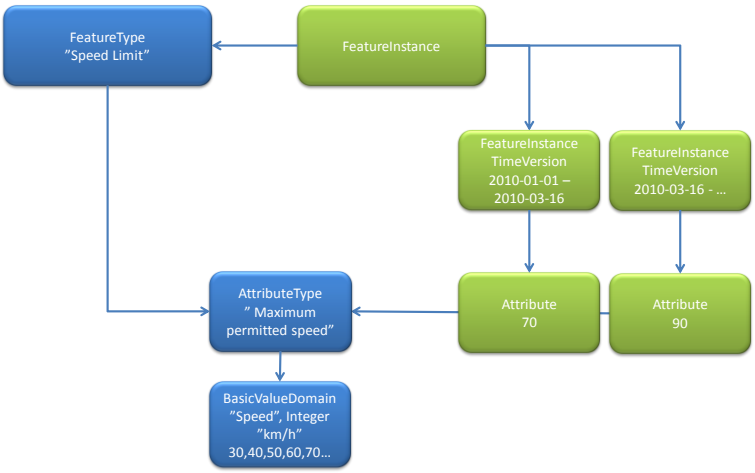


Figure 52 - Relations between events/features and types of features

FeatureInstance corresponds to a feature occurrence. It contains data that is possible to interpret due to the references to the corresponding definitions in a feature catalog.

Time dimensions have to be managed for feature occurrences. In the Swedish standard (SIS/TK323 - SS 637006, 2006) this is managed by each feature occurrence having a set of time versions, where each time version describes the state of the feature during the specified time period. In this model, there are three variations with regard to history management:

- *FeatureInstanceWithoutHistory* represents features that exist independently of the time dimension
- *FeatureInstanceWithHistory* represent features that have a time dimension. However, how a specific feature occurrence changes over time, is not defined. A change in the actual conditions in reality means that one feature occurrence ends and another arises.
- *FeatureInstanceWithTimeVersions* represent features having a time dimension and may additionally also represent how a specific feature instance changes over time. A change in the actual conditions in reality is represented by time versions for the relevant feature instance

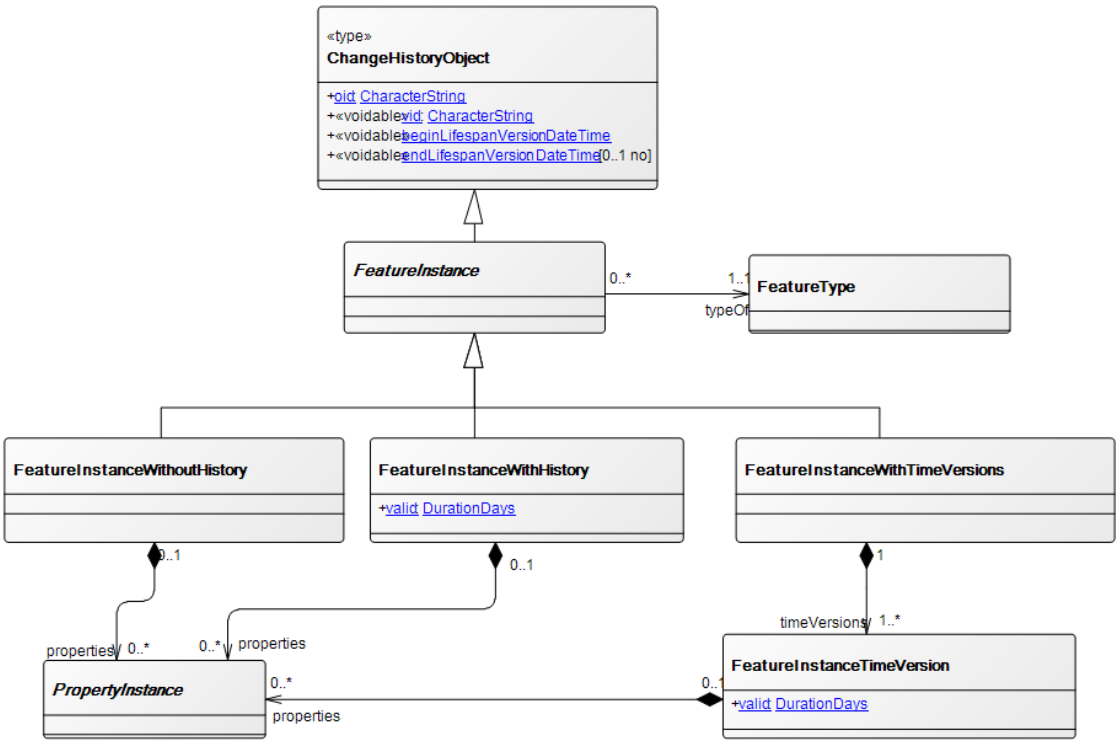


Figure 53 - TransportProperty overview

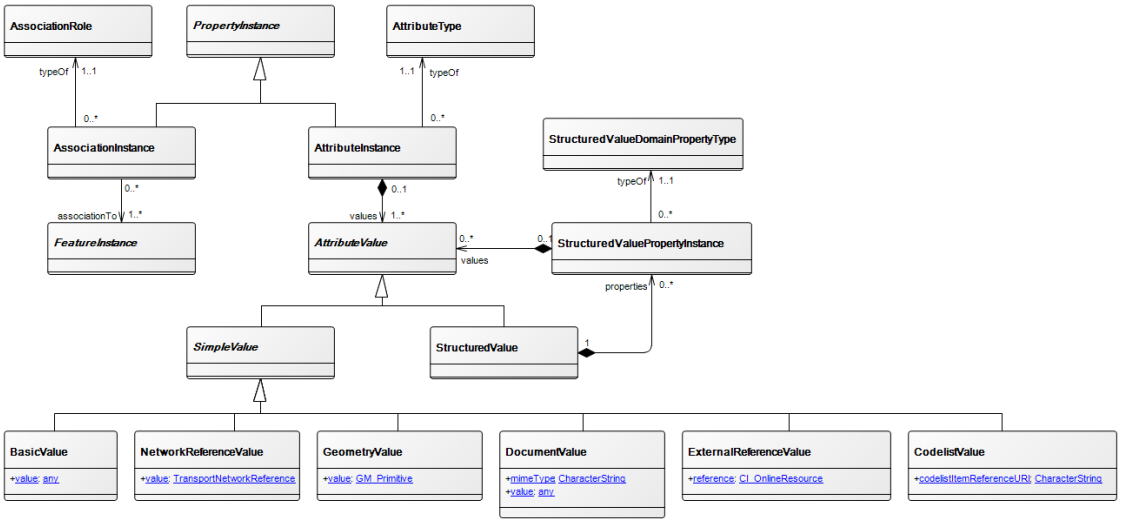


Figure 54 - Attribute overview

5.3.9.1 Class FeatureInstance

A single instance of a particular feature type which represents a feature in the real world. It is an abstract base class for the feature, with or without a history.

Note:

It corresponds to FI_FeatureInstance in (SIS/TK323 - SS 637006, 2006).

5.3.9.1.1 Relations

Name	Type	Description
typeOf (SIS/TK323 - SS 637006, 2006)	FeatureType	The feature type that classifies the feature

5.3.9.2 Class FeatureInstanceWithoutHistory

Single instance of a particular feature type represented without limitation of validity.

Note:

It corresponds to FI_FeatureWithoutHistory in (SIS/TK323 - SS 637006, 2006).

5.3.9.2.1 Relations

Name	Type	Description
properties [0..*] (SIS/TK323 - SS 637006, 2006)	PropertyInstance	The attributes and associations that represent the properties of a feature

5.3.9.3 Class FeatureInstanceWithHistory

Single instance of a particular feature type represented with limitation of validity.

Note:

Has no counterpart in (SIS/TK323 - SS 637006, 2006).

5.3.9.3.1 Attribute

Name	Type	Description
valid	DurationDays	The period during which the (real) feature is valid

5.3.9.3.2 Relations

Name	Type	Description
properties [0..*] (SIS/TK323 - SS 637006, 2006)	PropertyInstance	The attributes and associations that represent the characteristics of a feature

5.3.9.4 Class FeatureInstanceWithTimeVersions

A single instance of a particular feature type which is represented by one or more time versions. This means that one can follow state changes for specific feature instances

Note:

It corresponds to FI_FeatureWithHistory i (SIS/TK323 - SS 637006, 2006).

5.3.9.4.1 Relations

Name	Type	Description
timeVersions [0..*] (SIS/TK323 - SS 637006, 2006)	FeatureInstanceTimeVersion	Time versions of the feature, each of which represents the characteristics of the feature during a specific period of time.

5.3.9.5 Class FeatureInstanceTimeVersion

Representation of the characteristics of a feature during a specific period of time.

Note:

It corresponds to FI_FeatureTimeVersion in (SIS/TK323 - SS 637006, 2006).

5.3.9.5.1 Relations

Name	Type	Description
valid	DurationDays	Specifies the time period during which the time version is valid.

5.3.9.5.2 Relations

Name	Type	Description
properties [0..*] (SIS/TK323 - SS 637006, 2006)	PropertyInstance	The attributes and associations that represent the characteristics of the feature time version

5.3.9.6 Class PropertyInstance

A class that represents the properties of a feature. It is an abstract class.

Note:

It corresponds to FI_PropertyInstance in (SIS/TK323 - SS 637006, 2006).

5.3.9.7 Class AssociationInstance

A class that represents an association between features. It is a subclass to *PropertyInstance*.

Note:

It corresponds to FI_AssociationInstance in (SIS/TK323 - SS 637006, 2006).

5.3.9.7.1 Relations

Name	Type	Description
typeOf (SIS/TK323 - SS 637006, 2006)	AssociationType	Specifies type of association
associationTo [1..*] (SIS/TK323 - SS 637006, 2006)	FeatureInstance	Specifies (instantiates) connections to other features

5.3.9.8 Class AttributeInstance

A property instance which represents a characteristic of a feature in the form of an attribute value. It is a Subclass to *PropertyInstance*.

Note:

It corresponds to FI_AttributeInstance in (SIS/TK323 - SS 637006, 2006).

5.3.9.8.1 Relations

Name	Type	Description
typeOf (SIS/TK323 - SS 637006, 2006)	AttributeType	Specifies the type of attribute
values [1..*] (SIS/TK323 - SS 637006, 2006)	AttributeValue	Specifies (instantiates) attribute values that represent the properties of the feature

5.3.9.9 Class AttributeValue

Abstract class that represents a property of a feature, and is the base class for the concrete attributes values that are retrieved from the corresponding value domains of the catalog model.

Note:

It corresponds to FI_AttributeValue in (SIS/TK323 - SS 637006, 2006).

5.3.9.10 Class SimpleValue

An abstract base class that represents an arbitrary kind of a simple (not complex) attribute value. It is a subclass to *AttributeValue*.

Note:

The corresponding class is missing in (SIS/TK323 - SS 637006, 2006)*, but is available here, mostly because of "symmetry reasons" to make the model more clear, and for it to be linked with the catalog model in a more apparent way.*

5.3.9.11 Class StructuredValue

Represents an occurrence of a structured attribute value comprising a number of properties (the types in turn, can be simple or composite). It is a subclass to *AttributeValue*.

Note:

It corresponds to FI_StructuredAttributeValue in (SIS/TK323 - SS 637006, 2006).

5.3.9.11.1 Relations

Name	Type	Description
properties [0..*]	StructuredValuePropertyInstance	Represents (instantiates) the properties included in the structured attribute value.

5.3.9.12 Class StructuredValuePropertyInstance

Represents an occurrence of a property, which is included in a part of a structured attribute value.

Note:

It corresponds to FI_StructuredAttributeMemberInstance in (SIS/TK323 - SS 637006, 2006).

5.3.9.12.1 Relations

Name	Type	Description
typeOf (SIS/TK323 - SS 637006, 2006)	StructuredValueDomainPropertyType	Specifies the type for the property.

values [0..*]	AttributeValue	Attribute values belonging to the property.

5.3.9.13 Class BasicValue

One type of a simple value attribute, whose value domain is a *BasicValueDomain*.

Note:

It corresponds to FI_ThematicAttributeValue in (SIS/TK323 - SS 637006, 2006).

5.3.9.13.1 Attribute

Name	Type	Description
value	Any	The attribute value. *Note:* *The type of attribute value must correspond to the specified type restriction for the corresponding BasicValueDomain*

5.3.9.14 Class NetworkReferenceValue

One type of a simple value attribute, whose value domain is a *NetworkReferenceValueDomain*.

Note:

It corresponds to NW_ExtentAttributeValue in (SIS/TK323 - SS 637006, 2006).

This is the type of the attribute value to be used to hold network references, i.e. position relative to the reference network.

5.3.9.14.1 Attribute

Name	Type	Description
value	TransportNetworkReference	The attribute value, i.e. network reference. *Note:* *The type of network reference must correspond to the specified type restriction for the*

		corresponding NetworkReferenceValueDomain

5.3.9.15 Class GeometryValue

One type of a simple value attribute, whose value domain is geometry.

Note:

It corresponds to NW_SpatialAttributeValue in (SIS/TK323 - SS 637006, 2006).

5.3.9.15.1 Attribute

Name	Type	Description
value	GM_Primitive	The attribute value, i.e. geometry. *Note:* *The type of geometry must correspond to the specified type restriction for the corresponding GeometryValueDomain*

5.3.9.16 Class DocumentValue

One type of a simple value attribute, whose value domain is a document of any kind.

5.3.9.16.1 Attribute

Name	Type	Description
mimeType	CharacterString	Specifies the type of document stored in *value.* Should (unless otherwise stated) follow IANA MIME Media Types (http://www.iana.org/assignments/media-types/index.html) *Example:* *.docx – "application/vnd.openxmlformats-officedocument.wordprocessingml.document"* *.jpg – "image/jpeg*
value	Any	The attribute value, i.e. a representation of the document. *Note:*

		The document type must correspond to the specified type restriction for the corresponding DocumentValueDomain. *Questions about specific storage format (e.g. base64 encoding of binary documents, etc.) is referred to the implementation of the model.*

5.3.9.17 Class ExternalReferenceValue

A type of simple attribute value, whose value domain is an external reference, typically an available resource on the web.

5.3.9.17.1 Attribute

Name	Type	Description
reference	CI_OnlineResource (ISO/TC211 - ISO/TS 19115)	Reference to the external resource

5.3.9.18 Class CodelistValue

A type of simple attribute value, whose value domain is an external code list.

5.3.9.18.1 Attribute

Name	Type	Description
codelistItemReferenceURI	CharacterString	Reference to a specific code in the external code list. *Note:* *The exact form of such a reference needs to be further investigated. Presumably, principles of managing code lists (dvs. <<codeList>>) within INSPIRE can be used.*

5.3.10 Package – Change management (SIS/TK323 - SS 637007, 2006)

The package is based on (SIS/TK323 - SS 637007, 2006) and describes a model for representing changes (additions, updates, deletions) in datasets. The model is equally applicable to

representations in databases (logging) and data exchange. Note that the changes referred to, are changes of data (objects) and not changes in the reality that the objects represent. Data may change for several reasons:

- Reality has changed and the data needs to be adapted to this
- Data needs to be improved to better correspond to reality

This should therefore NOT be confused this with the history which, with the help of a time dimension in the data, describes how the reality changes over time.

The model assumes that changes are managed at object level, i.e. it is an object that is created, replaced or removed. An object is an instance of a class that has its own identity. The attributes of the object can change over time. This is represented in this model by the object being replaced in its entirety by a new version. However, this model does not put any aspects into what constitutes an object. The objects to be handled for changes should belong to a class that inherits from the class *ChangeHistoryObject*.

The requirement for an object to be handled for changes is that it has a unique identity that follows the object from the cradle to the grave. Objects that can be handled for changes also have an optional version identity. The purpose of this identity is to uniquely identify a specific version of the object. This can be used for several different cases:

- Detection of conflicts when objects can be updated simultaneously by different parties
- Detection of synchronization problems when there are parallel copies of data in different databases

Consider the example below:

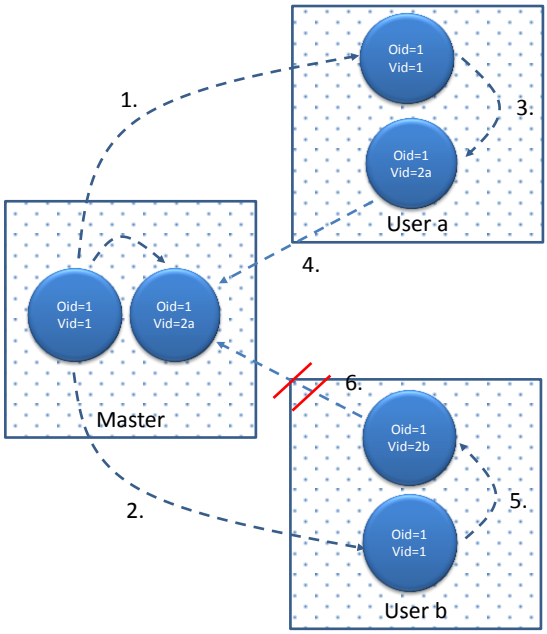


Figure 55 - Illustration of the update conflict

An object with oid=1 and vid=1 exists in a master-database .

1. User a retrieves the object for an update
2. User b retrieves the same object for an update
3. User a updates the object
    a. A version with vid=2a occurs
4. User a updates the object in the master database and indicates that the update is based on version vid=1
    a. A version with vid=2a occurs in the master database because the correct version with vid=1 exists in the master database
5. User b updates the object
    a. A version with vid=2b occurs
6. User b updates the object in the master database and indicates that the update is based on version vid=1
    a. The update will not be performed because the master database now contains the version with vid=1a of the object, which means that there is a conflict.

In summary, the model provides a solution to the following problem:

- Traceability in the data, i.e. an opportunity to, at object level, follow what happened, when it happened and who carried out the changes
- Possibility to exchange updates of data at object level. This provides opportunities to exchange small amounts of data when synchronizing different databases
- Possibility of conflict detection at object level

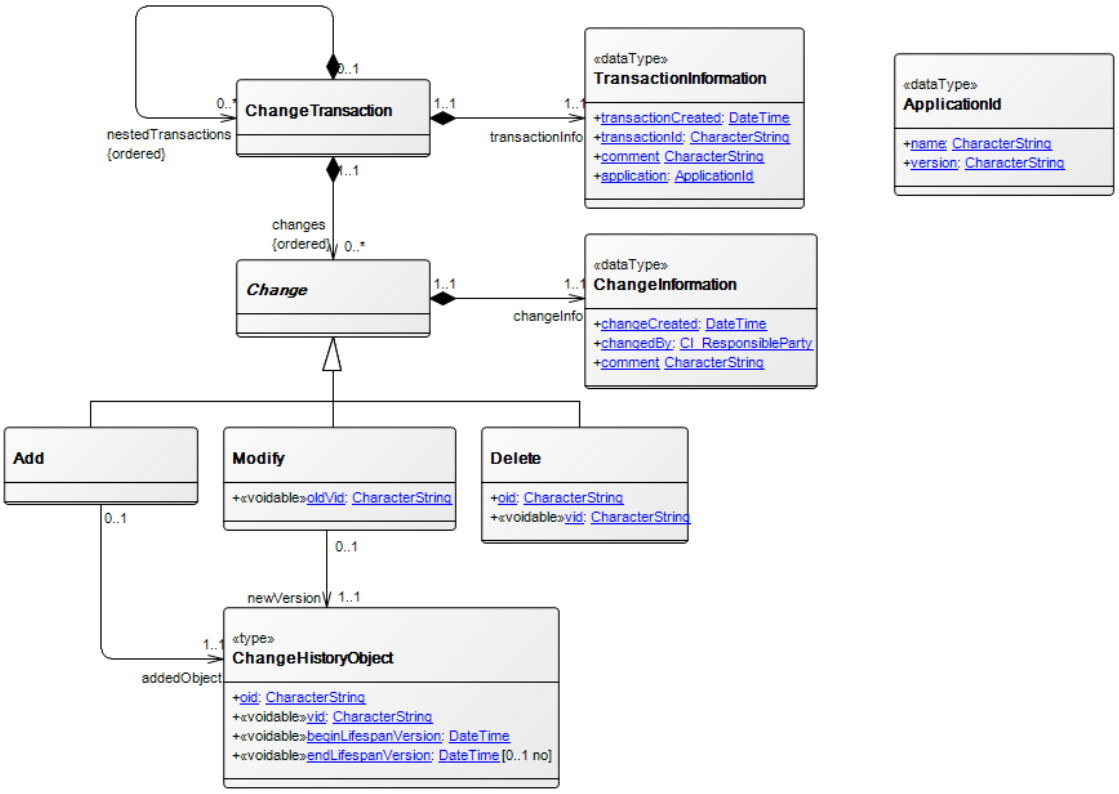


Figure 56 – Change management overview

5.3.10.1 Class ChangeTransaction (SIS/TK323 - SS 637007, 2006)

A change transaction is a set of changes and/or subordinate transactions performed, or are expected to be performed, in its entirety.

Note:

Corresponds to the class CR_ChangeTransaction in (SIS/TK323 - SS 637007, 2006)

5.3.10.1.1 Relations

Name	Type	Description
transactionInfo	TransactionInformation	Describes the transaction
nestedTransactions[0..*]	ChangeTransaction	Refers to the nested sub transactions, if any. *Note:* *This way, transactions can be described in a hierarchy.* *Example:*

		All change transactions that have occurred, since a specific date, are packaged in one coherent transaction for usage (e.g. the provision).

5.3.10.2 Class Change (SIS/TK323 - SS 637007, 2006)

An abstract base class that represents a change in an object of some type.

Note:

Corresponds to the class CR_Change in (SIS/TK323 - SS 637007, 2006)

5.3.10.2.1 Relations

Name	Type	Description
changeInfo	ChangeInformation	Describes the change

5.3.10.3 Class Add (SIS/TK323 - SS 637007, 2006)

Represent addition of an object.

Note:

Corresponds to the class CR_Add in (SIS/TK323 - SS 637007, 2006)

5.3.10.3.1 Relations

Name	Type	Description
addedObject	ChangeHistoryObject	Refers to the object that has been added. *Note:* *Referenced object should be found in the data set as a whole.*

5.3.10.4 Class Modify (SIS/TK323 - SS 637007, 2006)

Represent an update of an object.

Note:

Corresponds to the class CR_Modify in (SIS/TK323 - SS 637007, 2006)

5.3.10.4.1 Attribute

Name	Type	Description

<<voidable>> oldVid	CharacterString	Identifies the version used as starting point for the update.

5.3.10.4.2 Relations

Name	Type	Description
newVersion	ChangeHistoryObject	Refers to the new version of the object. *Note:* *Referenced object should be found in the data set as a whole.*

5.3.10.5 Class Delete (SIS/TK323 - SS 637007, 2006)

Represent removal of an object.

Note:

Corresponds to the class CR_Delete in (SIS/TK323 - SS 637007, 2006)

In a database that has a (real) time dimension for their objects, objects shall normally only be removed if the real object never existed in reality.

5.3.10.5.1 Attribute

Name	Type	Description
Oid	CharacterString	Identifies the object that has been removed.
<<voidable>> vid	CharacterString	Identifies the version of the deleted object.

5.3.10.6 Class <<type>> ChangeHistoryObject

The class is a template for the objects meant to be able for change management. Classes to be change managed shall inherit or realize this class.

Note:

Corresponds to the class CR_ChangeObject in (SIS/TK323 - SS 637007, 2006)

5.3.10.6.1 Attribute

Name	Type	Description

Oid	CharacterString	The unique identity of the object
<<voidable>> vid	CharacterString	The unique identity of the object version.
<<voidable>> beginLifespanVersion (INSPIRE DS TN)	DateTime (ISO/TC211 - ISO/TS 19103)	Date and time for when the object (i.e. data) was created or modified.
<<voidable>> endLifespanVersion (INSPIRE DS TN)	DateTime (ISO/TC211 - ISO/TS 19103)	Date and time for when the object (i.e. data) were replaced or removed.

5.3.10.7 Class <<DataType>> TransactionInformation

A class with metadata concerning a transaction. In the Swedish standard (SIS/TK323 - SS 637007, 2006) corresponding exist in the form of a list of pairs of tags and values (see the class *CR_TagValue* in the standard). In this model, metadata is expressed explicitly.

5.3.10.7.1 Attribute

Name	Type	Description
transactionId	CharacterString	The unique identity of the transaction.
transactionCreated	DateTime	Timing of the when the transaction was created.
Comment	CharacterString	Comment for the transaction.
Application	ApplicationId	Name and version of the application that created the transaction.

5.3.10.8 Class <<DataType>> ChangeInformation

A class with metadata concerning a change. In the Swedish standard (SIS/TK323 - SS 637007, 2006) corresponding exist in the form of a list of pairs of tags and values (see the class *CR_TagValue* in the standard). In this case, metadata is expressed explicitly.

5.3.10.8.1 Attribute

Name	Type	Description
changeCreated	DateTime	Timing of the change

changedBy	CI_ResponsibleParty (ISO/TC211 - ISO/TS 19115)	Identifies who performed the change.
Comment	CharacterString	Comment for the change.

5.3.10.9 Class <<DataType>> ApplicationId

A class that identifies an application with the name and version.

5.3.10.9.1 Attribute

Name	Type	Description
Name	CharacterString	Name of application
Version	CharacterString	Version of application

5.3.11 Package – Metadata

The Swedish national metadata profile (SIS/TK 489 - Nationell metadataprofil, 2012), see http://www.geodata.se/upload/dokument/geodataportalen/Metadata/nationell%20metadataprofil.pdf is referred to in order to describe datasets, such as a database content as a whole, an identifiable subset of a database, an object, a set of published data files or a single published data file.

For metadata on the object level the generic feature model is used (see 5.3.7 Management of network connected features). This model can be used to describe metadata both for other features and the different respective elements in the network. Note that since this information model does not specify a specific structure for features, it does not included any specification of the structure for such metadata-descriptive features.

5.4 Mapping Table between this information model, Swedish standards and INSPIRE DS TN

The table below provides a summary of the mapping between the classes in the Swedish standard, INSPIRE DS TN and this information model. All classes from the respective specifications are listed in the table.

Class in SS 637004/06/07	Class in INSPIRE	Class in new information model	Comment
Network			
NW_Object	TransportObject	--	No equivalent in the new info model. Abstract base class of SS to link to SS 637007. TransportObject in INSPIRE is the abstract base class for all network elements. Contains the attribute geographicalName. The fact that this class is missing in the new information model is not considered to have any consequences, since NW_Object lacks properties.
NW_LocationInstance	--	LocationInstance	INSPIRE has no designated class for this.
NW_NetElement	NetworkElement	TransportNetworkElement	NetworkElement in INSPIRE also includes other types of networks (not transport network)
NW_Node	TransportNode	TransportNode	

Class in SS 637004/06/07	Class in INSPIRE	Class in new information model	Comment
NW_RefNode	TransportNode RoadNode RailwayNode	TransportNode RoadNode RailwayNode	New info model: Only RoadNode has ports. Port model in SS has so far only been used in the exchange of road data. RailwayNode has furthermore a schematic geometry and a requirement to define the possible turns (see Turn) INSPIRE: Nodes have their own attributes (formOf…)
--	--	Turn	Used in the new info model to describe the possible (and preferred) turns in a RailwayNode
NW_Edge	--	--	The new info model has no classes without history. This is not expected to have any consequences in connection with exchanges according to Swedish standards. Current status without history can be generated from a model with a history in the same way as in today's NVDB system.
NW_ValidEdge	TransportLink RoadLink RailwayLink	TransportLink RoadLink RailwayLink	The concepts RefLinkPart and ValidEdge have been merged in the new info model. In practice, this does not mean any difference to today's NVDB RefLinkPart because each RefLinkPart has a start and end node. Transport Link is also the one who primarily carries

Class in SS 637004/06/07	Class in INSPIRE	Class in new information model	Comment
			information about the linear coordinates, as in Swedish Standard they are carried by ports. The model for the road has however been supplemented with ports for compatibility reasons (see NW_RefLink below). Railway Link additionally has schematic geometry and connection to kilometer-setting (*Swedish: kilometrering*).
NW_RefLink	TransportLinkSequence RoadLinkSequence RailwayLinkSequence	TransportLinkSequence RoadLinkSequence RailwayLinkSequence	RoadLinkSequence has ports. The difference is that TransportLinkSequence contains TransportLink instead of NW_RefLinkPart just as in INSPIRE. The assessment is that the mapping can be done between them in 1:1.
NW_RefLinkPart	TransportLink RoadLink RailwayLink	TransportLink RoadLink RailwayLink	The link also serves as a reference link part. In practice, the reference link portions are used in the NVDB today. The only difference is that the link in the new info model has an identity. It does not have an NW_RefLinkPart.
--	TransportLinkSet RoadLinkset RailwayLinkSet	TransportLinkSet RoadLinkset RailwayLinkSet	Class from INSPIRE assessed as essential for the new information model, for such as traffic sites.

Class in SS 637004/06/07	Class in INSPIRE	Class in new information model	Comment
--	TransportNetworkArea RoadArea RailwayArea	TransportNetworkArea RoadArea RailwayArea	Class from INSPIRE assessed as essential for the new information model, for such as built-up area.
--	--	GeometricRepresentation CurveRepresentation PointRepresentation	Classes for schematic geometry for rail
--		GeneralizedTransportLink	Class from INSPIRE (base class for TransportLink and TransportLinkSequence) to connect extent to both.
NW_NetBelonging	--	--	The new info model has no possibility today that a network element can be included in multiple networks. This is not expected to have any practical implications since the equivalent is not used in NVDB context.
NW_ValidNetBelonging	--	--	The new info model has no possibility today that a network element can be included in different networks over time. This is not expected to have any practical implications since the equivalent is not used in NVDB context.

Class in SS 637004/06/07	Class in INSPIRE	Class in new information model	Comment
NW_Net	TransportNetwork	TransportNetwork	In the Swedish standard NW_Net have the attributes *name*, *owner*, and *description*. *Owner* and *description* is currently missing in the new information model. The new info model follows INSPIRE and has geographicalName and typeOfTransport (Mode of transport). Is not expected to have any consequences because the data are not used in the NVDB context.
NW_Port	--	Port	
NW_RefLinkPort	--	RoadLinkSequencePort	The connection is not bidirectional in the new information model because it is deemed to provide an easier handling. Compatibility with SS is still achieved.
NW_RefNodePort	--	RoadNodePort	The connection is not bidirectional in the new information model because it is deemed to provide an easier handling. Compatibility with SS is still achieved.
NW_Direction	--	--	No need to direct links to their geometry. When mapping *direction* shall always be *same*. The limitation is not expected to have any practical consequences.
NW_DetailLevelCode	--	DetailLevelCode	The new info model also contains the detail levels micro and macro for railway.

Class in SS 637004/06/07	Class in INSPIRE	Class in new information model	Comment
NW_Length	--	--	Have no class for this. Using ISO Measure. Mapping is possible.
NW_RelativeDistance	--	--	Have no class for this. Using ISO Measure. Mapping is possible.
NW_System	--	TransportSystem	Swedish standard has 0.1 respectively 0..* in cardinality for superElement and subElements. Do not understand why. In the new info model we demand at least 1. Will have no practical consequences in that the functionality is not used in today's NVDB and the new information model is more restrictive in this respect.
NW_Correspondence	--	--	No need for this has been seen in the new information model. Is not expected to have any consequences as it is not used in the NVDB. Has to do with storing counterparts when sharing of links.
NW_ValidCorrespondence	--	--	See above.
NW_TrafficCategory	--	--	Swedish standard has a model where mode of transport is defined as an NW_TrafficCategory, i.e. not a foregone conclusion which available modes of transport there are. Thereafter each mode of transport refers to dispersions to

Class in SS 637004/06/07	Class in INSPIRE	Class in new information model	Comment
			specify which routes are included and not included.
--	TypeOfTransport	TypeOfTransport	The new info model follows INSPIRE where the transport types are predefined (road, rail, air, maritime and cable). Each network belongs to only one transport type. Is not expected to present a problem because NW_TrafficCategory is not used in NVDB.
NW_Period	validFrom/validTo	DurationDays	DurationDays allows only day-dissolution. NW_Period in SS allows arbitrary dissolution. In NVDB day day-dissolution is used. This restriction means no practical limitations.
NW_LinkPosition	--	LinkPosition	
NW_LinkPositionRelDist	--	LinkPositionMeasure	
NW_LinkPositionDist	Corresponds to the length-attribute in INSPIRE	LinkPositionDist	
NW_LinkPositionStart	--	LinkPositionStart	
NW_LinkPositionEnd	--	LinkPositionEnd	

Class in SS 637004/06/07	Class in INSPIRE	Class in new information model	Comment
NW_LinkPositionPoint	--	LinkPositionPoint	
--		LinkPositionReferent	Has been added in the new info model to cope with kilometer-marking. Kilometer- markings cannot be exchanged via the SS. No problem in practice, because the systems are mappable.
--	MarkerPost	Referent KMPost	See above (KMPost represents km-pole). No clear use in INSPIRE. There is no possibility to use as reference.
NW_Extent	NetworkReference	TransportNetworkReference	A NW_Extent in Swedish standard has cardinality 0..1 for locationInstance, i.e. allow extent without network elements. The New information model requires exactly 1 locationInstance. Means no consequence in practice since extent without the network elements are not considered particularly meaningful.
NW_LinkExtent	LinkReference	TransportLinkReference	The new information model has broken out all attributes except for direction to a class of its own TransportLinkReferenceAttr to have a "pure" class with only direction in order to define the turns. Even INSPIRE has direction on this level.

Class in SS 637004/06/07	Class in INSPIRE	Class in new information model	Comment
--	--	TransportLinkReferenceAttr	Subclass to TransportLinkReference that adds the rest of the extent attributes. heightPosition in SS is named verticalPosition in the new info model. laneCode is down-lifted to the road specific class because it has no application to railway.
NW_PointExtent	SimplePointReference	PointLinkReference	INSPIRE only has atPosition:Length and offset.
NW_LineExtent	SimpleLinearReference	LineLinkReference	INSPIRE only has fromPosition:Length, toPosition:Length and offset.
--	--	LineRoadLinkReference	Broken out from LineLinkReference in order not to overload the railway with the possibility to make extent to traffic lane.
--	--	PointRoadLinkReference	Broken out from PointLinkReference in order not to overload the railway with the possibility to make extent to traffic lane.
NW_LateralPosition		LateralPosition	Possibility for left_and_right removed in the new info model.
NW_LocDirection		LocDirection	Possibility for same_and_opposite removed in the new info model.
NW_HeightPosition		VerticalPosition	
NW_LaneCode		--	Translated only to CharacterString in SS. Unnecessary typing.
NW_NodeExtent		TransportNodeReference	

Class in SS 637004/06/07	Class in INSPIRE	Class in new information model	Comment
NW_NodeExtentAttr		TransportNodeReferenceAttr	heightPosition renamed to verticalPosition in the new info model.
NW_RoadExtent		LineRoadLinkRoadReference	
NW_TurnExtent		TurnReference	
NW_ManoeuvreExtent		ManoeuvreReference	
--	--	PointReference	Class of the new information model to describe extent which is either to the node or point on the link.
--	--	TransferNodeReference	In the new information model to be able to link data to TransferNode
--	--	StopPointReference	In the new information model to be able to link data to StopPoint
--	--	AccessPointReference	In the new information model to be able to link data to AccessPoint
--	--	TransportNetworkAreaReference	In the new information model to be able to link data to TransportNetworkArea
--	--	TransportLinkSetReference	In the new information model to be able to link data to TransportLinkSet
--	--	InterNetworkRelationshipReference	Abstract base class to InterNetworkConflictReference and InterNetworkConnectionReference below.
--	--	InterNetworkConflictReference	In the new information model to be able to link data to

Class in SS 637004/06/07	Class in INSPIRE	Class in new information model	Comment
			InterNetworkConflict (level crossing)
--	--	InterNetworkConnectionReference	In the new information model to be able to link data to InterNetworkConnection
NW_SystemExtent	--	--	Have not seen any need to tie data to TransportSystem
NW_LinkRole	--	LinkRole	
NW_ExtentValueDomain	--	NetworkReferenceValueDomain	Do not have different classes in the new information model. Instead, we have an enumeration in NetworkReferenceValueDomain that specifies which type of dispersion that is valid. The new information model adds a possibility to specify the validity for transport types and levels of detail.
NW_LinkExtentValueDomain	--	NetworkReferenceValueDomain	See above
NW_NodeExtentValueDomain	--	NetworkReferenceValueDomain	See above
NW_ExtentAttributeValue	--	NetworkReferenceValue	
NW_NetConnector	--	--	The new info model do not have classes but instead history

Class in SS 637004/06/07	Class in INSPIRE	Class in new information model	Comment
NW_ValidNetConnector	NetworkConnection	InterNetworkConnection	NetworkConnection in INSPIRE connects 2..* network element and may represent connections that are: - CrossBorderConnected - CrossBorderIdentical - Intermodal (i.e. switching point in local traffic /transferNode) **Perhaps InterNetworkConnection should also involve boundary nodes in the new info model??**
--	--	InterNetworkRelationship	Base class of the new information model since it apart from InterNetworkConnection also has InterNetworkConflict.
--	--	InterNetworkConflict	A new need that has been identified in the new information model, as for example for level crossing road/rail.
NW_Terminal	NetworkConnection	--	The new info model has no classes without history. INSPIRE: s model do not support the stop and access points and also lack history.
NW_ValidTerminal	--	TransferNode	In the new info model the starting point is retrieved from ISO 19147. Furthermore, a TransferNode is a LocationInstance to where you can connect extent (i.e. events). The functionality is not used today at

Class in SS 637004/06/07	Class in INSPIRE	Class in new information model	Comment
			the exchange with NVDB. If you want to do so in the future, the information is mappable.
NW_ExchangePoint	--	--	The new info model has no classes without history.
NW_ValidExchangePoint	--	StopPoint	In the new info model the starting point is retrieved from ISO 19147. Furthermore, a StopPoint is a LocationInstance to where you can connect dispersions (i.e. events).
--	--	AccessPoint	In the new info model the starting point is retrieved from ISO 19147. Furthermore, an AccessPoint is a LocationInstance to where you can connect extent (i.e. events).
--	TransportTypeValue	TransportTypeValue	Attempted refinement of this with transport types, transport modes and means of transport (missing in SS) and also in relation to ISO 19147
--	VehicleTypeValue??	TransportMeanCode	Attempted refinement of this with transport types, transport modes and means of transport (missing in SS) and also in relation to ISO 19147. INSPIRE has a list of its own ...

Class in SS 637004/06/07	Class in INSPIRE	Class in new information model	Comment
--	--	TransportModeCode	Attempted refinement of this with transport types, transport modes and means of transport (missing in SS) and also in relation to ISO 19147
--	TransportPoint	--	Abstract base class in INSPIRE. Unsure use in INSPIRE (base class to MarkerPost)
--	NetworkProperty	--	Abstract base class for events in INSPIRE. The new info model and SS has a generic event model.
--	MaintenanceAuthority	--	Event/feature
--	OwnerAuthority	--	Event/feature
--	VerticalPosition	--	Event/feature
--	TrafficFlowDirection	--	Event/feature
--	ConditionOfFacility	--	Event/feature (compare with status)
--	RestrictionForVehicles	--	Event/feature
--	AccessRestriction	--	Event/feature
--	Road	--	A network element in INSPIRE but event/feature in the new information model.
--	ERoad	--	A network element in INSPIRE but event/feature in the new information model.
--	RoadServiceArea	--	A network element in INSPIRE but event/feature in the new information model.

Class in SS 637004/06/07	Class in INSPIRE	Class in new information model	Comment
--	VehicleTrafficArea	--	A network element in INSPIRE but event/feature in the new information model.
--	FunctionalRoadClass	--	Event/feature
--	FormOfWay	--	Event/feature
--	RoadWidth	--	Event/feature
--	SpeedLimit	--	Event/feature
--	NumberOfLanes	--	Event/feature
--	RoadServiceType	--	Event/feature
--	RoadName	--	Event/feature
--	RoadSurfaceCategory	--	Event/feature
--	RailwayLine	--	A network element in INSPIRE but event/feature in the new information model.
--	RailwayYardNode	--	A network element in INSPIRE but event/feature in the new information model.
--	RailwayStationNode	--	A network element in INSPIRE but event/feature in the new information model.
--	RailwayYardArea	--	A network element in INSPIRE but event/feature in the new information model.
--	RailwayStationArea	--	A network element in INSPIRE but event/feature in the new information model.
--	NominalTrackGauge	--	Event/feature
--	RailwayUse	--	Event/feature

Class in SS 637004/06/07	Class in INSPIRE	Class in new information model	Comment
--	DesignSpeed	--	Event/feature
--	NumberOfTracks	--	Event/feature
--	RailwayType	--	Event/feature
--	RailwayElectrification	--	Event/feature
--	Network for cable, sea and air transport	--	Outside the scope of the SS and the new info model
Event/Feature model			
FT_FeatureCatalogue	--	FeatureCatalogue	
FT_CatalogueEntry	--	CatalogueEntry	
FT_FeatureType	--	FeatureType	
FT_ValueDomain	--	ValueDomain	
FT_Category	--	Category	
FT_PropertyType	--	PropertyType	A small difference between the new info model and SS is that the new information model owns a FeatureType for all its PropertyTypes (both AttributeTypes and AssociationTypes). In SS a FeatureType only owns it's AttributeTypes and the ownership of AssociationTypes is unclear.
FT_AttributeType	--	AttributeType	
FT_AssociationType	--	AssociationType	see PropertyType above
FT_InstanceHistorySelect	--	InstanceHistoryType	UsesType as suffix since Select feels a little EXPRESS-inspired.

Class in SS 637004/06/07	Class in INSPIRE	Class in new information model	Comment
FT_AssociationTypeSelect	--	--	In the new info model, we distinguish only if an association is a composition or not. The semantics of Aggregation feels unclear and is therefore not used. Because of this, this has only become a isComposition : boolean in AssociationType
FT_OrderAndUniquenessConstraint	--	OrderAndUniquenessConstraint	
FT_SimpleValueDomain	--	SimpleValueDomain	
FT_StructuredValueDomain	--	StructuredValueDomain	
FT_StructuredValueDomainMember	--	StructuredValueDomainPropertyType	PropertyType feels like a better concept than member.
FT_SpatialValueDomain	--	GeometryValueDomain	Feels like a better name because we in this model restrict to describe geometry attributes and not topology attributes.
FT_LocationalValueDomain	--	--	Not used in practice in NVDB and is therefore not used in the new info model. May be handled with one or more BasicValueDomain.
FT_MetaDataValueDomain	--	--	Not used in practice in NVDB and is therefore not used in the new info model. May be handled with one or more BasicValueDomain.
FT_ThematicValueDomain	--	BasicValueDomain	
FT_TemporalValueDomain	--	--	Not used in practice in NVDB and is therefore not used in the new info

Class in SS 637004/06/07	Class in INSPIRE	Class in new information model	Comment
			model. May be handled with one or more BasicValueDomain.
FT_EventValueDomain	--	--	Not used in practice in NVDB and is therefore not used in the new info model. May be handled with one or more BasicValueDomain.
FT_StateValueDomain	--	--	Not used in practice in NVDB and is therefore not used in the new info model. May be handled with one or more BasicValueDomain.
FT_CycleValueDomain	--	--	Not used in practice in NVDB and is therefore not used in the new info model. May be handled with one or more BasicValueDomain.
--	--	DocumentValueDomain	To store BLOB in a generic event/feature model
--	--	ExternalReferenceValueDomain	To store external references in a generic event/feature model
--	--	CodeListValueDomain	To store references to code in a generic event/feature model
FT_StdValueRange	--	StdValueRange	
FT_ValidValue	--	ValidValue	
FT_ValidString	--	ValidString	
FT_ValidEnumeration	--	ValidEnumeration	
FT_ValidRange	--	ValidRange	
FT_ValidRangeValueType	--	ValidRangeValueType	

Class in SS 637004/06/07	Class in INSPIRE	Class in new information model	Comment
FT_Period	--	DurationDays	DurationDays allows only day dissolution. FT_Period in SS allows arbitrary dissolution. In NVDB day is used.
--	--	DataType	In SS TypeName is used, which is unspecified. In the new information model, we specify the possible basic data types in an enumeration.
FI_FeatureInstance	--	FeatureInstance	
FI_FeatureWithHistory	--	FeatureInstanceWithTimeVersions	In the new information model we differentiate between events which are authentic and events that have time versions (where you can follow an objects change over time).
FI_FeatureWithoutHistory	--	FeatureInstanceWithoutHistory	
--	--	FeatureInstanceWithHistory	Feature with validity not with time versions.
FI_FeatureTimeVersion	--	FeatureInstanceTimeVersion	
FI_PropertyInstance	--	PropertyInstance	
FI_AssociationInstance	--	AssociationInstance	
FI_AttributeInstance	--	AttributeInstance	
FI_AttributeValue	--	AttributeValue	
--	--	SimpleValue	Added to the new information model to balance the model next to StructuredValue (see below). Adds nothing beyond that.
FI_ThematicAttributeValue	--	BasicValue	

Class in SS 637004/06/07	Class in INSPIRE	Class in new information model	Comment
FI_LocationalAttributeValue	--	--	Not used in practice in NVDB and is therefore used not in the new information model. May be handled with one or more Basic Value.
FI_MetaDataAttributeValue	--	--	Not used in practice in NVDB and is therefore used not in the new information model. May be handled with one or more Basic Value.
FI_TemporalAttributeValue	--	--	Not used in practice in NVDB and is therefore used not in the new information model. May be handled with one or more Basic Value.
FI_SpatialAttributeValue	--	GeometryValue	Feels like a better name because we in this model restrict to describe the geometry attributes and not topology attributes.
FI_StructuredAttributeValue	--	StructuredValue	
FI_StructuredAttributeMemberInstance	--	StructuredValuePropertyInstance	Uses property instead of member as a concept
FI_ThematicValue	--	--	Data found directly in Basic Value
FI_LocationalValue	--	--	Data found directly in Basic Value
FI_TemporalValue	--	--	Data found directly in Basic Value
FI_SpatialValue	--	--	Data found directly in GeometryValue
FI_MetaDataValue	--	--	Data found directly in BasicValue

Class in SS 637004/06/07	Class in INSPIRE	Class in new information model	Comment
--	--	DocumentValue	To store BLOB in a generic event/feature model.
--	--	ExternalReferenceValue	To store external references in a generic event/feature model.
--	--	CodeListValue	To store references to code in external code list in a generic event/feature model.
FI_ChangedFeatureWithHistory	--	--	Covered by FeatureInstanceWithTimeVersions
FI_ChangedFeatureWithoutHistory	--	--	Covered by FeatureInstanceWithoutHistory
Change management			
CR_ChangeTransaction	--	ChangeTransaction	The new info model has fixed attributes to describe the metadata of the transaction instead of TagValue
CR_Change	--	Change	The new info model has fixed attributes to describe the metadata of the transaction instead of TagValue
CR_Add	--	Add	
CR_Modify	--	Modify	The new info model contains only one identity for oldVersion not a reference to the old object as in SS
CR_Delete	--	Delete	The new info model contains only one identity for deletedObject not a reference to the old object as in SS

Class in SS 637004/06/07	Class in INSPIRE	Class in new information model	Comment
CR_ChangeObject	--	ChangeHistoryObject	The new info model contains begin and endLifespanVersion from INSPIRE
CR_TagValue	--	--	The new info model rather contains predefined data for the transaction and changes in information
Metadata			
Quality Model SS 637004	Dataset level metadata	National metadata profile	In the new information model it is referenced to national metadata profile. Unclear if this is used in practice in connection with data exchange according to SS.

6 Source references

INSPIRE DS TN. (u.d.). *INSPIRE*. Hämtat från Infrastructure for Spatial Information in the European Community: http://inspire.jrc.ec.europa.eu/documents/Data_Specifications/INSPIRE_DataSpecification_TN_v3.1.pdf den 01 03 2012

INSPIRE GCM. (u.d.). *INSPIRE*. Hämtat från Infrastructure for Spatial Information in the European Community: http://inspire.jrc.ec.europa.eu/documents/Data_Specifications/D2.5_v3.2.pdf

ISO/TC211 - ISO 19107. (u.d.). *ISO/TC 211 Geographic information/Geomatics*. Hämtat från http://www.isotc211.org den 09 03 2012

ISO/TC211 - ISO 19108. (u.d.). *ISO/TC 211 Geographic information/Geomatics*. Hämtat från http://www.isotc211.org/ den 01 03 2012

ISO/TC211 - ISO 19109. (u.d.). *ISO/TC 211 Geographic information/Geomatics*. Hämtat från http://isotc211.org den 16 03 2012

ISO/TC211 - ISO 19147. (u.d.). *ISO/TC 211 Geographic information/Geomatics*. Hämtat från http://www.isotc211.org den 07 03 2012

ISO/TC211 - ISO 19148. (u.d.). *ISO/TC 211 Geographic information/Geomatics*. Hämtat från http://www.isotc211.org den 09 03 2012

ISO/TC211 - ISO/TS 19103. (u.d.). *ISO/TC 211 Geographic information/Geomatics*. Hämtat från http://www.isotc211.org/ den 01 03 2012

ISO/TC211 - ISO/TS 19115. (u.d.). *ISO/TC 211 Geographic information/Geomatics*. Hämtat från http://www.isotc211.org den 15 03 2012

ISO/TC211. (u.d.). *ISO/TC 211 Geographic information/Geomatics*. Hämtat från http://www.isotc211.org/ den 01 03 2012

Johansson Jacques, A. (2011). *Projektspecifikation Nya informationsmodeller*. Trafikverket.

OMG. (u.d.). *OCL 2.2*. Hämtat från omg.org: http://www.omg.org/spec/OCL/2.2/ den 07 03 2012

OMG. (u.d.). *Unified Modelling Language*. Hämtat från http://www.uml.org/ den 01 03 2012

Sillevis Smitt, S. L. (2011). *BandIT Nät och Nod*. Borlänge: Triona AB.

SIS/TK 489 - Nationell metadataprofil. (2012). *Geodata - Nationell metadataprofil - Specifikation och vägledning, version 3.1.1*. SIS/TK 489.

SIS/TK320 - SS 637004. (2009). *SS 637004 - Geografisk information - Väg- och järnvägsnät - Applikationsschema*. Stockholm: SIS.

SIS/TK323 - SS 637006. (2006). *SS 637006 - Geografisk information - Typoberoende representation av geografiska företeelser*. SIS.

SIS/TK323 - SS 637006. (2006). *SS 637006 - Geografisk information - Typoberoende representation av geografiska företeelser*. SIS.

SIS/TK323 - SS 637007. (2006). *SS 637007 - Geografisk information - Representation av förändringar i datamängder*. SIS.