INTERNATIONAL RAILWAY SOLUTION

IRS 30100

1st edition, 2016-9

RailTopoModel - Railway infrastructure topological model

Reference Number
IRS 30100:2016

International Railway Solution to be classified in volumes of UIC

III

Application:

With effect from 1st September 2016
All members of the International Union of Railway

Record of updates:

September 2016	First issue

Warning

Printed by the International Union of Railways (UIC)
16, rue Jean Rey 75015 Paris - France, September 2016
Dépôt Légal September 2016

ISBN 978-2-7461-2513-1

The International Railway Solution

The International Railway Standards (IRS) are structured in a General Part and in some eventual Application Parts.

The General Part is valid worldwide, while the Application Parts are valid for a specific railway application, based on a geographical or on a service implementation.

The eventual Application Parts may thus be added according to the current needs of the Railway Community.

Structure of the International Railway Standard:

IRS 30100 RailTopoModel - Railway infrastructure topological model

General Part
Application Part(s) : none

CONTENTS

Foreword

The RailTopoModel Project, led by UIC with major contributions from several Railway Infrastructure Managers and Industrials companies, aims to define a universal description of railways business objects, independent of usages (usage-agnostic), structured in layers (topology, referencing, infrastructure, signalling, ..., project life cycle), and open to future evolutions. The RailTopoModel Project aims to cover progressively the complete Railways Business Objects Model.

IRS 30100 RailTopoModel is intended to be used in all business processes dealing with the design, construction, operation and maintenance of a railway network. The IRS 30100 is the foundation for quick, unambiguous and error-free data storage and data exchange inside and between these business processes.

The RailTopoModel abstracts the underlying, necessary concepts in the form of a UML2.0 class diagram.

An important part of these concepts is supported by a generic model description of the railway topology, in such a way that it applies to any aggregation level in which a railway network may be represented.

Consequently all objects are in abstract terms (classes) directly or indirectly related or connected to the topology of their appropriate aggregation level. Besides physical railway constituents, objects also refer to several kinds of characteristics of a railway. Also, the positioning of objects and instances of classes is covered by different kinds of positioning methods.

The overview below (not following UML conventions) introduces the main objects and dimensions of RailTopoModel.

Functional coverage

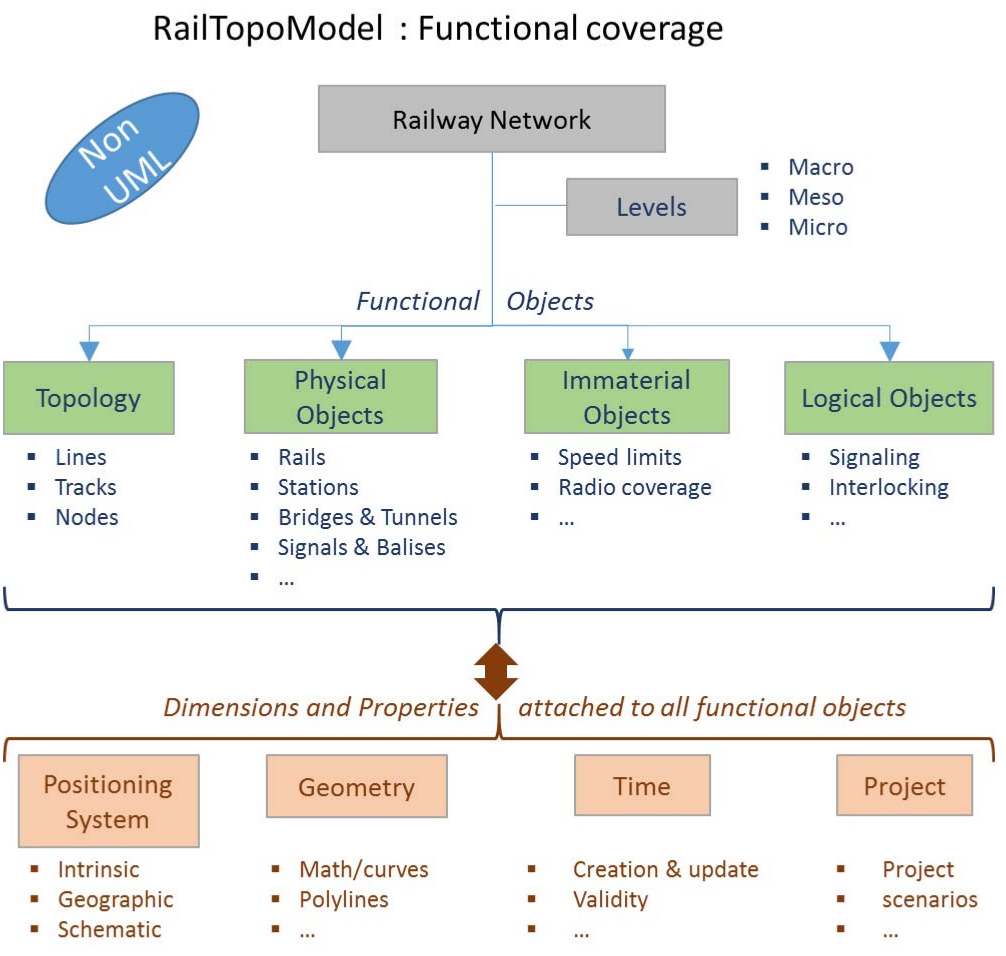


Fig. 1 - Functional coverage of the RailTopoModel

Additional explanations, application examples, documentation and wiki about RailTopoModel (current version and further developments) may be found on the web, following this link: http://www.railtopomodel.org.

Summary

The present standard, IRS30100, complements ISO 191xx series standards by specifying semantics and providing functionalities that are relevant to railway systems.

The present standard is intended to facilitate the implementation of infrastructure management information systems. It includes natively the geographic dimension, and therefore fulfils, inter alia, the requirements of the INSPIRE Directive, when these requirements apply to railway infrastructure. No current ISO 191xx series standard deals specifically with the challenges posed by consistent, scale-independent railway infrastructure data modelling, since these ISO standards stipulate at a higher level.

The present standard deals with semantics close to *EN 28701:2012 "Identification of fixed objects in public transport"* (see Normative references - page 4), namely fixed objects such as infrastructure, or events such as works. *EN 28701:2012*, however, is multimodal, and addresses the transport infrastructure mainly from the point of view of passenger information and timetable management. While the semantics are close, and can fairly easily be linked, the present standard aims at a wider usage (asset management and operational planning and management as well) in a narrower field of application.

Field of application:

- The IRS 30100 RailTopoModel describes a framework of concepts, to support the description of railway infrastructure, starting from the iron network and including business objects: network topology, infrastructure elements, their description, referencing and positioning, their behaviour, etc.

- The RailTopoModel should be used especially when there is a need to describe the network (structure and topology) at various levels of detail, depending on intended usage and on data availability. It is especially put the RailTopoModel at use when infrastructure data is expected to be used by various stakeholders for purposes not precisely known in advance, e.g. for network design and maintenance, traffic scheduling, and traffic management. The description can be as general as corridors; it can be detailed at line level, track level, down to physical components such as switches, lineside signals, or balises. An unlimited set of properties can be attached to component classes, for purposes such as conformity assessment, technical characteristics, life cycle data, including economic aspects, etc.

Normative references

1. railML® - The XML-Interface for railway applications. [Online] [Cited: 01 10 2015.] http://railml.org/.

2. Feasibility Study "UIC RailTopoModel and data exchange format" railML®. [Online] 27. 09 2013. [Retrieved: 01. 10 2015.] http://documents.railml.org/science/270913_trafIT_FinalReportFeasibilityStudyRailTopoModel.pdf.

3. RailTopoModel - Draft. railML®. [Online] 17. 09 2013. [Retrieved: 01. 10 2015.] http://documents.railml.org/science/201213_UIC_RailTopoModel_DraftDec13.pdf.

4. Documents Associated With Unified Modeling Language (UML), V2.4.1. Object Management Group. [Online] 2011. [Retrieved: 01. 10 2015.] http://www.omg.org/spec/UML/2.4.1/.

5. Gély, L., et al. ; A Multi Scalable Model Based On A Connexity Graph Representation; WitPress. [Online] 2010. [Retrieved: 01. 10 2015.] http://www.witpress.com/elibrary/wit-transactions-on-the-built-environment/114/21421; [Print] WIT Transactions on the Built Environment; Computers in Railways XII, 2010, pp.193-204.

6. Data model. Wikipedia. [Online] [Retrieved: 01. 10 2015.] https://en.wikipedia.org/wiki/Data_model.

7. Community of European Railways. CER. [Online] [Retrieved: 01. 10 2015.] http://www.cer.be/.

8. European Rail Infrastructure Managers. EIM. [Online] [Retrieved: 01. 10 2015.] http://www.eimrail.org/.

9. European Train Control System. Wikipedia. [Online] [Retrieved: 01. 10 2015.] https://en.wikipedia.org/wiki/European_Train_Control_System.

10. About INSPIRE. INSPIRE. [Online] [Retrieved: 01. 10 2015.] http://inspire.ec.europa.eu/index.cfm/pageid/48.

11. Object-oriented programming. Wikipedia. [Online] [Retrieved: 01. 10 2015.] https://en.wikipedia.org/wiki/Object-oriented_programming.

12. Rail Freight Corridors (RFCs). RNE. [Online] [Retrieved: 01. 10 2015.] http://www.rne.eu/rail-freight-corridors-rfcs.html.

13. Recommendation on Specification of RINF. European Railway Agency. [Online] [Retrieved: 01. 10 2015.] http://www.era.europa.eu/Document-Register/Pages/Recommendation-on-specification-of-RINF.aspx.

14. Union internationale des chemins de fer. Union internationale des chemins de fer. [Online] [Retrieved: 01. 10 2015.] http://www.uic.org/.

15. Unified Modeling Language. Wikipedia. [Online] [Retrieved: 01. 10 2015.] https://en.wikipedia.org/wiki/Unified_Modeling_Language.

16. Object Management Group. Object Management Group. [Online] [Retrieved: 01. 10 2015.] http://www.omg.org/.

17. High-speed rail in Europe. Wikipedia. [Online] [Retrieved: 01. 10 2015.] https://en.wikipedia.org/wiki/High-speed_rail_in_Europe

The RailTopoModel is based on the following norms resp. standards:

Unified Modeling Language (UML), V2.4.1 (4)

ISO 19148:2012, Geographic information - Linear referencing

List of abbreviations

CER	Community of European Railway and Infrastructure Companies (Community of European Railways, n.d.)
EIM	European Rail Infrastructure Managers (European Rail Infrastructure Managers, n.d.)
ERA	European Railway Agency (now EUAR: European Union Agency for Railways)
ERIM	European Railway Infrastructure Masterplan (at UIC) (ERIM-Project: Publication of the UIC Railway Topology Model, 2014)
ETCS	European Train Control System (European Train Control System)
EU	European Union
GPS	Global Positioning System
IM	Infrastructure Manager
INSPIRE	Infrastructure for Spatial Information in the European Community (About INSPIRE)
LRS	Linear Referencing System
OP	Operational Point (RINF concept)
railML®	Railway Markup Language (1)
RINF	Register of Infrastructure (European Register for railway network Infrastructure) at ERA (Recommendation on Specification of RINF)
SOL	Section Of Line (RINF concept)
UIC	International Union of Railways (Union internationale des chemins de fer, n.d.)
UML	Unified Markup Language (general-purpose modelling language) (Unified Modeling Language)

General Part

1 - RailTopoModel high level concepts

1.1 - Introduction

R.30100.3 The ultimate goal of RailTopoModel is to propose a universal representation of a railway network and associated events, to support and facilitates business development within the rail sector.

R.30100.4 For this purpose, RailTopoModel is based on a graph model, as far as topology is concerned.

R.30100.5 The first objective is to ensure that the model supports current and future railway business needs. To achieve this, the model fulfils the following criteria:

R.30100.6 - The Model provides a topological representation of the iron network which is fully connected and can be visualized schematically. It supports the display of track locations at any detail level, from corridors down to tracks.

R.30100.7 - The Model enables data to be aggregated and disaggregated, while managing connections between detail levels (or “scales”), to make sure that data consistency is retained across all scales.

R.30100.8 - The Model allows permitted routes to be identified, based on network topology and other available information such as events (track possessions), power supply characteristics, signalling assets, etc.

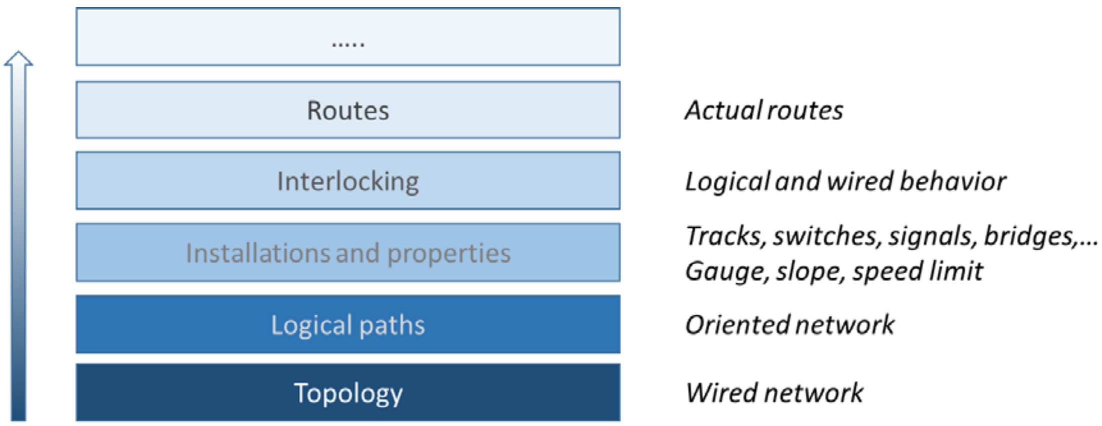
R.30100.9 - The Model supports multiple referencing systems, thus ensuring consistency during transformation from one referencing system to another one. Primary examples are:

R.30100.10 • Linear referencing – using mileposts and “rail addresses”;
R.30100.11 • Positioning - using geographic reference systems;
R.30100.12 • Screen (schematic) coordinates.

R.30100.13 - The model defines and locates “point”, “linear” and “areal” entities, i.e.:

R.30100.14 • Points or nodes, such as any installation and equipment or event, etc.;
R.30100.15 • Lines or edges, such as speed limits, slopes, platforms etc.;
R.30100.16 • Areal objects, such as track circuits, tunnels, stations, etc.

R.30100.17 This model is designed to be enriched progressively with new concepts to support business usages as they evolve



R.30100.19 *Fig. 2 - Layers*

1.2 - Previous work

R.30100.21 The foundation principles of RailTopoModel are based on previous work which is documented, inter alia:

R.30100.22 - in the paper “A multi scalable model based on a connexity graph representation” presented at the 12th International Conference on Computer System Design and Operation in the Railways and other Transit Systems, COMPRAIL 2010, Aug 2010, Beijing, China [5] ;

R.30100.23 - in the feasibility study “UIC RailTopoModel and data exchange format” [2].

1.3 - Related work

R.30100.25 The RailTopoModel is one important basis for the further development of railML®. railML® is an open-source, XML-based data exchange format for IT usages in railways. railML® is developed and maintained by the railML.org initiative. railML.org develops new versions of railML®, starting with railML® 3.0, that are based on RailTopoModel.

R.30100.26 The current state of development of railML® 3.0 is described on the railML.org website [1].

1.4 - Fundamental principles

R.30100.28 Managing and operating a railway network, for both traffic management and maintenance activities, leads to some specific challenges for IT systems and their algorithms for problems like:

R.30100.29 - ensuring consistency in a business process spread over multiple departments working at both line and track levels;

R.30100.30 - supporting a shared view between traffic management and works planning over time, from design to operation;

R.30100.31 - making capacity planning and control-command share a common view on interlocking, from design to operation, including simulations.

R.30100.32 First trials, proof on concepts, and operational developments based on those concepts, prove that the limitations encountered with traditional "monolithic" infrastructure descriptions can be solved using RailTopoModel.

R.30100.33 The aim of any modelling approach is to create an abstract representation of reality. Put simply, it should enable users to understand the following:

R.30100.34 - What: the semantics of assets, i.e. what is installed in the field.

R.30100.35 - Where: the location of assets.

R.30100.36 - How: the connections and dependencies between assets.

R.30100.37 - When: the life cycle of assets.

R.30100.38 - Why: the business rules which dictate how the infrastructure behaves and how it is operated.

R.30100.39 The "what", "where", "how" and "when" are the building blocks of this Model, the description of structuring objects, independently from any usage; i.e. these are the independent foundation layers on which the Model is built.

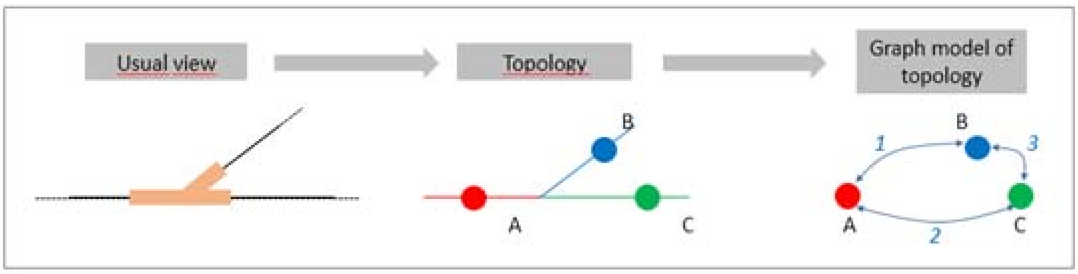
R.30100.40 The "why" drives the Model application, and helps utilise the information in the foundation layers to identify how the network is operated. Those rules will be populated step by step, driven by particular business project plans; these could be, for instance, rules and behaviours for interlocking.

1.5 - Modelling principles

R.30100.42 RailTopoModel is based on Graph Theory, as far as topology is concerned.

R.30100.43 Topology is only logical and therefore independent of any physical or technical items used to represent it. Topology does not for instance assume that the network described is a road network or a railway network.

R.30100.44 Traditionally, a railway network would be represented, at track level, with “nodes” being switches and “edges” being tracks. This is not in line with graph theory, since resources are identified, in the traditional representation, as nodes or edges: In graph theory all resources are nodes, and only nodes; edges represent relations between nodes, and only relations. The network topology must therefore be described as a graph in the following manner:



R.30100.46 *Fig. 3 - From the usual view to the graph model*

R.30100.47 In the graph model of topology, all the nodes and edges of the above usual interpretation are instead derived from a single, so-called “NetElement” class. The NetElements A, B, C are related with each other by the edges 1, 2, 3 that define their connections. A railway graph on any level is in principle a directed graph, even though in most cases it is assumed that rail connections can be used in two directions. The outcome is an accurate railway network functional description. For that reason, in the context of topology, the physical switch device is not considered.

R.30100.48 This principle, which is very different from the usual schematic presentation of a railway network, is consequently applied in the Model.

R.30100.49 Actually, from a graph point of view, NetElements are nodes, and their mutual relations are edges. Now regardless of whether we have to deal with (for example) a section of line or an operational point, we can assign any characteristic to it (as an instance of the class NetEntity). This is what a connexity graph (see point 1.2 - page 9) is all about.

1.6 - Multilevel architecture

1.6.1 - Overview over levels

R.30100.52 One purpose of the Model is to provide a standard network description with various levels of detail, following common railway practice and recent, sector-wide applications such as RINF. Those different “views” of a given network are linked by aggregation rules.

R.30100.53 Depending on business needs or maturity level, data can be entered and used at any scale in the RailTopoModel, without lower levels of detail if these are not required, but ensuring consistency with future evolutions at other levels.

R.30100.54 Implementations of the Model should include at least one of the following levels. Other levels may be defined and used that may be intermediate, or more detailed, or less detailed.

Table 1 : Usual levels

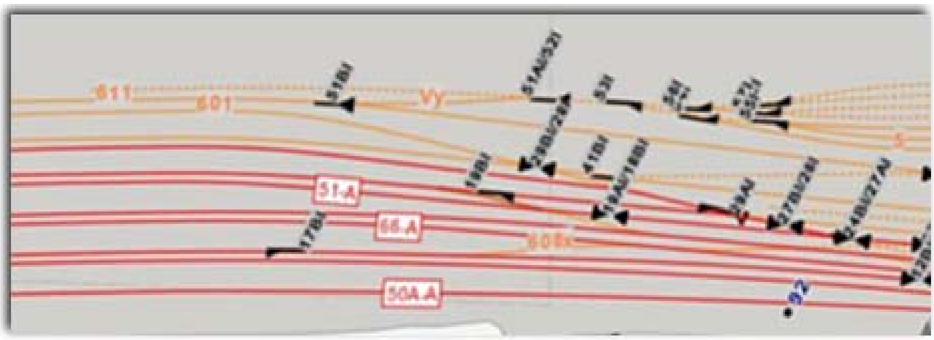
Level	Description	Use cases / examples
Micro	Large scale Detailed information at track level. Basis = Switches or buffer stops that are connected by tracks.	ETCS, Interlocking, maintenance, asset (lifecycle) management
Meso	Intermediate scale Functional information at track level. Basis = Operating points that are connected by one or more tracks.	Visualise and process capacity properties of Sections of Lines. Capacity properties are directly linked with the number of tracks.
Macro	Small scale Minimal track level information. Basis = Operating points connected with each other via single connections (one or more tracks).	Network of railway lines and stations. Timetabling information.

R.30100.69 The RailTopoModel itself does not mandate particular description levels. The analysis of potential use cases brought forward three description levels of high significance for railway IT systems.

R.30100.70 As the model describes a generic network, and as every detailed level shares the same concepts, an unspecified number of other levels may be derived according to the requirements of the respective use case.

1.6.2 - Micro level

R.30100.72 This level defines the network in a way very close to the physical level as commonly viewed, as illustrated below:



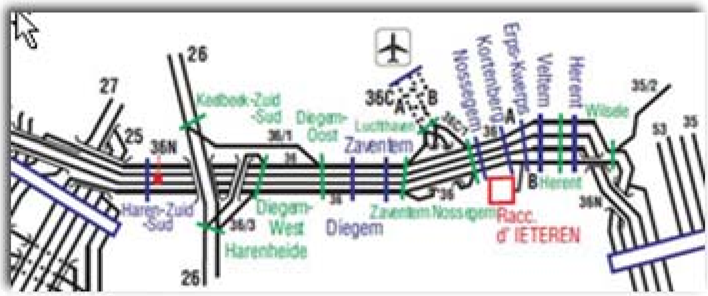
R.30100.74 *Fig. 4 - Micro level sample*

R.30100.75 At micro level, the non-linear elements (defined in section 6.2.5) are the switch points, the network borders, maybe some administrative points (ownership boundaries), buffer stops.

The linear elements (defined in point 2.2.4 - page 29) are the tracks connected to, rather than connecting, those non-linear Elements.

1.6.3 - Meso level

R.30100.77 The Meso level brings the description of the tracks between the operational points of the network into focus.

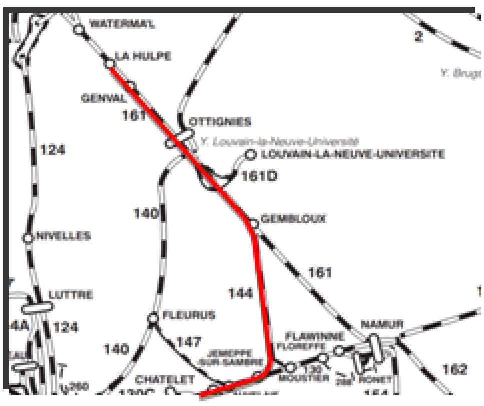


R.30100.79 *Fig. 5 - Meso level sample*

R.30100.80 The non-linear elements are the Operating Points (OP = stations, yards, junctions, boundaries), and linear elements are the tracks connecting Ops.

1.6.4 - Macro level

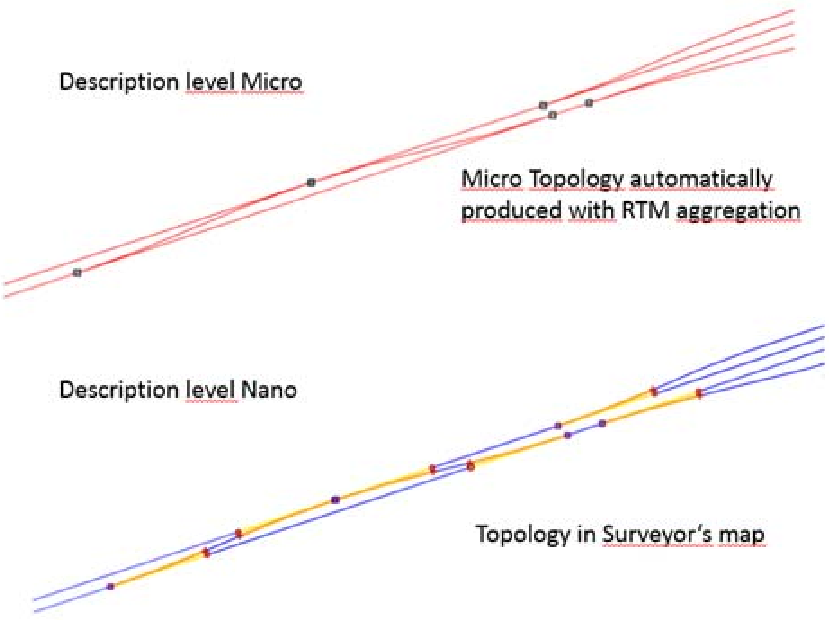
R.30100.82 The Macro level aims to describe the network at regional or national level, with the non-linear elements being the boundaries and the major OP’s, while the linear elements are the sections of lines connecting those points.



R.30100.84 *Fig. 6 - Macro level sample*

1.6.5 - Nano level and other levels

R.30100.86 The nano level could be described as a properly attributed surveyor's map, including topological properties of the rail network in the finest possible granularity.



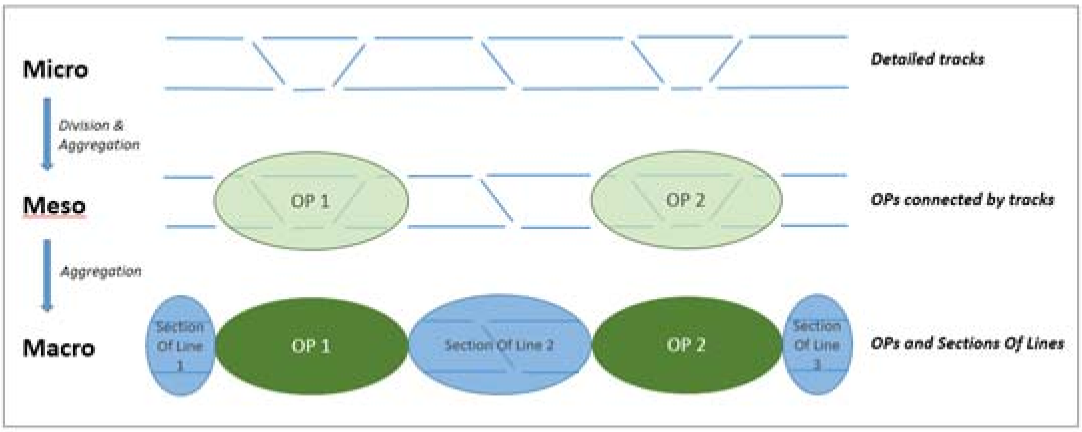
R.30100.88 *Fig. 7 - Nano description level*

R.30100.89 Typically, this level will be built starting from the micro level, by using "switch templates". Conversely, topological properties of the Micro level can be automatically produced from the Nano topology using RTM aggregation. In case the detailed nano level information does not exist, it is possible to add navigability information manually to the track edges at micro level.

R.30100.90 Use cases for the Nano level would include interlocking and asset management, for instance.

1.6.6 - Aggregation principle

R.30100.92 The following chart shows the principle for aggregation from tracks (micro) to OPs (meso), then Sections of Lines (macro):



R.30100.94 *Fig. 8 - Aggregation example*

1.7 - Packages and main elements

1.7.1 - General

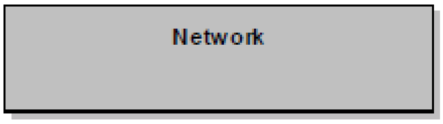
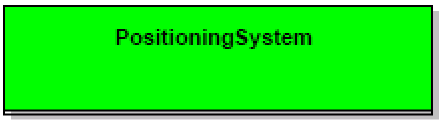
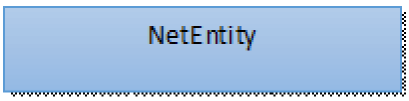
R.30100.97 The RailTopoModel is described in UML notation.

R.30100.98 Similar to UML, the modelling concepts of the RailTopoModel are grouped within packages. A package consists of a collection of tightly coupled modelling concepts. Each package deals with a specific aspect of the model.

1.7.2 - Packages

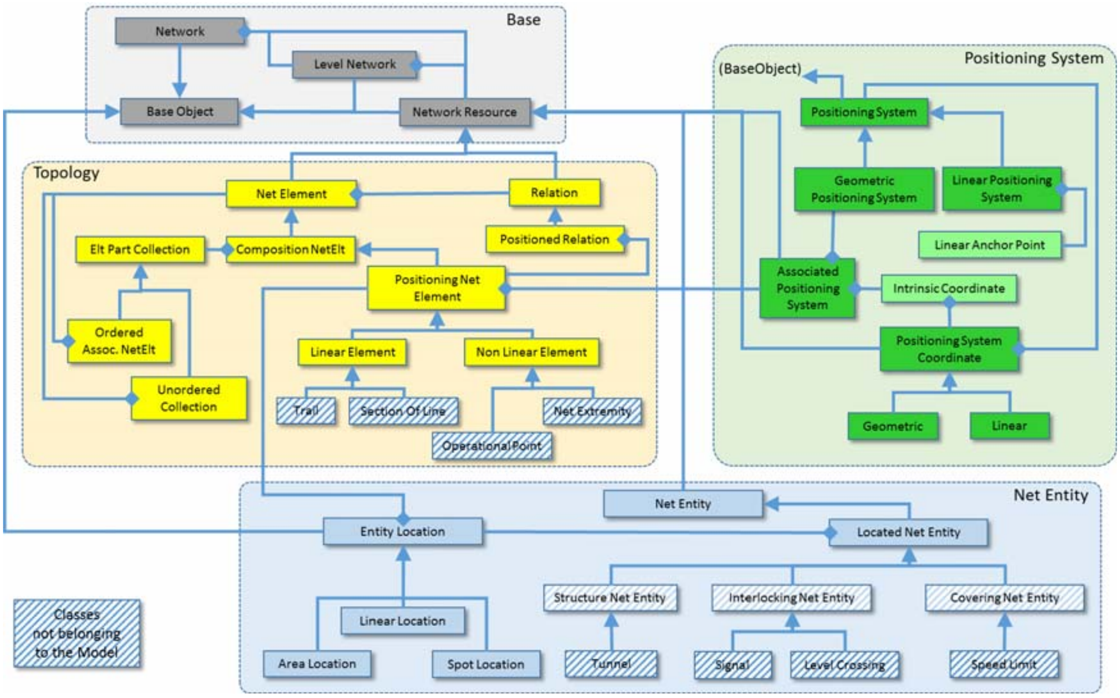
R.30100.100 The RailTopoModel representation in UML consists of four packages: Base, Topology, Positioning Systems, and Net Entities. These packages are depicted using specific colour codes.

Table 2 : RTM Packages

Package	Colour code	Main element(s)
Base	Grey 	Network, LevelNetwork
Topology	Yellow 	NetElement, Relation, CompositionNetElement
Positioning Systems	Green 	PositioningSystem, IntrinsicCoordinate
Net Entities	Light blue 	LocatedNetEntity, EntityLocation

1.7.3 - Model overview

R.30100.104 Note: not all classes are represented in the following class diagram. The diagram follows simplified UML conventions: full arrows represent generalizations, links with diamonds represent associations (possibly aggregations or compositions). A few classes (not part of the Model) have been added for illustrative purposes. This figure as a whole is not normative.



R.30100.106 *Fig. 9 - RTM Class diagram*

R.30100.107 The packages and classes are described into more detail under chapter 6, following conventions detailed under point 1.8.

1.8 - Conventions for package and class description

R.30100.109 All concepts of RailTopoModel are depicted as UML classes. Each RailTopoModel concept is described in a sub-section of point 2 - page 20.

R.30100.110 The names of the RailTopoModel concepts are enclosed in double quotation marks like in "NetworkResource".

R.30100.111 Description items consist of:

R.30100.112 1. Definition

R.30100.113 2. Diagram with current class and its attributes including inherited attributes

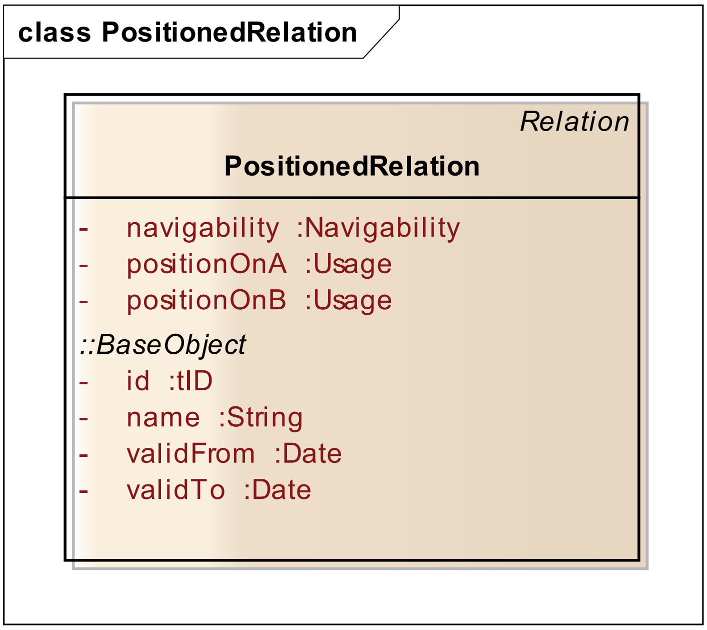
R.30100.114 3. Description of associations

R.30100.115 4. Table with own attributes (inherited attributes may not be shown)

R.30100.116 5. Class diagram containing all classes up to the root class and all associated classes

R.30100.117 6. List of classes that are derived from the current class.

R.30100.118 A sample diagram representing a class and its attributes is shown below:



R.30100.120 *Fig. 10 - Sample class representation*

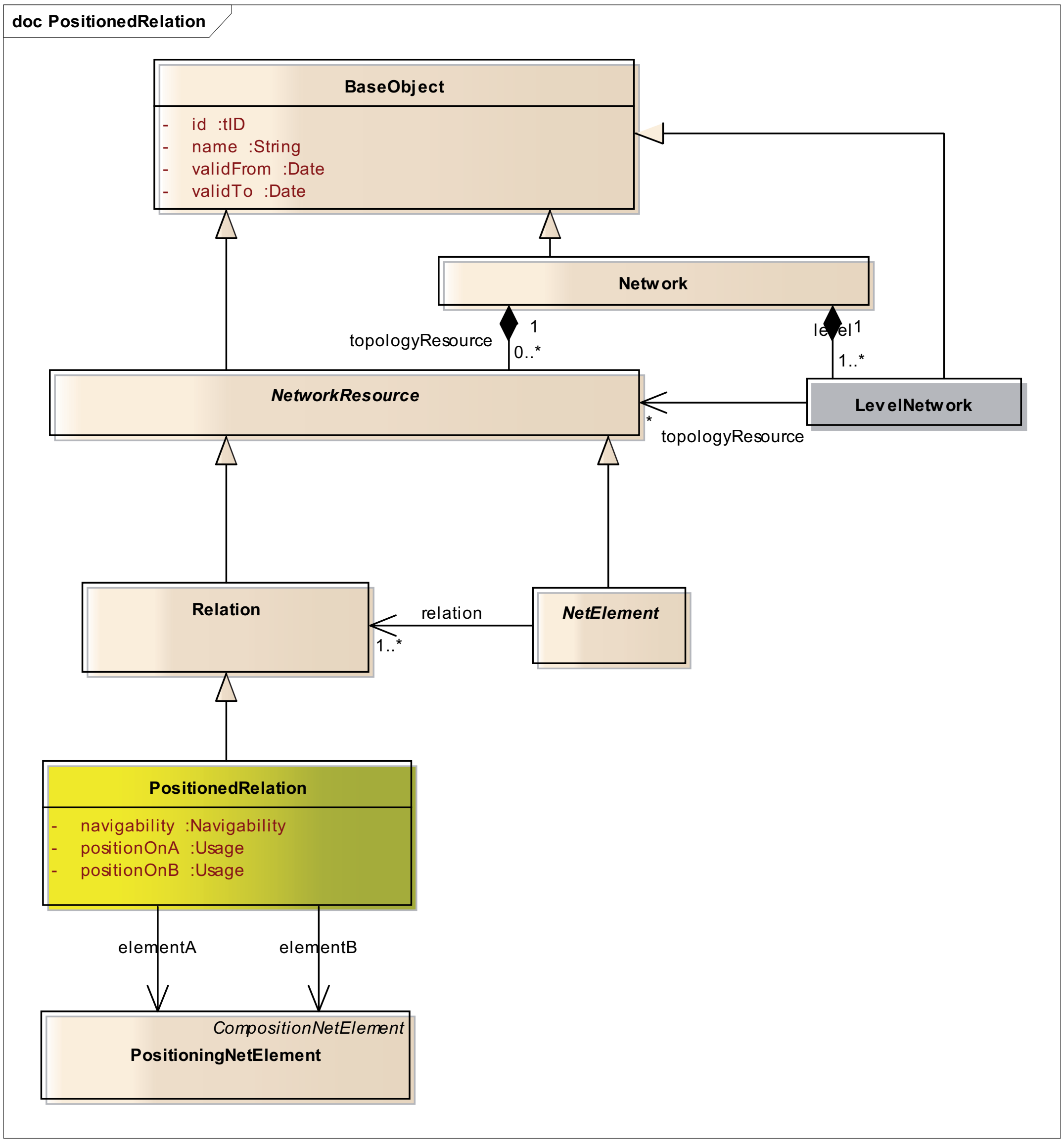
R.30100.121 The upper compartment of the UML-Class rectangle contains the name of current class in the centre (“PositionedRelation”).

R.30100.122 If the current class has a parent class, then the name of the parent class is found in the upper right corner of the upper compartment (“Relation”).

R.30100.123 Attributes of the current class are found at the top of the second compartment, mentioning the name of the attribute (e.g. “positionOnA”) and the type of the attribute (e.g. “Usage”).

R.30100.124 Attributes of the parent class (or parent classes of the parent class, up to the root class), if shown, are also contained in the second compartment below the name of class they belong to (e.g. “BaseObject”). Those inherited attributes are also shown with their name (“validFrom”) and their type (“Date”).

R.30100.125 If the current class possesses a parent class or an associated class then a second diagram containing all classes up to the root class and the associations is included.



R.30100.127 *Fig. 11 - Example of a class diagram containing all classes up to the root class*

R.30100.128 The second diagram type shows the full context of the current class within the model.

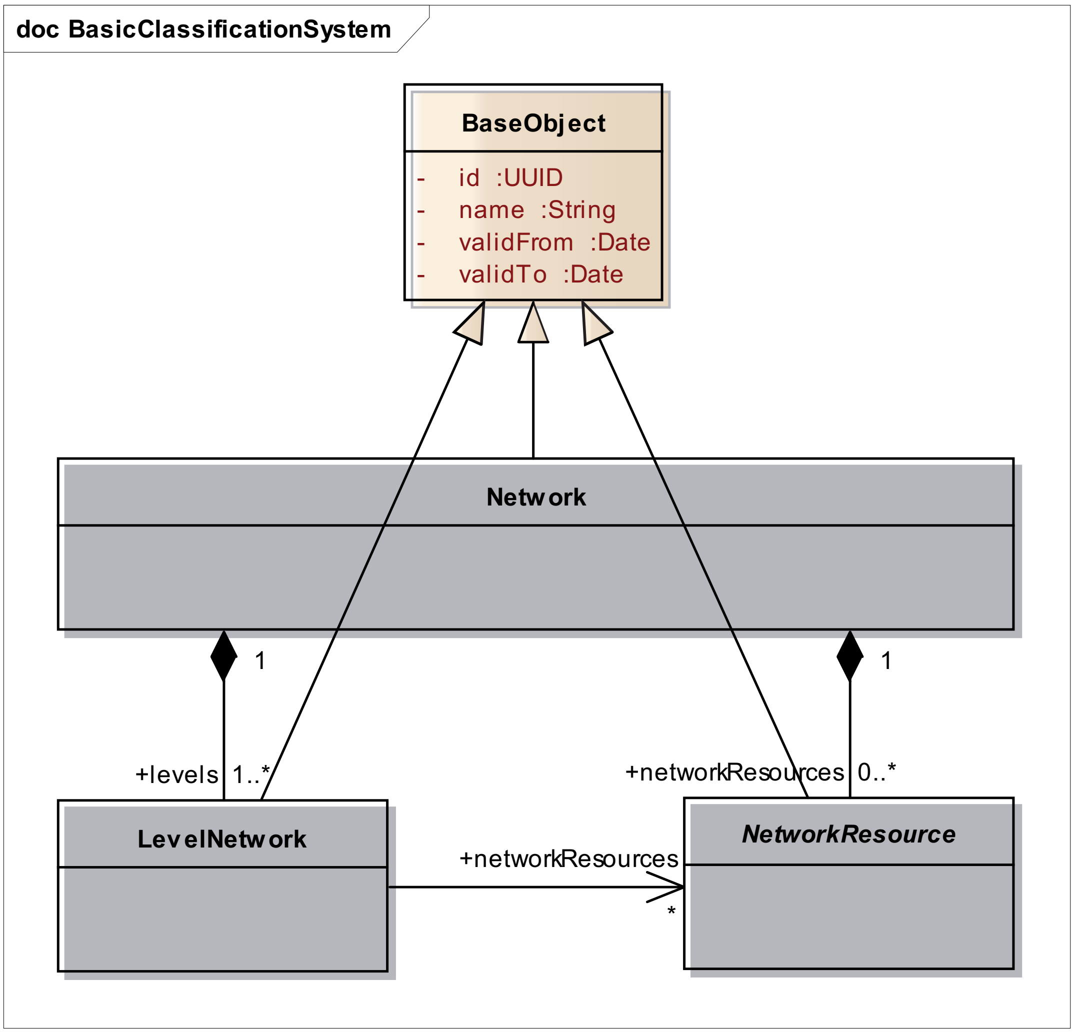
R.30100.129 The current class itself is depicted with the colour of the respective package. All generalizations (up to the root class) are shown. All associations of the current class and all associations of parent classes are shown. Attributes are always shown in the second compartment of the class they belong to.

2 - Package and class description

2.1 - Package: Base

2.1.1 - General

R.30100.133 The package “Base” is centred on the classes “Network” and “LevelNetwork”.



R.30100.135 *Fig. 12 - Base package overview*

R.30100.136 An instance of “Network” can be considered as a set of description levels (“LevelNetwork”) and a set of “NetworkResource” instances.

R.30100.137 “NetworkResource” depicts the building blocks of a “Network”, mainly

R.30100.138 - For Topology: “NetElement” (nodes), “Relation” (edges);

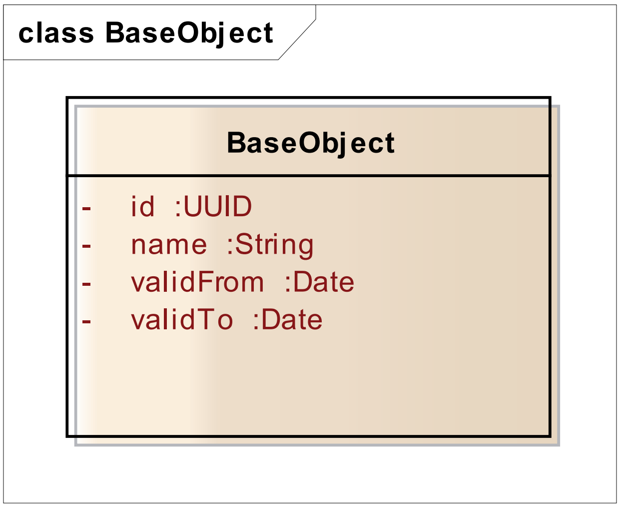
R.30100.139 - For Net Entities: “NetEntity” (e.g. physical assets, or speed limits), “EntityLocation” (their location), with the class “AssociatedNetElement” providing the link between assets and Topology;

R.30100.140 - The relevant positioning system, referred to via “AssociatedPositioningSystem”.

R.30100.141 An instance of “LevelNetwork” depicts a specific description level, as described under point 1.6 - page 11. The concept of description level is central to the goal of computational efficiency.

2.1.2 - BaseObject

R.30100.143 The base class "BaseObject" defines four properties shared by most objects in the RailTopoModel.



R.30100.145 *Fig. 13 - BaseObject*

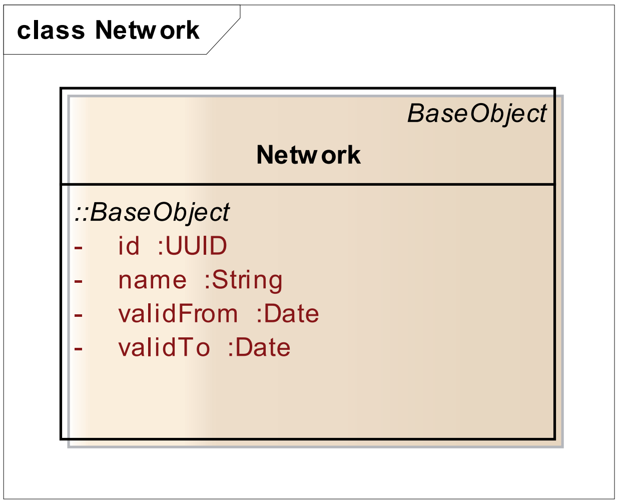
R.30100.146

Table 3 : BaseObject (Attributes)

Attributes		
id	UUID	Unique identifier. It is recommended to use an UUID whenever possible. For easy adaptation of existing systems, any other unique identifier is permitted.
name	String	Natural designation of the object.
validFrom	Date	Point in time where the object is available for usage for train operations (if empty, then the object is valid till the validTo date).
validTo	Date	Point in time where the object is no longer available for functional usage (if empty, then the object is valid since the validFrom date).

2.1.3 - Network

R.30100.164 The class “Network” defines the network being considered. It includes all resources that compose it (all Levels included), inter alia the topological, structural and positional properties exhibited by any railway network.

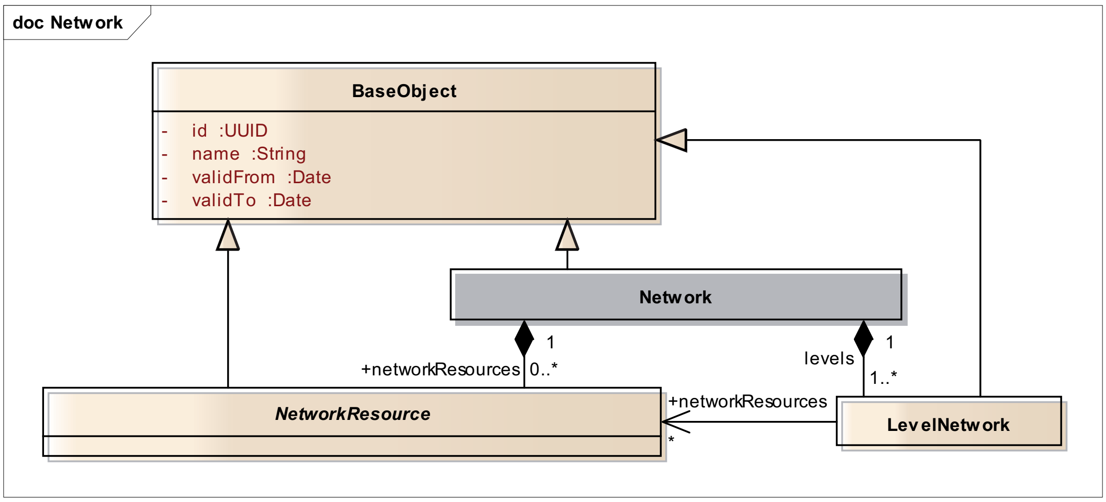


R.30100.166 *Fig. 14 - Network*

R.30100.167 The class "Network" is derived from "BaseObject".

R.30100.168 A "Network" is described in at least one "LevelNetwork". A "Network" may be described in more than one "LevelNetwork", typically for different levels of detail.

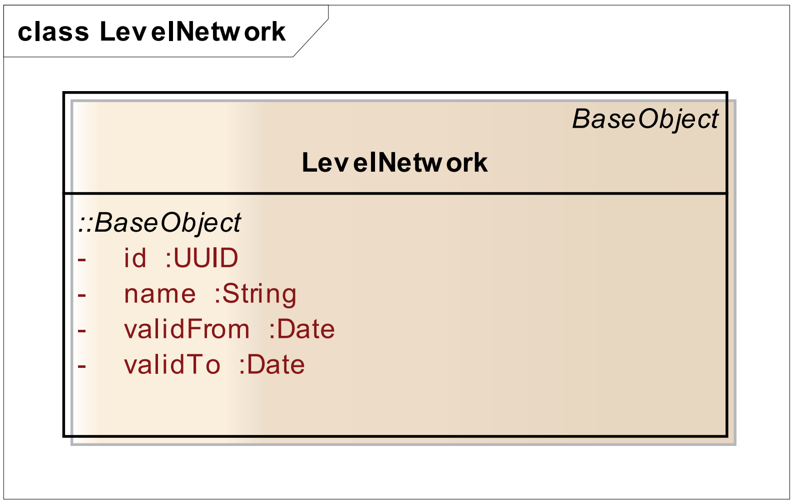
R.30100.169 Whenever an instance of "Network" is removed, all related "NetworkResource" instances and all related "LevelNetwork" instances are removed.



R.30100.171 *Fig. 15 - Network (Neighbourhood)*

2.1.4 - LevelNetwork

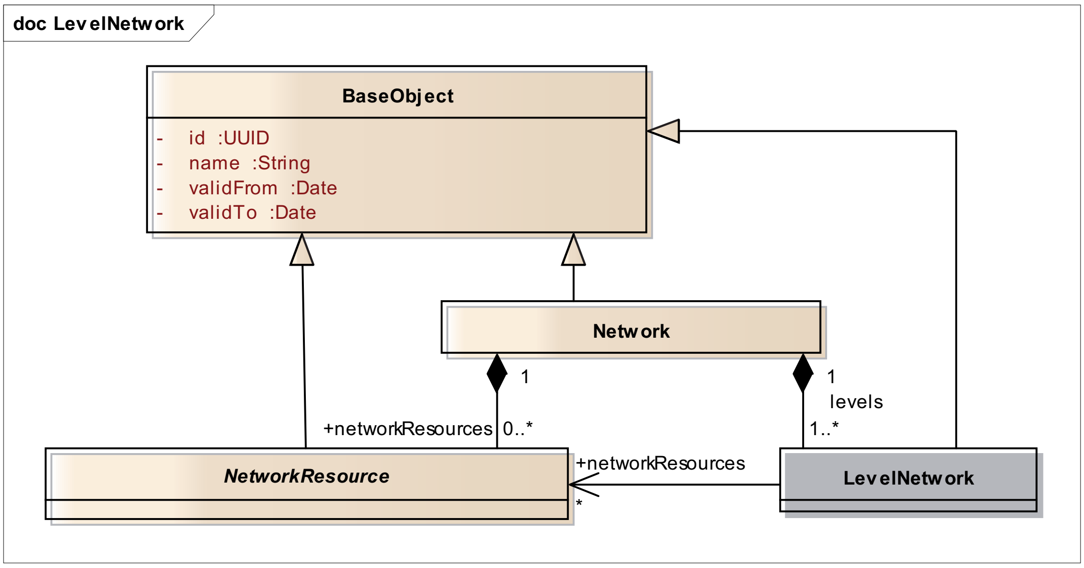
R.30100.173 The class "LevelNetwork" defines a consistent "view" of a Network at a certain level of granularity. An instance of this class therefore includes all resources that are required to define the corresponding level (e.g. micro/track, or macro/line).



R.30100.175 *Fig. 16 - LevelNetwork*

R.30100.176 The class "LevelNetwork" is derived from "BaseObject".

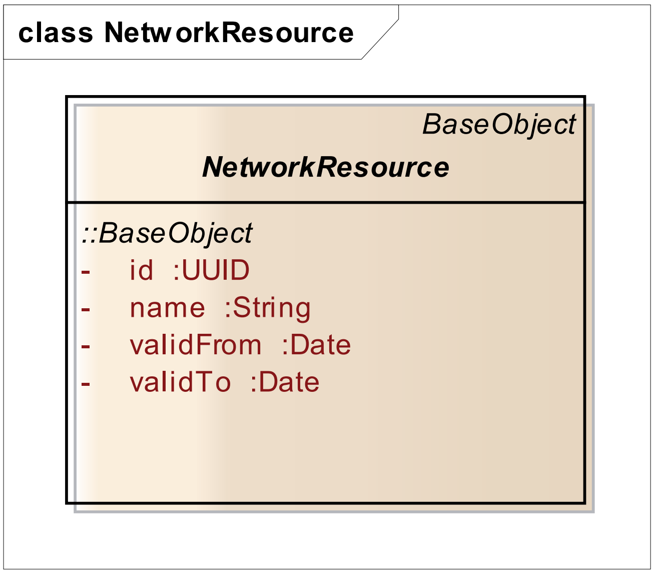
R.30100.177 A "LevelNetwork" belongs to exactly one "Network".



R.30100.179 *Fig. 17 - LevelNetwork (Neighbourhood)*

2.1.5 - NetworkResource

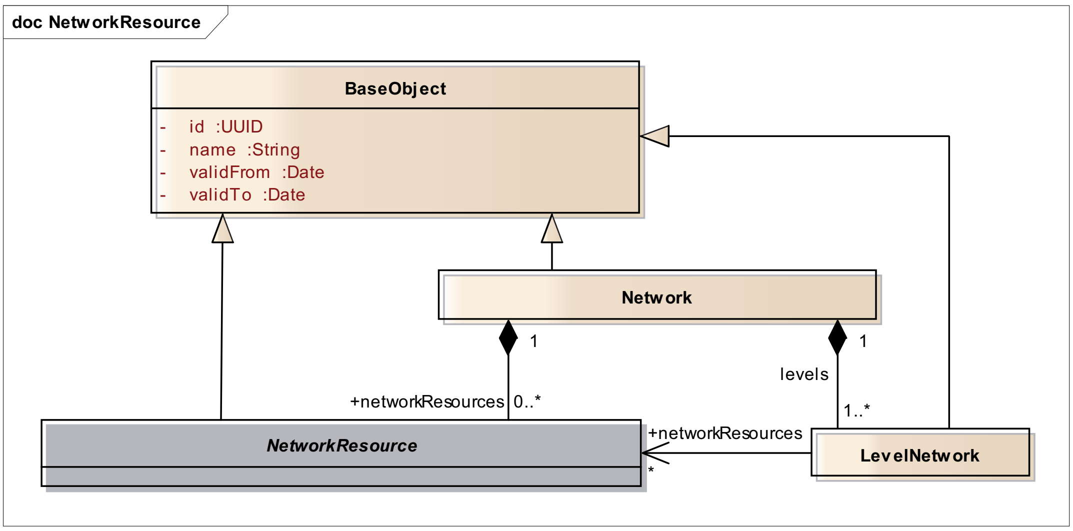
R.30100.181 Every object of the network is qualified as a resource. The class "NetworkResource" defines this concept.



R.30100.183 *Fig. 18 - NetworkResource*

R.30100.184 The class "NetworkResource" is derived from "BaseObject".

R.30100.185 A "NetworkResource" belongs to exactly one "Network".

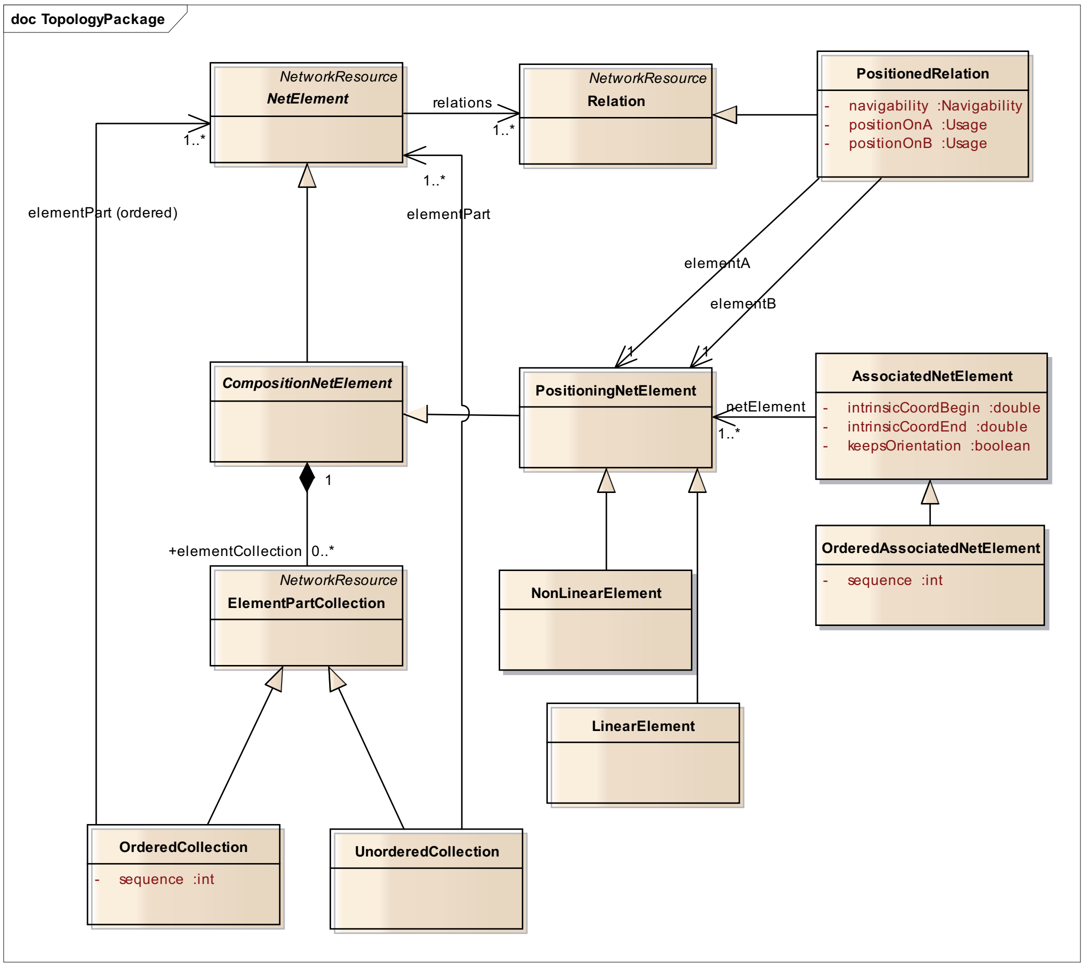


R.30100.187 *Fig. 19 - NetworkResource (Neighbourhood)*

2.2 - Package: Topology

2.2.1 - General

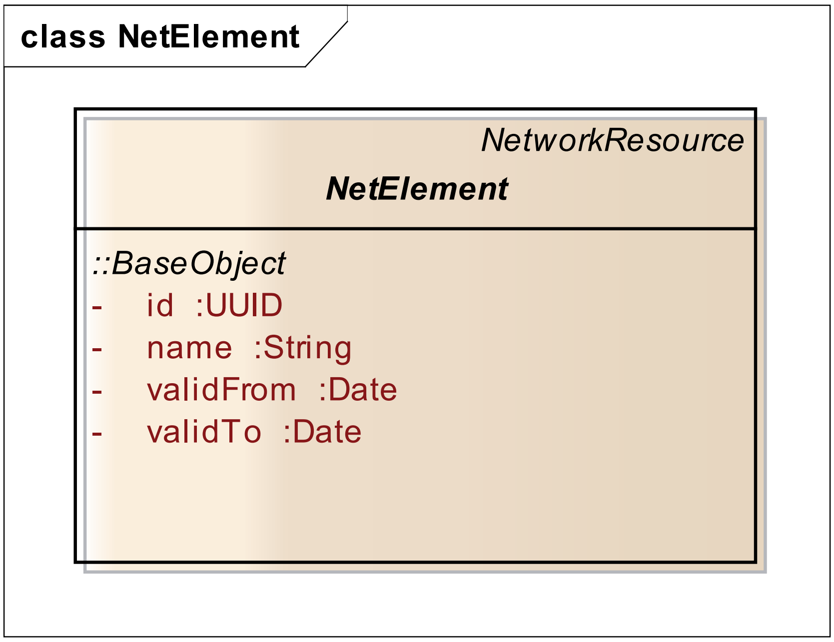
R.30100.190 The Topology package essentially applies Graph theory concepts to the RailTopoModel. Nodes are conceptually embodied by “NetElement”, and edges by “Relation”. “CompositionNetElement” is the class, derived from “NetElement” that will ultimately allow the assembly of nodes into bigger nodes, and zooming in and out from one level to another.



R.30100.192 *Fig. 20 - Topology package overview*

2.2.2 - NetElement

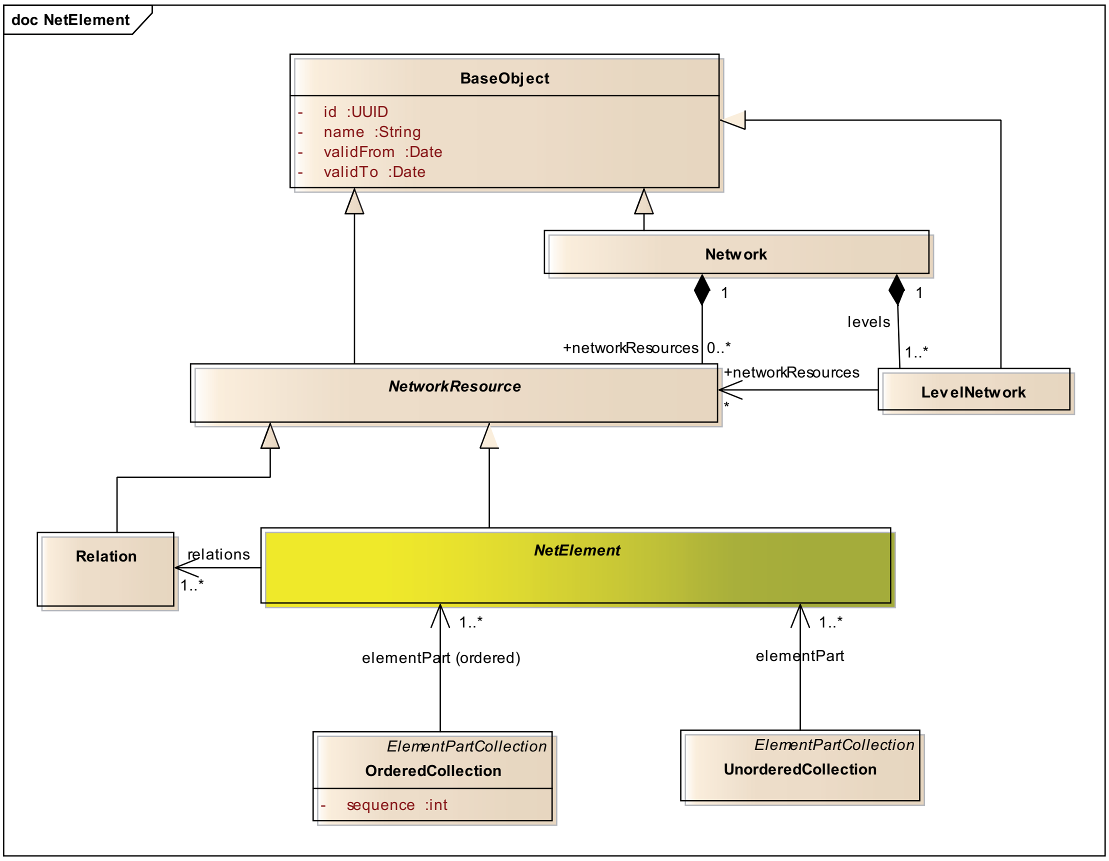
R.30100.194 The class "NetElement" defines the base member of topology in a connexity graph of a network (at any level).



R.30100.196 *Fig. 21 - NetElement*

R.30100.197 The class "NetElement" is derived from "NetworkResource".

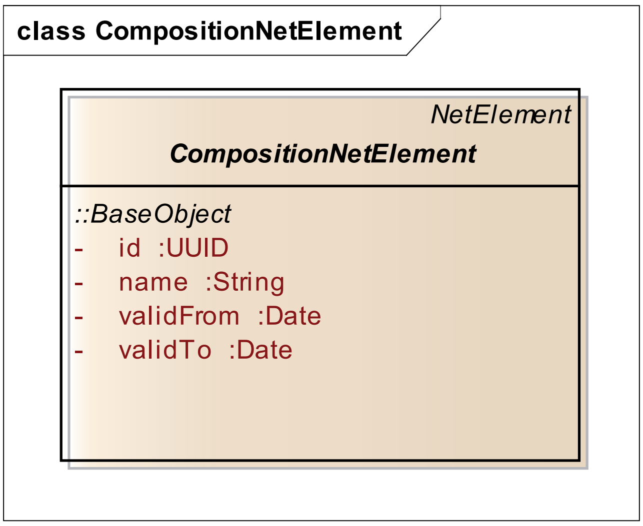
R.30100.198 Each "NetElement" takes part in one or many "Relation" with other "NetElement" instances.



R.30100.200 *Fig. 22 - NetElement (Neighbourhood)*

2.2.3 - CompositionNetElement

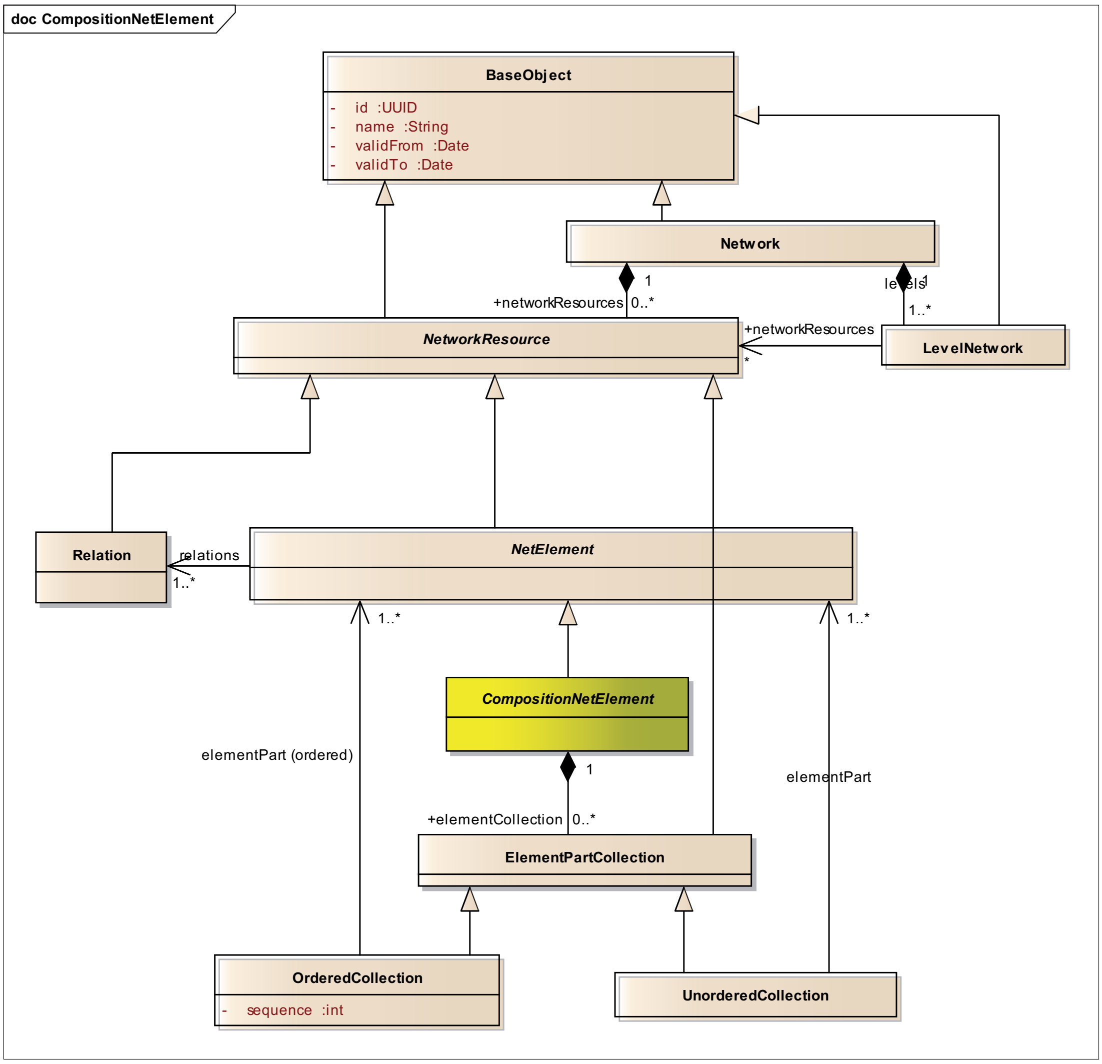
R.30100.202 The class "CompositionNetElement" carries the generic concept of topological aggregation. It defines a topological element that aggregates some other topological element from another level (e.g. a macro element aggregates micro elements).



R.30100.204 *Fig. 23 - CompositionNetElement*

R.30100.205 The class "CompositionNetElement" is derived from "NetElement".

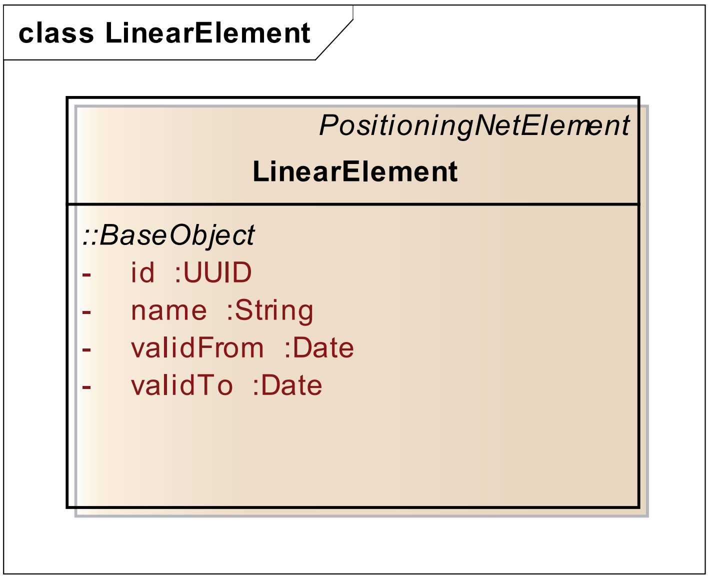
R.30100.206 Whenever an instance "CompositionNetElement" is removed, all related "ElementPartCollection" instances are removed.



R.30100.208 *Fig. 24 - CompositionNetElement (Neighbourhood)*

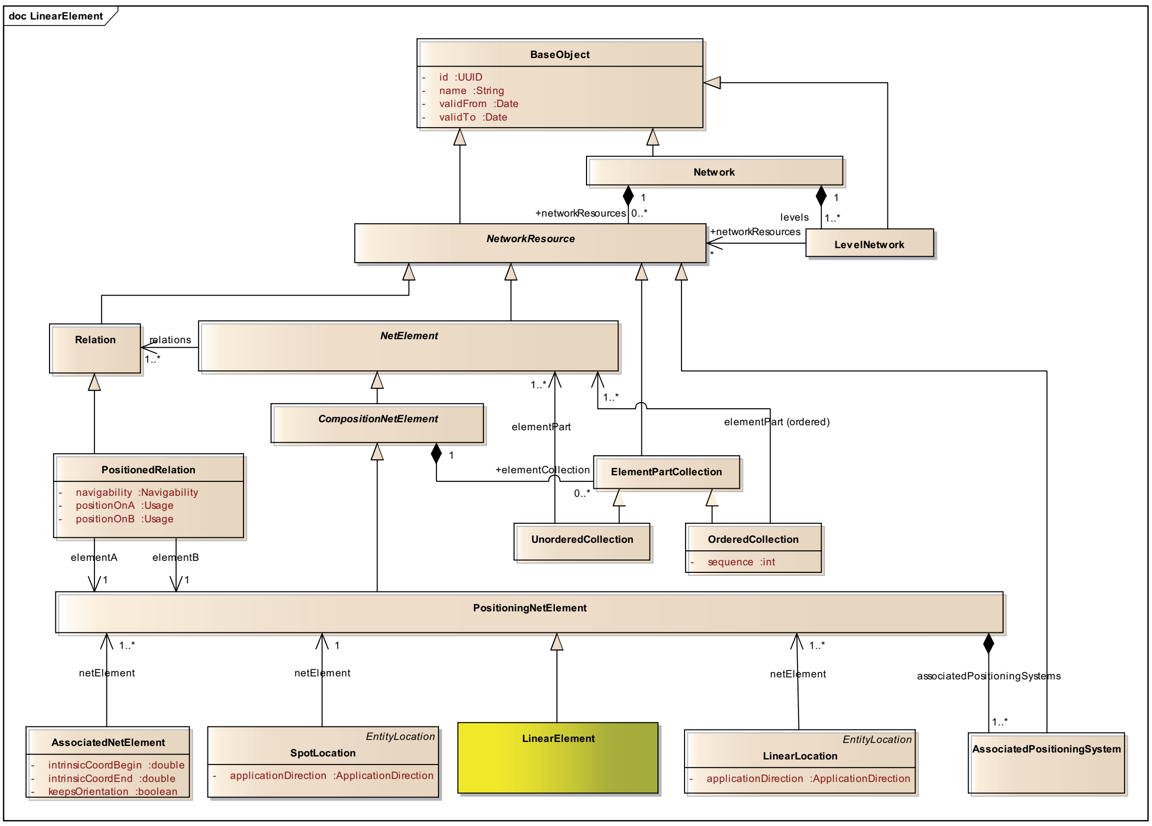
2.2.4 - LinearElement

R.30100.210 The class "LinearElement" defines "PositioningNetElement" instances that are one-dimensional.



R.30100.212 *Fig. 25 - LinearElement*

R.30100.213 The class "LinearElement" is derived from "PositioningNetElement”.



R.30100.215 *Fig. 26 - LinearElement (Neighbourhood)*

R.30100.216 Notes:

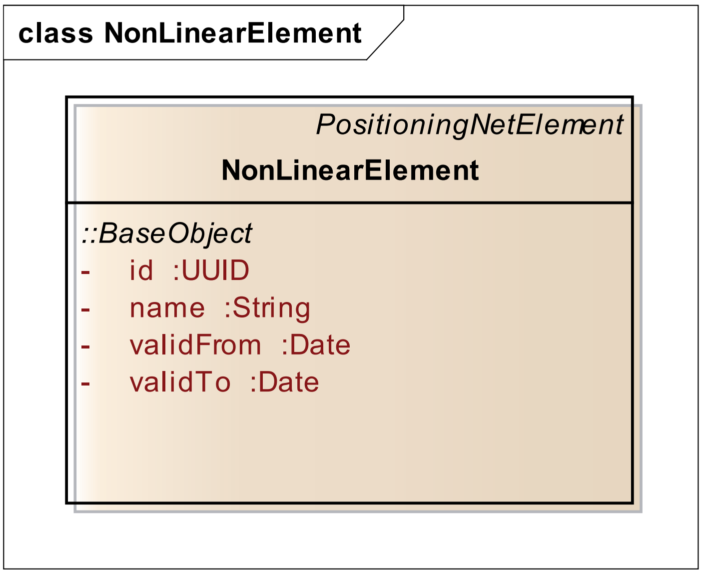
R.30100.217 - The classes “Trail” and “SectionOfLine”, shown on the overall simplified class diagram (see point 1.7.3 - page 17), are derived from “LinearElement”. These two classes are not part of the RailTopoModel; they are provided here for illustration purposes

R.30100.218 - A “Trail” is an uninterrupted track between two adjacent switches, or between a switch and an adjacent buffer stop. “Uninterrupted” means that there are no other switches in that connection. Therefore, the class “Trail” can represent nodes at Micro level (see point 1.6.2 - page 12), according to Graph theory and to the modelling principles presented under point 1.5 - page 10

R.30100.219 - Similarly, a “SectionOfLine”, being a line section between two adjacent Operational Points, would be an important class of nodes to be used at Macro level (see point 1.6.4 - page 13).

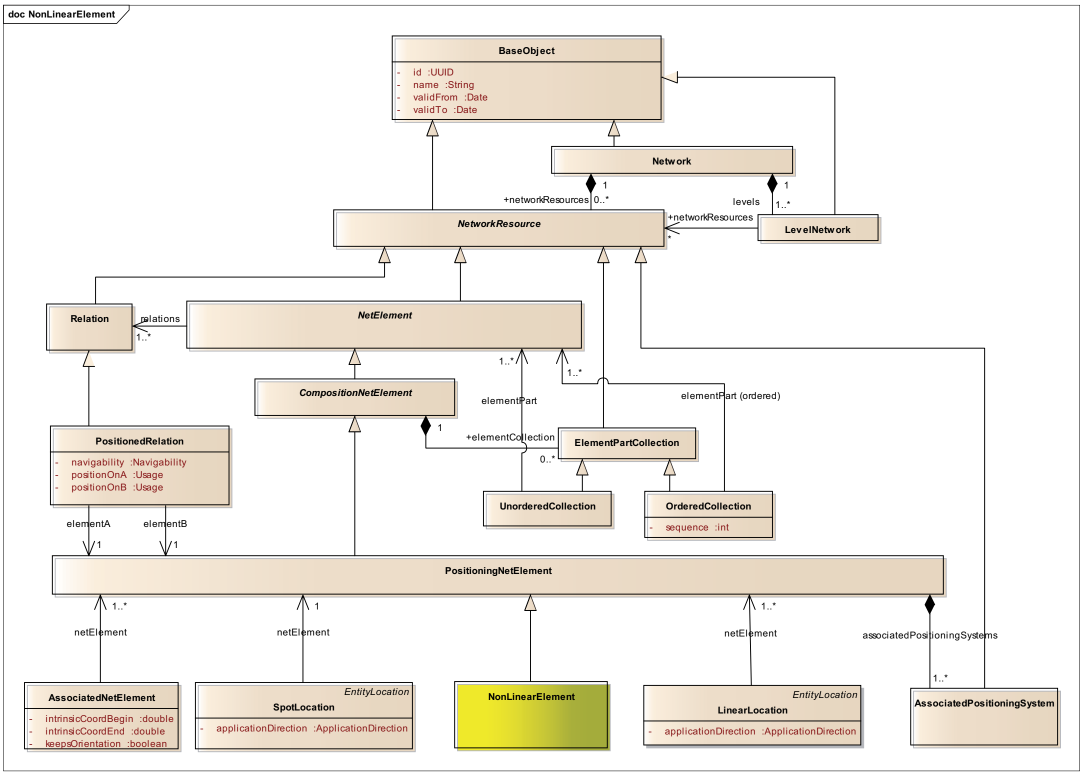
2.2.5 - NonLinearElement

R.30100.221 The class "NonLinearElement" defines "PositioningNetElement" instances with no dimensions (spots).



R.30100.223 *Fig. 27 - NonLinearElement*

R.30100.224 The class "NonLinearElement" is derived from "PositioningNetElement".



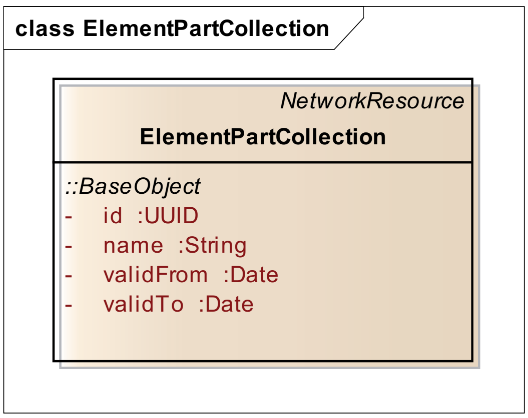
R.30100.226 *Fig. 28 - NonLinearElement (Neighbourhood)*

R.30100.238 Note:

R.30100.239 The classes “OperationalPoint” and “NetLimit”, derived from “NonLinearElement”, are not part of the RailTopoModel. These are provided as two examples of classes that may represent nodes (in the sense of Graph theory) at macro level.

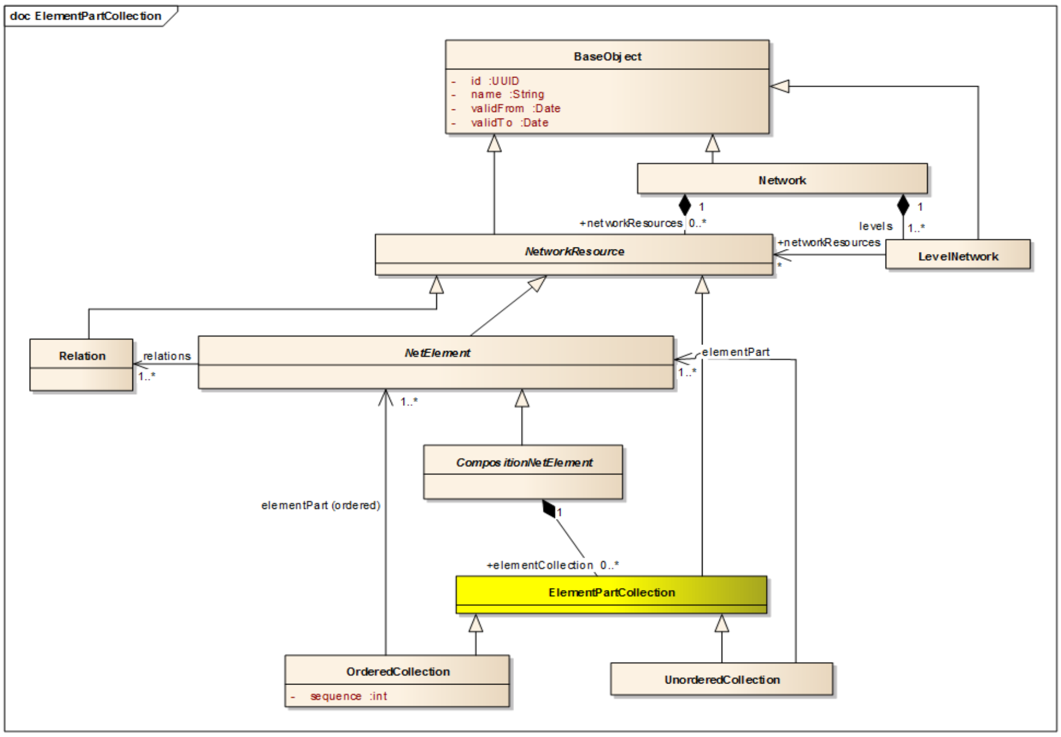
2.2.6 - ElementPartCollection

R.30100.241 The class "ElementPartCollection" defines the collection of Net elements to be aggregated into the higher level NetElement (Generic class).



R.30100.243 *Fig. 29 - ElementPartCollection*

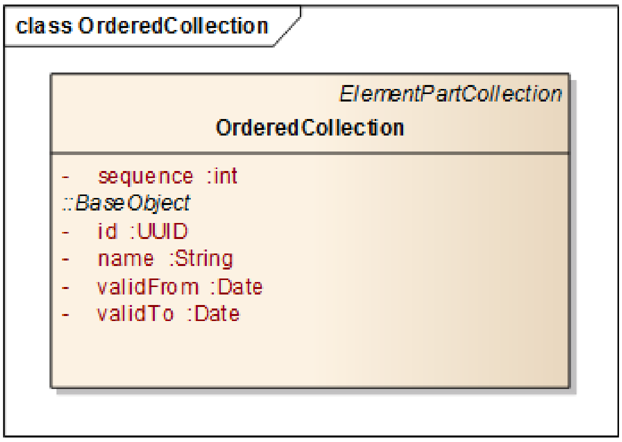
R.30100.244 The class "ElementPartCollection" is derived from "NetworkResource". An "ElementPartCollection" instance belongs to exactly one "CompositionNetElement" instance.



R.30100.246 *Fig. 30 - ElementPartCollection (Neighbourhood)*

2.2.7 - OrderedCollection

R.30100.248 The class "OrderedCollection" is a subclass of ElementPartCollection, dedicated to ordered NetElements (required to build a route).



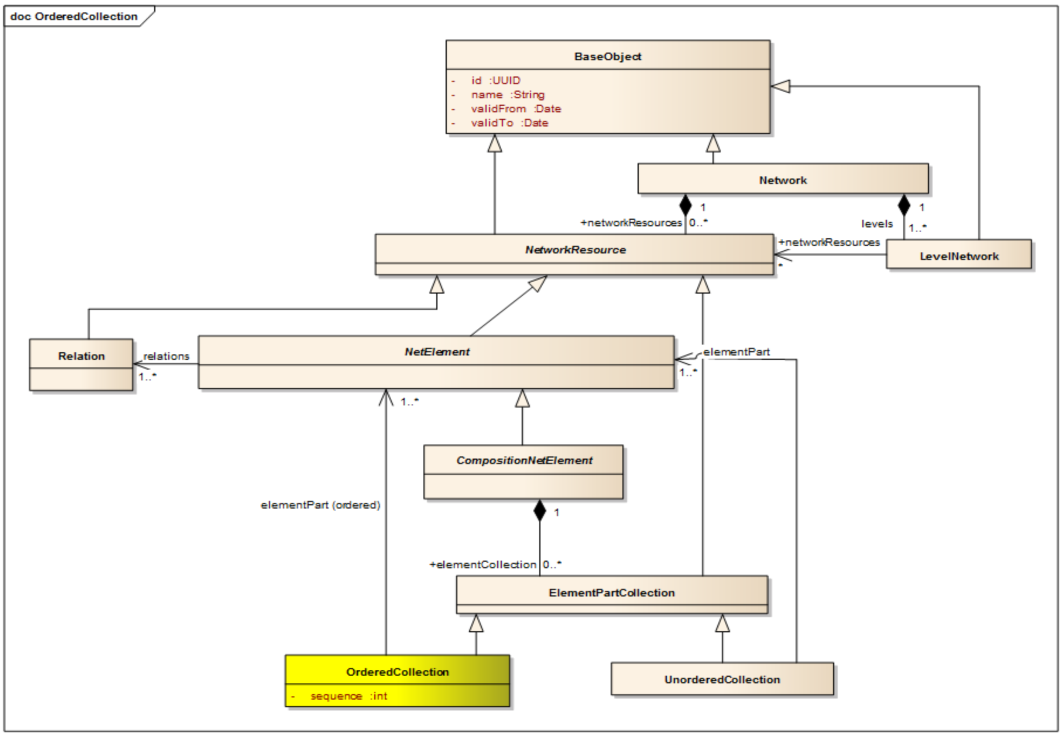
R.30100.250 *Fig. 31 - OrderedCollection*

R.30100.251

Table 4 : OrderedCollection (Attributes)

Attributes		
sequence	Int	Sequence of the child element within the ordered collection

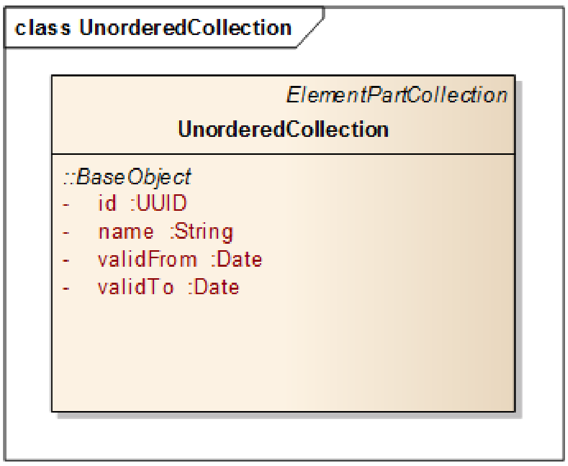
R.30100.259 The class "OrderedCollection" is derived from "ElementPartCollection".



R.30100.261 *Fig. 32 - OrderedCollection (Neighbourhood)*

2.2.8 - UnorderedCollection

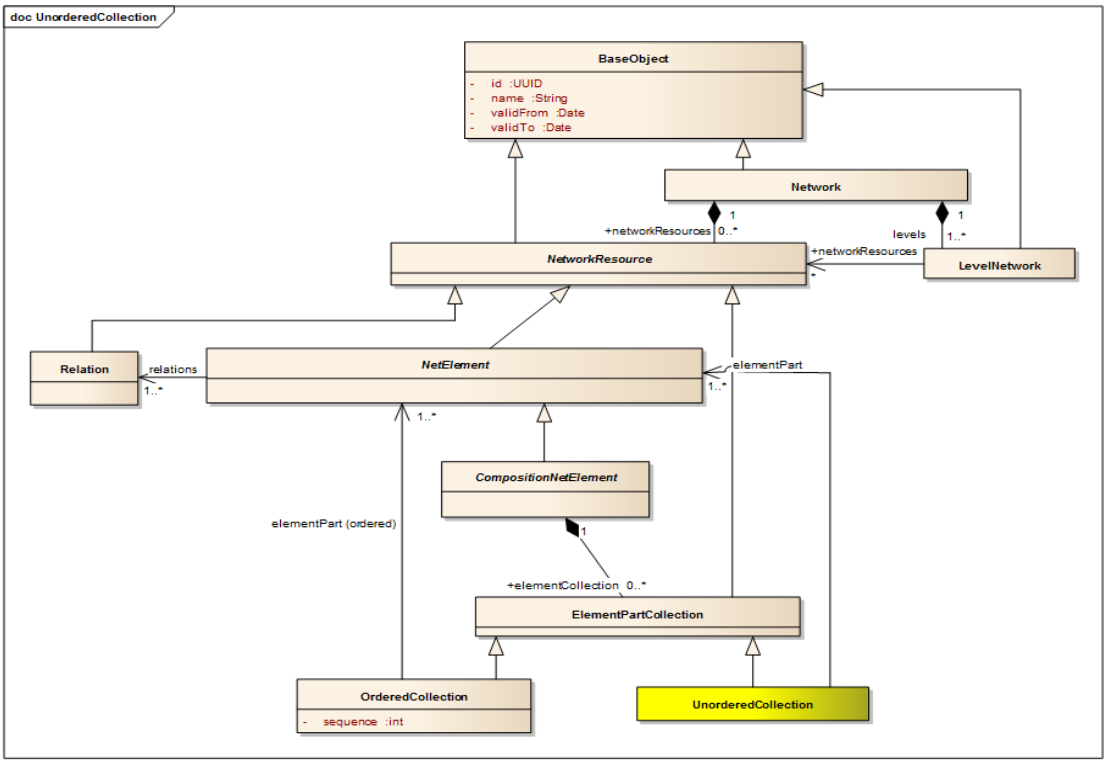
R.30100.263 The class "UnorderedCollection" is a subclass of ElementPartCollection that is dedicated to unordered NetElements (bulk list without need for routes).



R.30100.265 *Fig. 33 - UnorderedCollection*

R.30100.266 The child "NetElement" instances possess no ordering property.

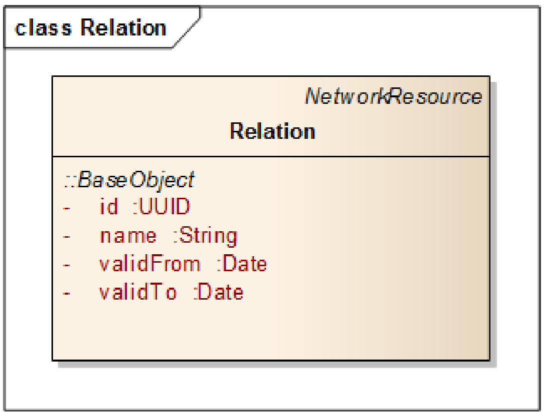
R.30100.267 The class "UnorderedCollection" is derived from "ElementPartCollection".



R.30100.269 *Fig. 34 - UnorderedCollection (Neighbourhood)*

2.2.9 - Relation

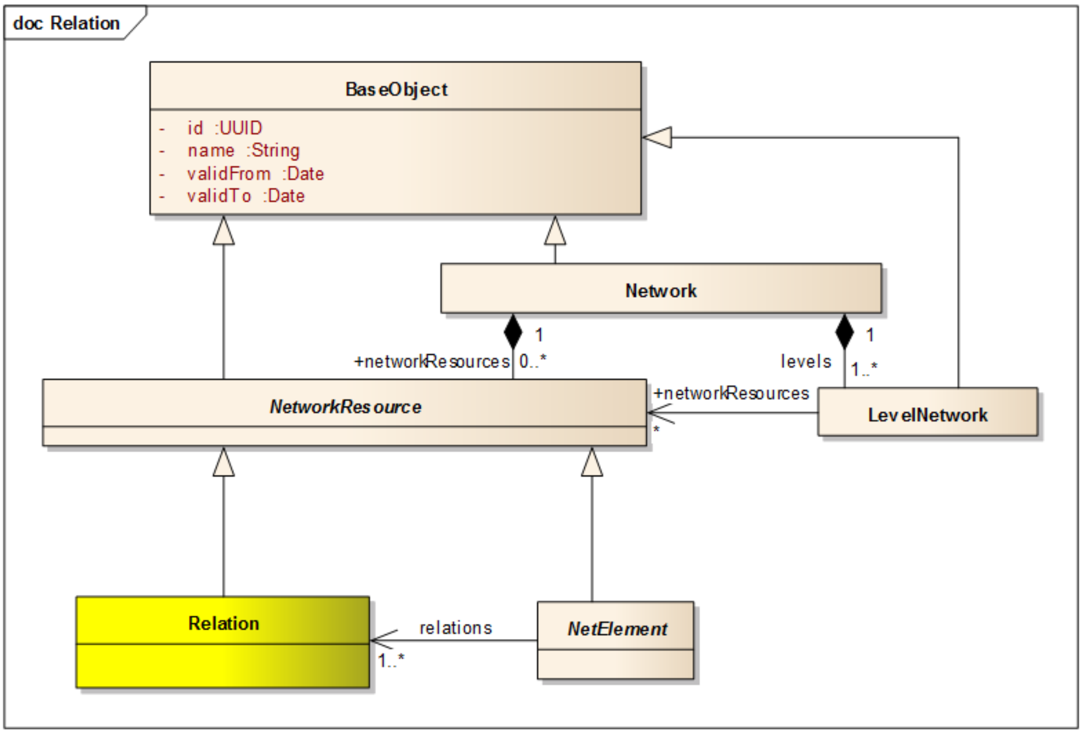
R.30100.271 The class "Relation" defines the connexity relation between two NetElements in the connexity graph of the network.



R.30100.273 *Fig. 35 - Relation*

R.30100.274 The class "Relation" is derived from "NetworkResource".

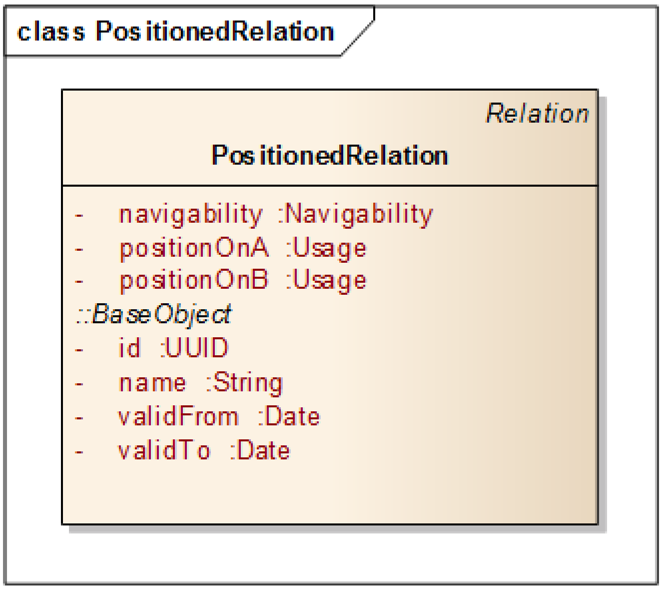
R.30100.275 In a functional railway network, each instance of "Relation" typically brings together two "NetElement" instances. “Relation” can be seen as the base class to define edges in the sense of Graph theory.



R.30100.277 *Fig. 36 - Relation (Neighbourhood)*

2.2.10 - PositionedRelation

R.30100.279 The class "PositionedRelation" is a subclass of “Relation”, defining an oriented relation between exactly two PositioningNetElements.



R.30100.281 *Fig. 37 - PositionedRelation*

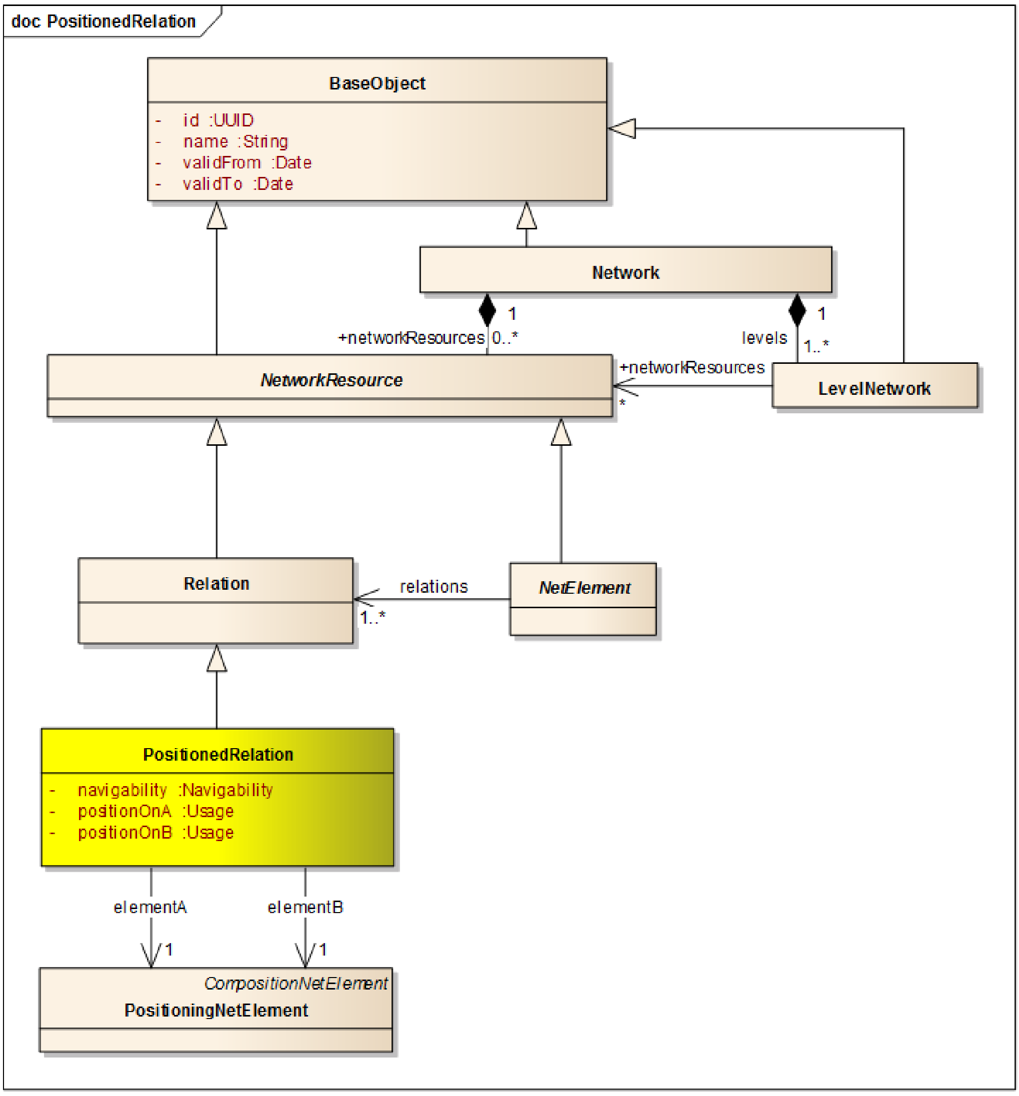
R.30100.282 The class "PositionedRelation" is derived from "Relation".

R.30100.283 One connected "NetElement" is designated code "A", the other connected "NetElement" is designated code "B".

R.30100.284

Table 5 : PositionedRelation (Attributes)

Attributes			
navigability	Navigability	AB	it is possible to move a train from NetElement "A" to NetElement "B". It is not possible to move it from NetElement B to NetElement A
		BA	it is possible to move a train from NetElement "B" to NetElement "A". It is not possible to move it from NetElement A to NetElement B
		Both	it is possible to move a train from "A" to "B" as well as from "B" to "A".
		None	it is not possible to move a train across this "Relation" in any direction.
positionOnA	Usage	0	the "Relation" is using the start of NetElement A
		1	the "Relation" is using the end of NetElement A
positionOnB	Usage	0	the "Relation" is using the start of NetElement B
		1	the "Relation" is using the end of NetElement B

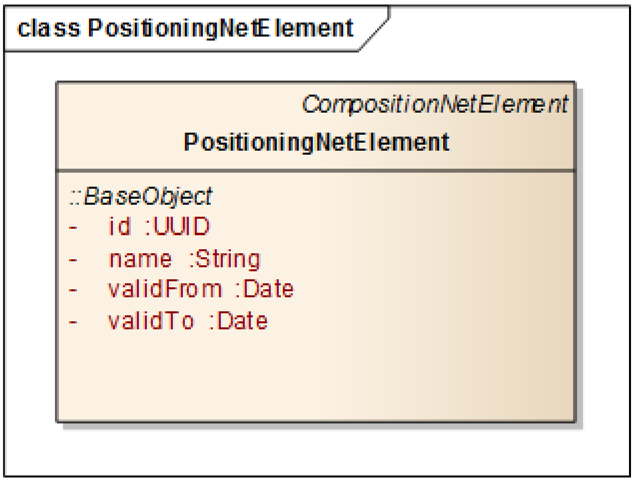


R.30100.323

Fig. 38 - PositionedRelation (Neighbourhood)

2.2.11 - PositioningNetElement

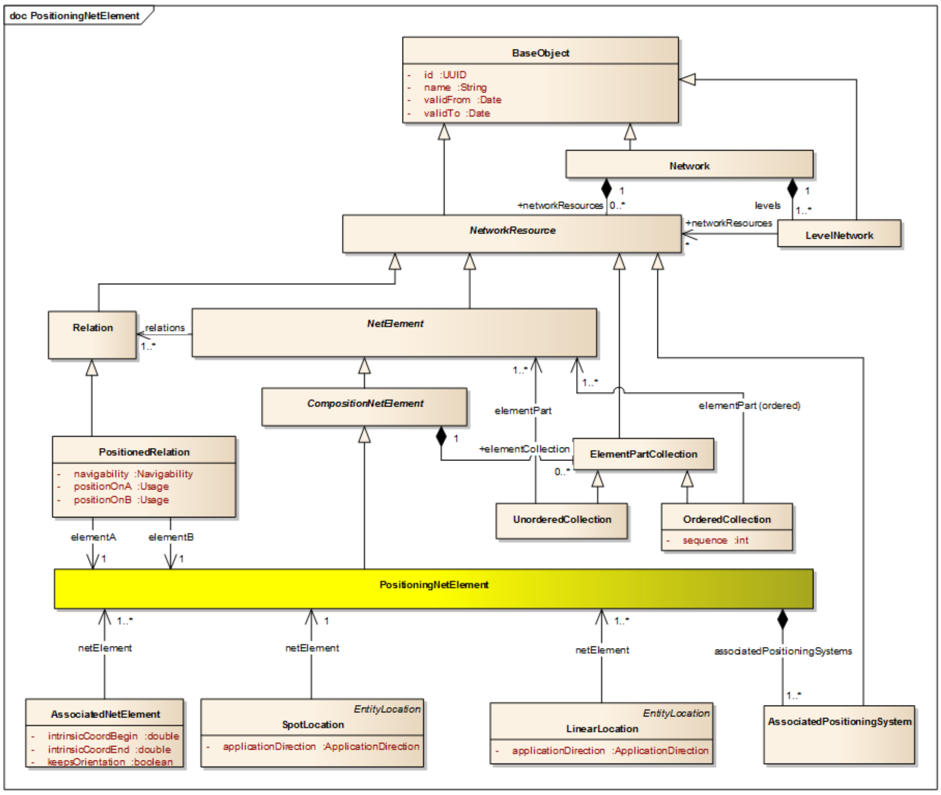
R.30100.325 The class "PositioningNetElement" defines a NetElement requiring at least one Positioning System, with orientation (carried by IntrinsicCoordinate).



R.30100.327 *Fig. 39 - PositioningNetElement*

R.30100.328 The class "PositioningNetElement" is derived from "CompositionNetElement". Each "PositioningNetElement" contains at least one instance of "AssociatedPositioningSystem".

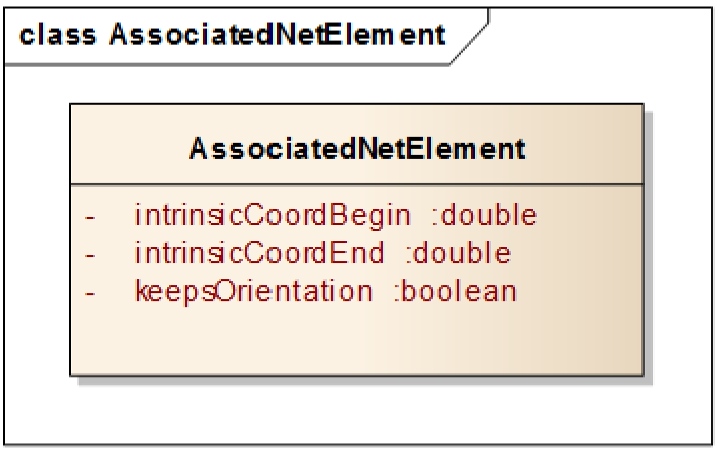
R.30100.329 Whenever an instance of "PositioningNetElement" is removed, all related "AssociatedPositioningSystem" instances are removed.



R.30100.331 *Fig. 40 - PositioningNetElement (Neighbourhood)*

2.2.12 - AssociatedNetElement

R.30100.333 The class "AssociatedNetElement" defines topological structures and location information in relation between "NetElement" instances and in relation between one "NetElement" instance and location information for "NetEntity" instances.

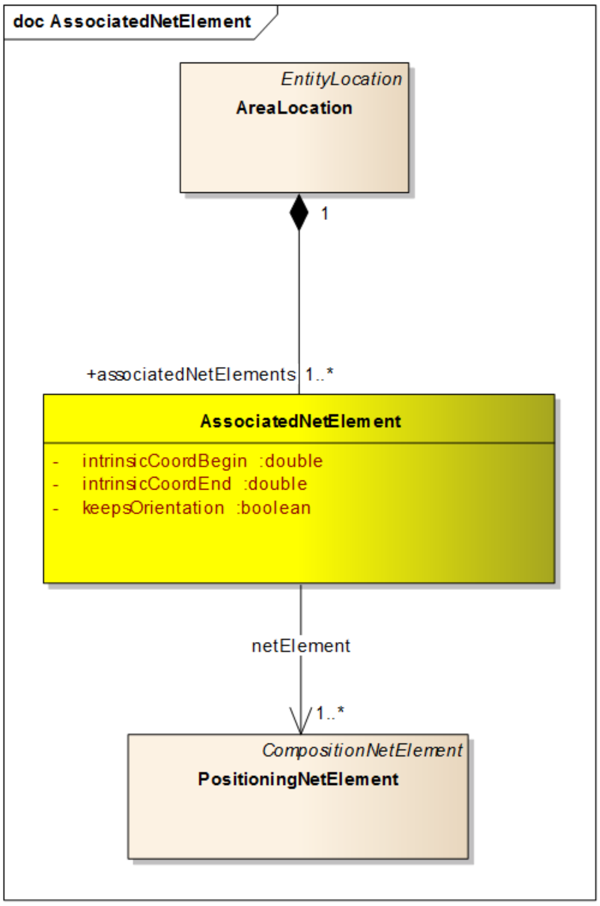


R.30100.335 *Fig. 41 - AssociatedNetElement*

R.30100.336 The class "AssociatedNetElement" has no specific parent class.

Table 6 : AssociatedNetElement (Attributes)

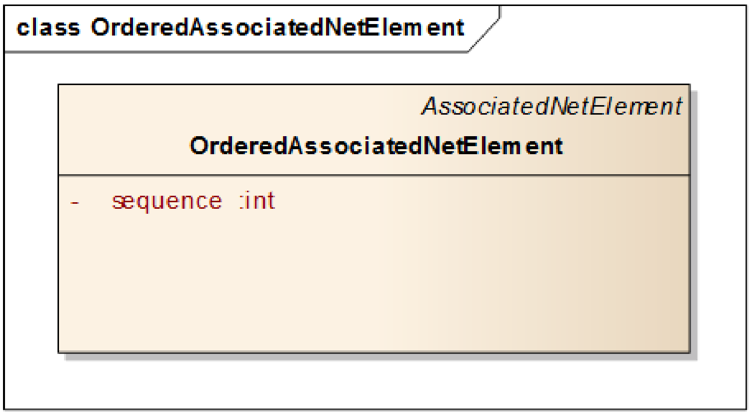
Attributes			
intrinsicCoordBegin	double	Start location of the "NetEntity" instance in relation to the "PositioningNetElement" which is used for positioning within the network.	
intrinsicCoordEnd	double	End location of the "NetEntity" instance in relation to the "PositioningNetElement" which is used for positioning within the network.	
keeps Orientation	boolean	Child LinearElement keeps same Orientation as parent LinearElement	0 (false) : Orientation is not relevant 1 (true) : Orientation is relevant



R.30100.356 *Fig. 42 - AssociatedNetElement (Neighbourhood)*

2.2.13 - OrderedAssociatedNetElement

R.30100.358 The class "OrderedAssociatedNetElement" defines the ordered sequences of "AssociatedNetElement" instances which together describe the complete structure of a "LinearLocation" instance.

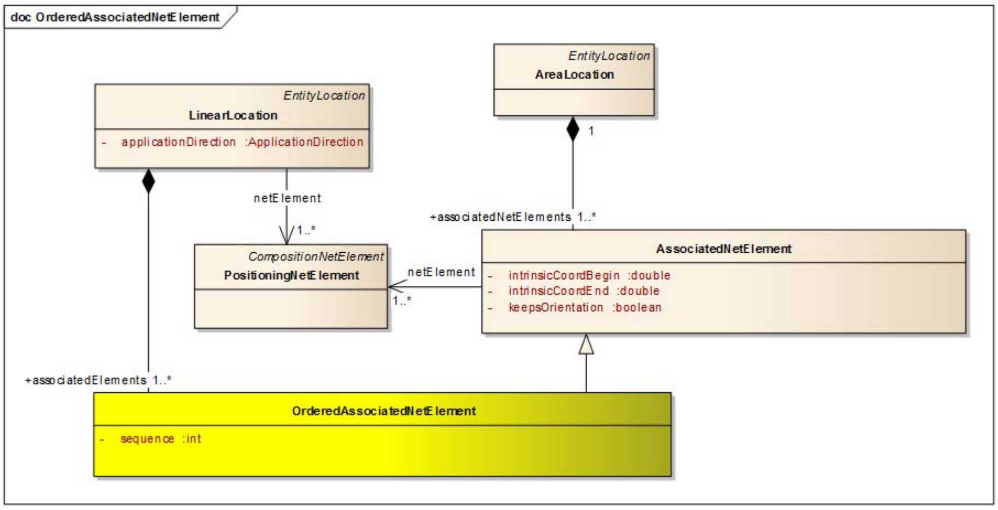


R.30100.360 *Fig. 43 - OrderedAssociatedNetElement*

R.30100.361

Table 7 : OrderedAssociatedNetElement (Attributes)

Attributes		
Sequence	Int	



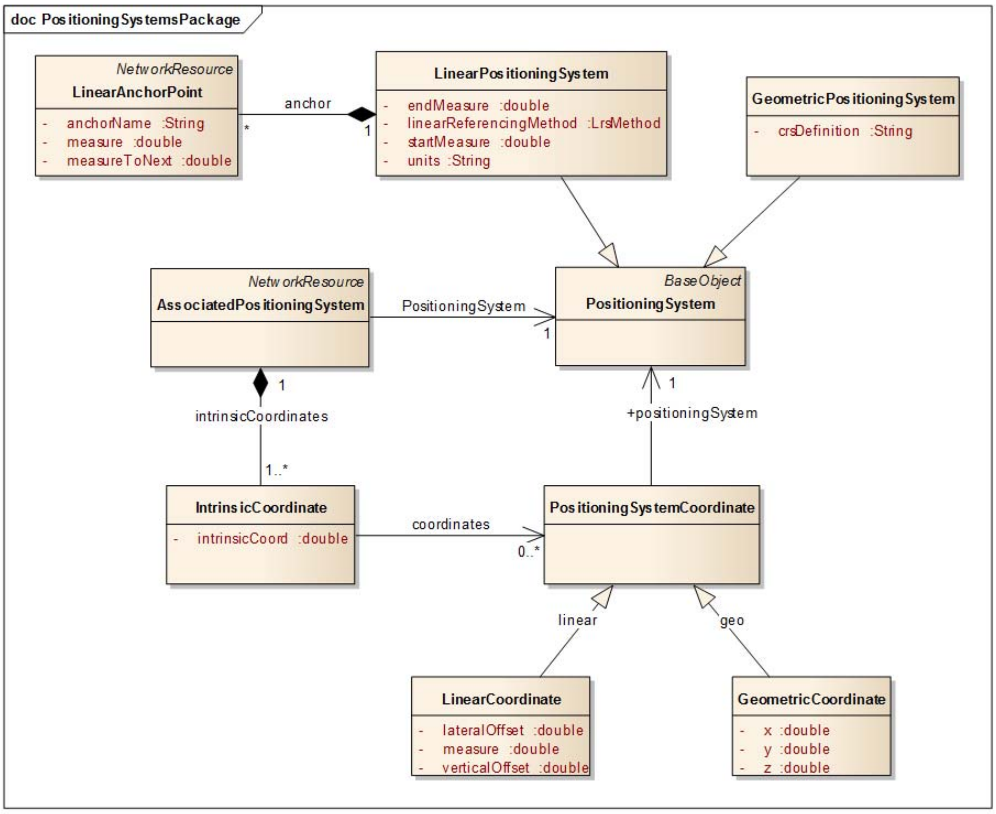
R.30100.370

Fig. 44 - OrderedAssociatedNetElement (Neighbourhood)

2.3 - Package: Positioning Systems

2.3.1 - General

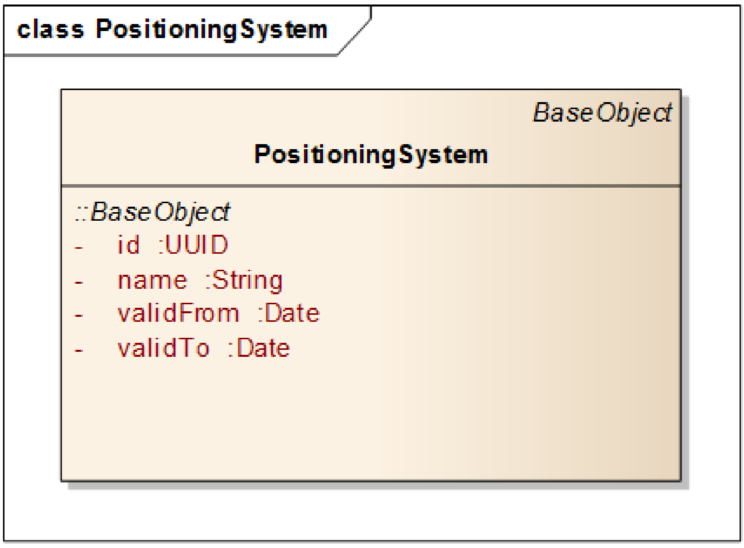
R.30100.373 The “Positioning Systems” package offers a catalogue of positioning methods, which currently fall into three categories: intrinsic, linear and geographic. The network topology may use any of these, only the intrinsic positioning being mandatory.



R.30100.375 *Fig. 45 - Positioning systems package overview*

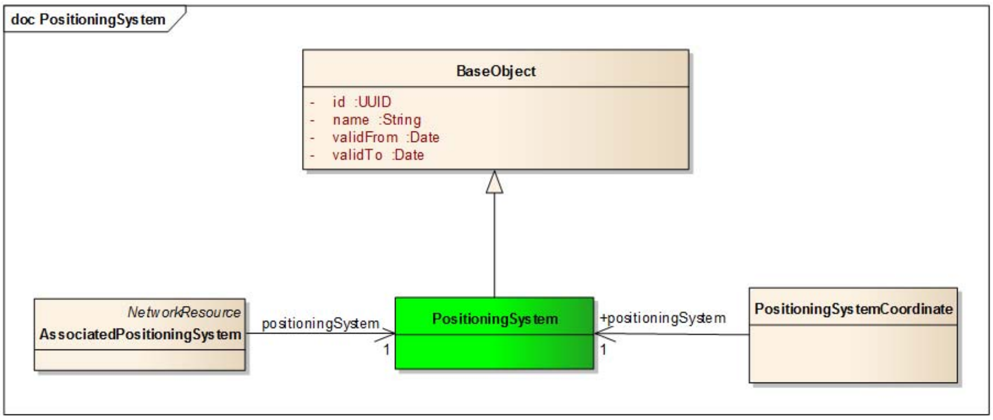
2.3.2 - PositioningSystem

R.30100.377 The class "PositioningSystem" defines the generic concept of a positioning system.



R.30100.379 *Fig. 46 - PositioningSystem*

R.30100.380 The class "PositioningSystem" is derived from "BaseObject".



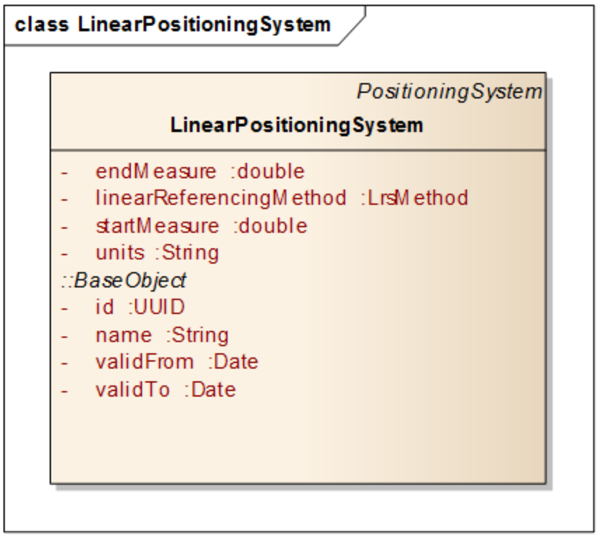
R.30100.382 *Fig. 47 - PositioningSystem (Neighbourhood)*

2.3.3 - LinearPositioningSystem

R.30100.384 The class "LinearPositioningSystem" defines a "PositioningSystem" where a "line of reference" together with a single number allows a location within a railway network to be defined.

R.30100.385 In railway business a “line of reference” is very often represented with a line number or a track number together with a start mileage and an end mileage.

R.30100.386 Note: RailTopoModel makes no assumption about the nature of the “line of reference”.



R.30100.388 *Fig. 48 - LinearPositioningSystem*

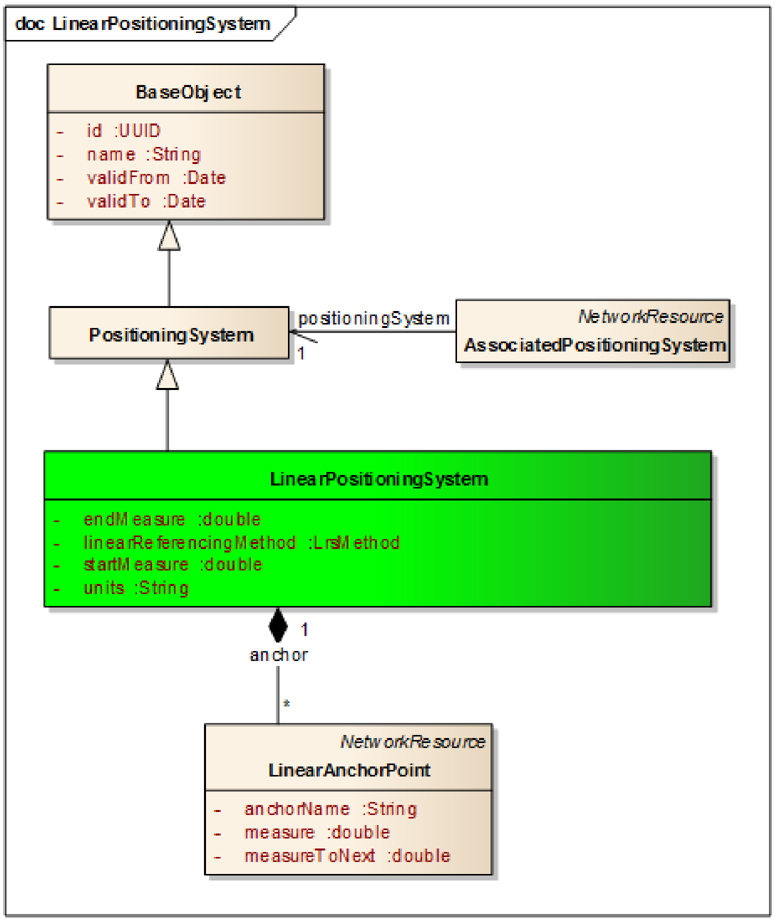
R.30100.389 The class "LinearPositioningSystem" is derived from "PositioningSystem".

R.30100.390

Table 8 : LinearPositioningSystem (Attributes)

Attributes			
linearReferencingMethod	LrsMethod	Method for linear referencing	absolute
			relative
			interpolation
startMeasure	Double	Value for measurement at the beginning of the “LinearPositioningSystem”	
endMeasure	Double	Value for measurement at the end of the “LinearPositioningSystem”	
units	String	Units for measurement (e.g. kilometre, metre, mile)	

R.30100.420 Whenever an instance of "LinearPositioningSystem" is removed, all related "LinearAnchorPoint" instances are removed.

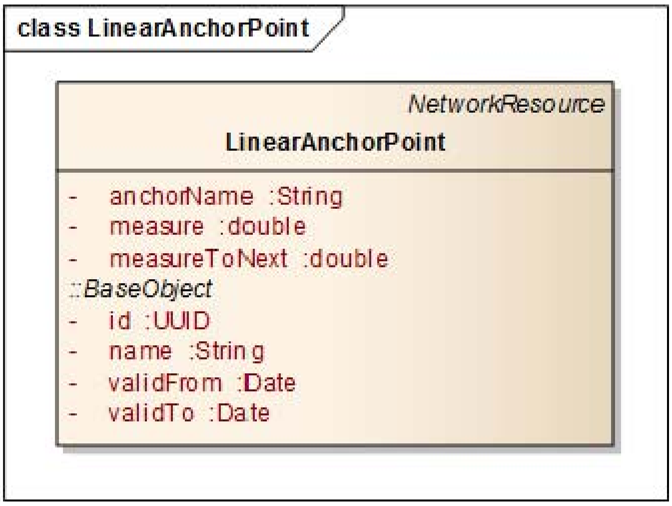


R.30100.422 *Fig. 49 - LinearPositioningSystem (Neighbourhood)*

2.3.4 - LinearAnchorPoint

R.30100.424 The class "LinearAnchorPoint" defines an ordered set of named points within a "LinearPositioningSystem", which are used to transform between LRS based locations suitable for field work and locations using intrinsic coordinates. Each point contains an LRS measure and the distance to next “LinearAnchorPoint” instance.

R.30100.425 This information allows the mapping of LRS locations to intrinsic locations.



R.30100.427 *Fig. 50 - LinearAnchorPoint*

R.30100.428 The class "LinearAnchorPoint" is derived from "NetworkResource".

R.30100.429 A "LinearAnchorPoint" belongs to exactly one "LinearPositioningSystem".

R.30100.430

Table 9 : LinearAnchorPoint (Attributes)

Attributes		
anchorName	String	Name of the “LinearAnchorPoint” instance which is unique within the given “LinearPositioningSystem”
measure	double	Measure of the Anchor Point within the given “LinearPositioningSystem”
measureToNext	double	Basis for modified interpolation of location in the interval up to the next “LinearAnchorPoint” of the given “LinearPositioningSystem”.

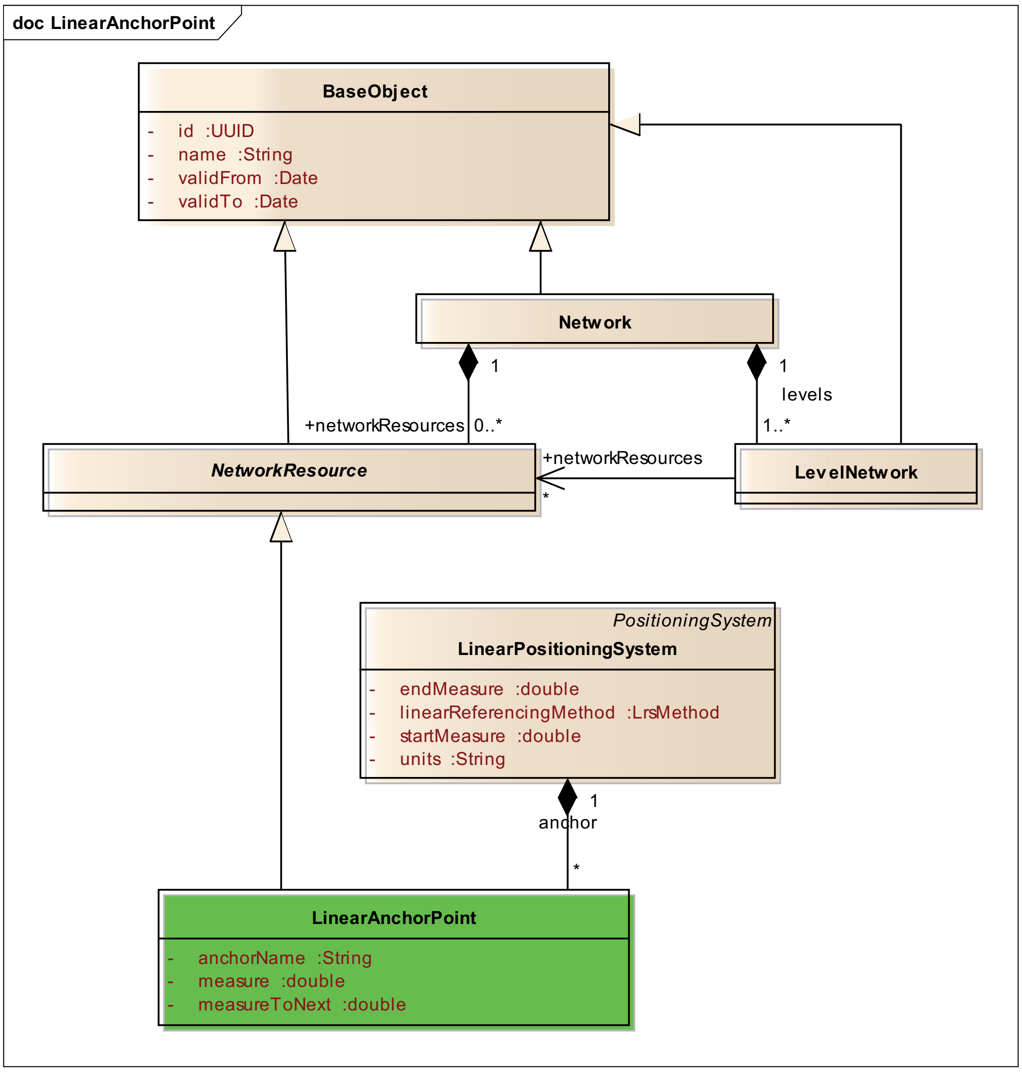
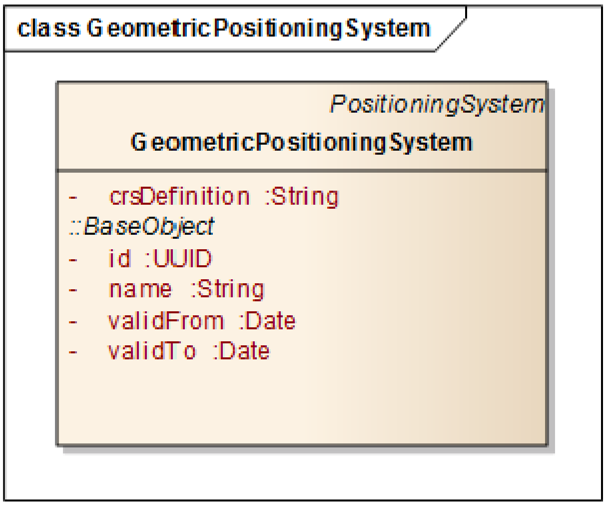


Fig. 51 - LinearAnchorPoint (Neighbourhood)

2.3.5 - GeometricPositioningSystem

R.30100.447 The class "GeometricPositioningSystem" defines schematic, geographic or geodetic Coordinate Reference Systems which are used to position "NetElement" instances or "NetEntity" instances. In the context of RailTopoModel, "GeometricPositioningSystem" instances are used to support the transformation between intrinsic locations and geometric coordinates.



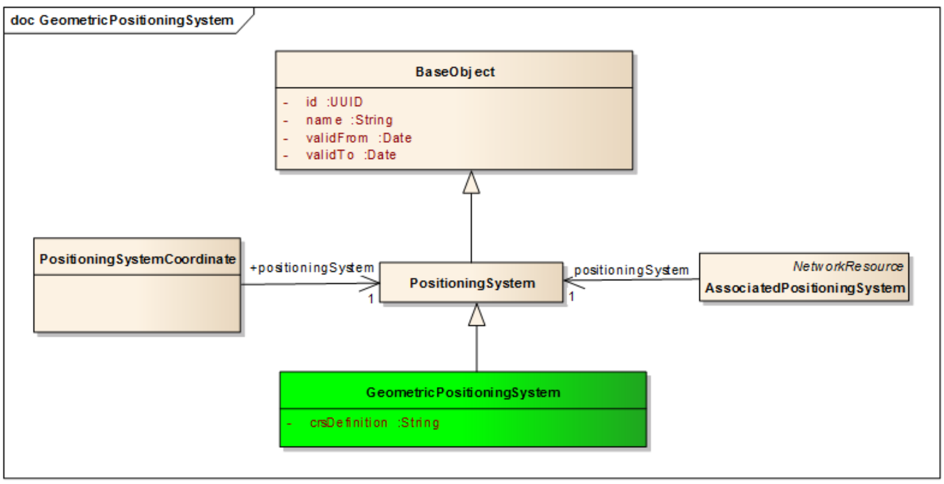
R.30100.449 *Fig. 52 - GeometricPositioningSystem*

R.30100.450 The class "GeometricPositioningSystem" is derived from "PositioningSystem".

R.30100.451

Table 10 : GeometricPositioningSystem (Attributes)

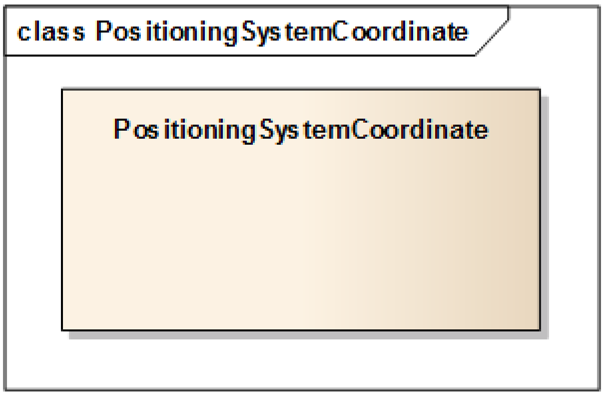
Attributes		
crsDefinition	String	Coordinate Reference System



R.30100.460 *Fig. 53 - GeometricPositioningSystem (Neighbourhood)*

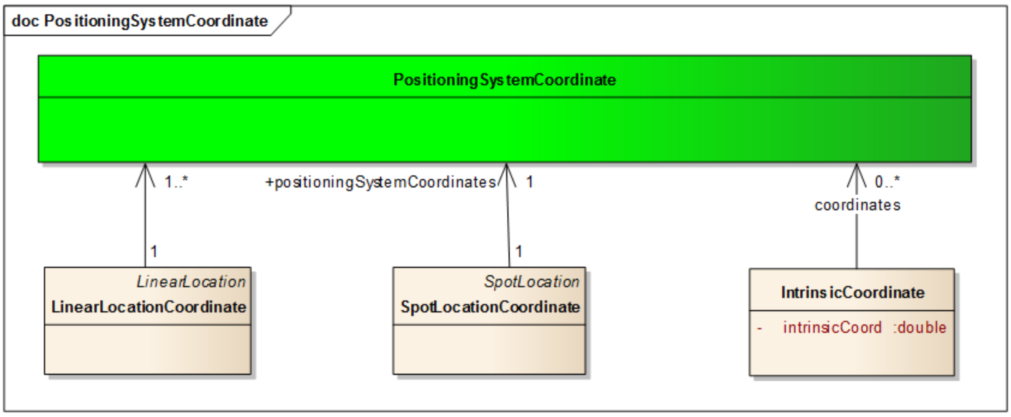
2.3.6 - PositioningSystemCoordinate

R.30100.462 The class "PositioningSystemCoordinate" defines the generic concept of a coordinate in a positioning system that is used to specify locations for "NetEntity", “PositioningNetElement”, and all other objects of the network. These coordinates are either expressed as “GeometricCoordinate”, or “LinearCoordinate”, or any future type of coordinate.



R.30100.464 *Fig. 54 - PositioningSystemCoordinate*

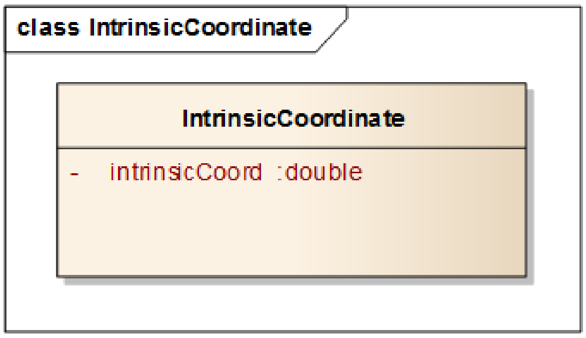
R.30100.465 The class "PositioningSystemCoordinate" has no specific parent class.



R.30100.467 *Fig. 55 - PositioningSystemCoordinate (Neighbourhood)*

2.3.7 - IntrinsicCoordinate

R.30100.469 The class "IntrinsicCoordinate" defines a coordinate which is used to specify locations in reference to "NetElement" instances. An intrinsic coordinate may have an arbitrary real number in interval [0,1] of associated “PositioningSystemCoordinate” instances. 0 and 1 correspond to the extremities of the element.



R.30100.471 *Fig. 56 - IntrinsicCoordinate*

R.30100.472 The class "IntrinsicCoordinate" has no specific parent class.

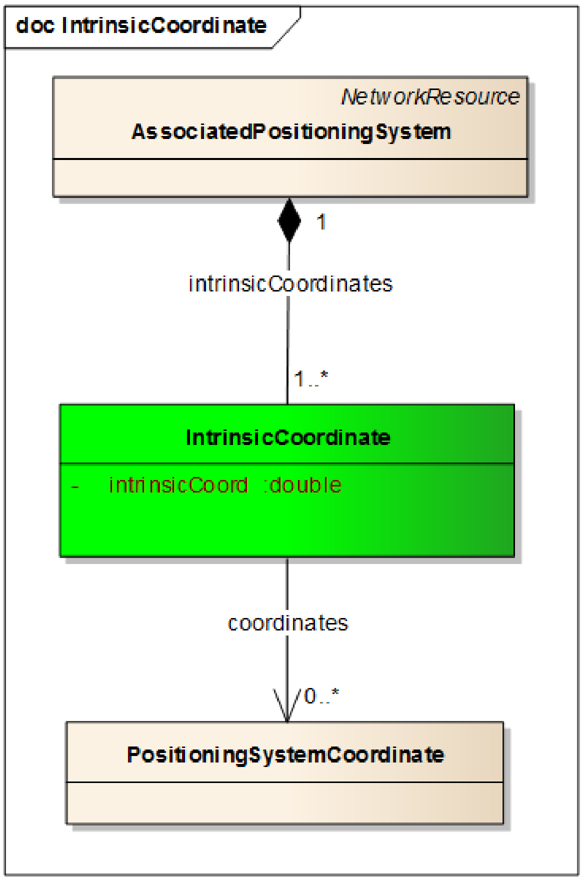
R.30100.473

Table 11 : IntrinsicCoordinate (Attributes)

Attributes		
intrinsicCoord	double	Location in reference to the chosen NetElement, given as value in the interval from 0 to 1.

R.30100.481 An "IntrinsicCoordinate" belongs to exactly one "AssociatedPositioningSystem".

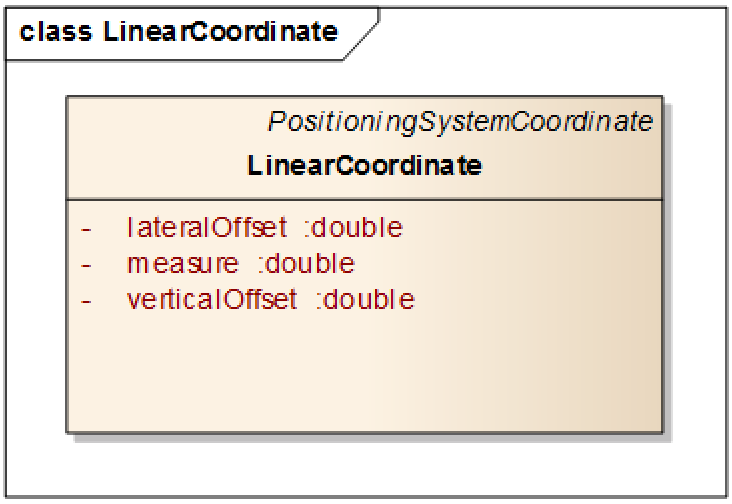
R.30100.482 An “IntrinsicCoordinate” may have multiple associated traditional coordinates. This is required for transformation between intrinsic locations and traditional locations.



R.30100.484 *Fig. 57 - IntrinsicCoordinate (Neighbourhood)*

2.3.8 - LinearCoordinate

R.30100.486 The class "LinearCoordinate" defines a location in reference to a given "LinearPositioningSystem".



R.30100.488 *Fig. 58 - LinearCoordinate*

R.30100.489
R.30100.490 The class "LinearCoordinate" is derived from "PositioningSystemCoordinate".

Table 12 : LinearCoordinate (attributes)

Attributes		
lateralOffset	double	distance perpendicular to the “line of reference”
measure	double	position at the “line of reference” (possibly adjusted to local anomalies using “LinearAnchorPosition”)
verticalOffset	double	Height above the “line of reference” at the position defined by “measure”.

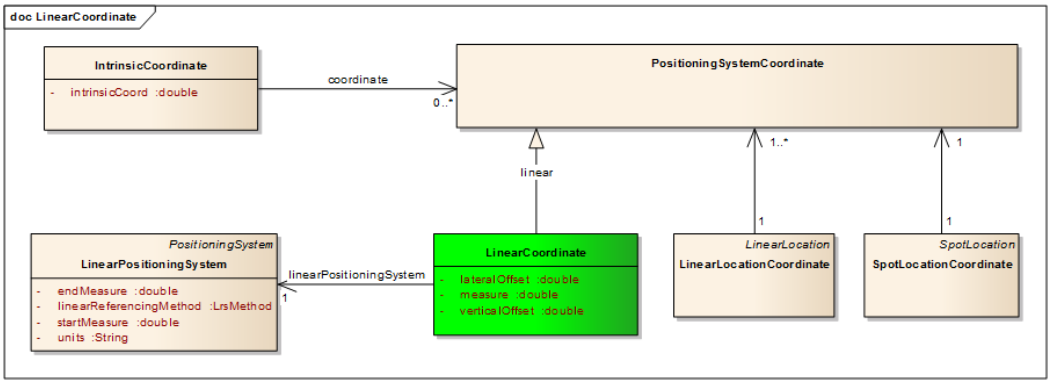
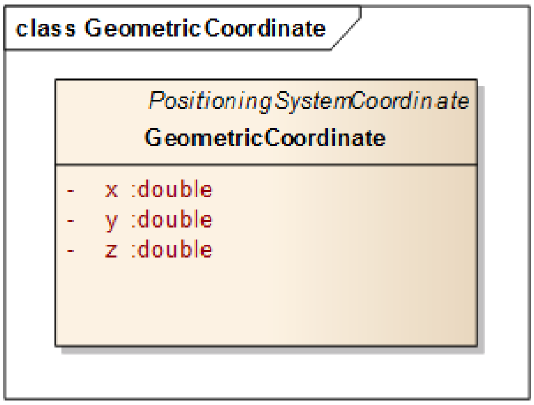


Fig. 59 - LinearCoordinate (Neighbourhood)

2.3.9 - GeometricCoordinate

R.30100.507 The class "GeometricCoordinate" defines one coordinate using a "GeometricPositioningSystem" as reference system. Depending on the properties of the coordinate system used, a coordinate consists of cartesian or spherical values. In case of 2D coordinate systems, the attribute z is undefined.



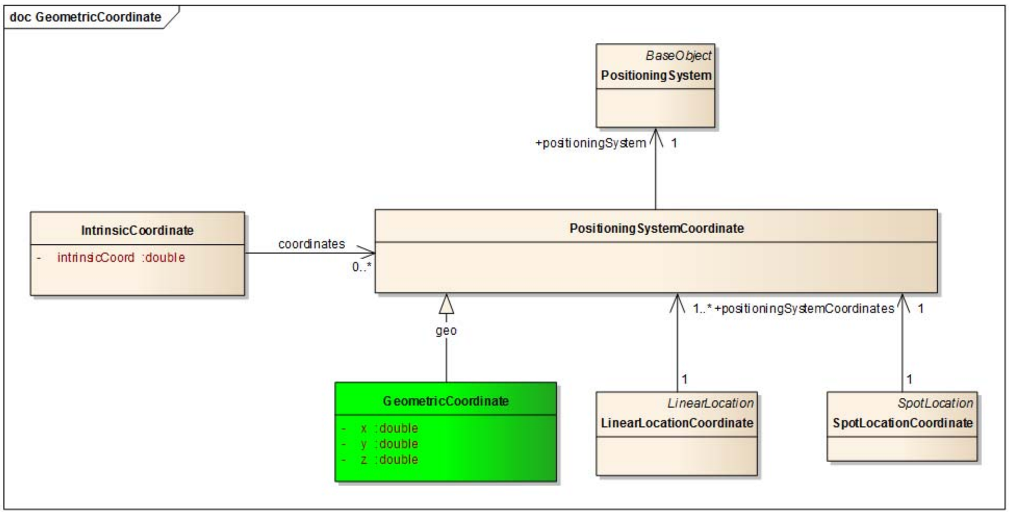
R.30100.509 *Fig. 60 - GeometricCoordinate*

R.30100.510 The class "GeometricCoordinate" is derived from "PositioningSystemCoordinate".

R.30100.511

Table 13 : GeometricCoordinate (Attributes)

Attributes		
x	double	x value of cartesian coordinate, longitude of spherical coordinate
y	double	y value of cartesian coordinate, latitude of spherical coordinate
z	double	z value of cartesian coordinate, altitude of spherical coordinate

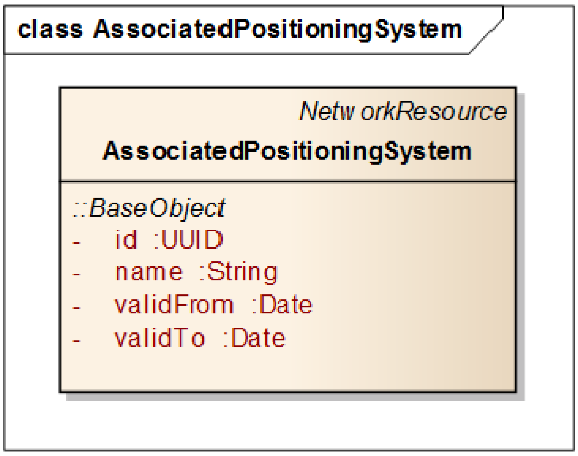


R.30100.526 *Fig. 61 - GeometricCoordinate (Neighbourhood)*

2.3.10 - AssociatedPositioningSystem

R.30100.528 The class "AssociatedPositioningSystem" defines the relation between a "PositioningNetElement" instance and a "PositioningSystem" instance.

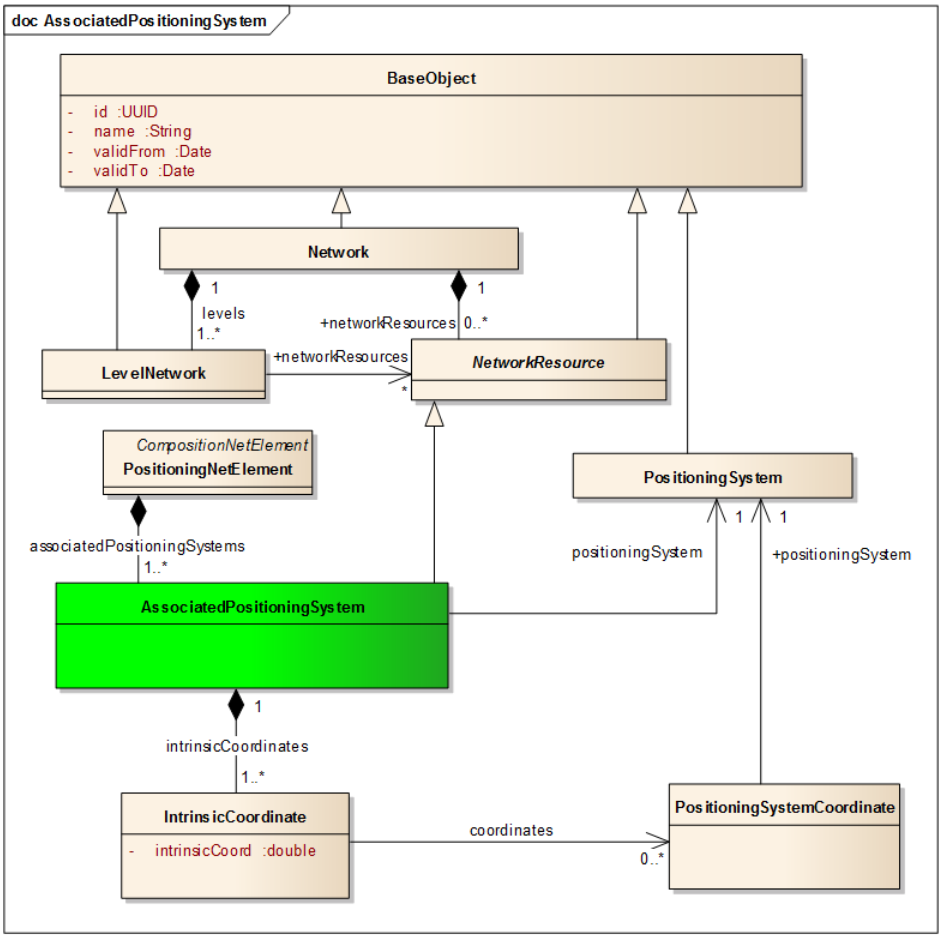
R.30100.529 The associated set of “IntrinsicCoordinate” together with the related “PositioningSystemCoordinate” instances define the translation parameters between “IntrinsicCoordinate” based locations, and locations based on external coordinates (“LinearLocationCoordinate” or “SpotLocationCoordinate”) using “LinearPositioningSystem” or “GeometricPositioningSystem” as a coordinate system.



R.30100.531 *Fig. 62 - AssociatedPositioningSystem*

R.30100.532 The class "AssociatedPositioningSystem" is derived from "NetworkResource".

R.30100.533 Any "AssociatedPositioningSystem" belongs to exactly one "PositioningNetElement".



R.30100.535

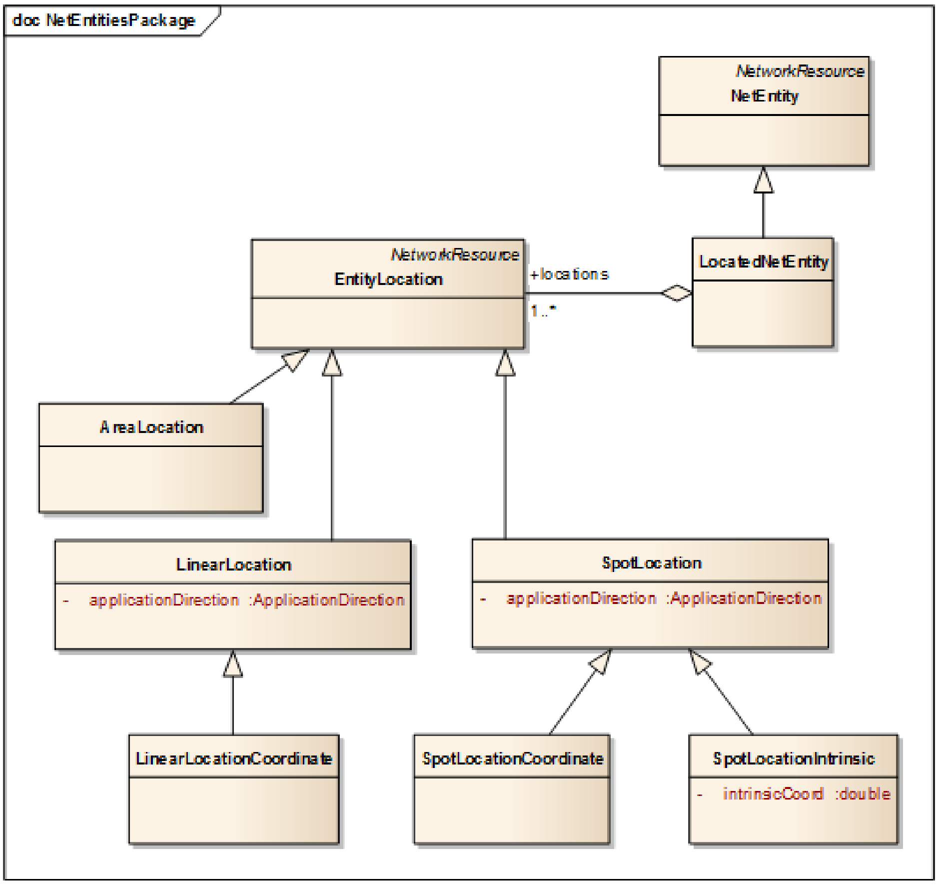
Fig. 63 - AssociatedPositioningSystem (Neighbourhood)

2.4 - Package: Net Entities

2.4.1 - General

R.30100.538 The “Net Entities” package defines the classes that allow to structure the functional description of the considered network, beyond the mere scope of topology. Net entities are the functional images of physical objects (such as bridges or tunnels, signals or level crossings, tracks and switches), or even immaterial objects (such as speed limits or radio coverage area).

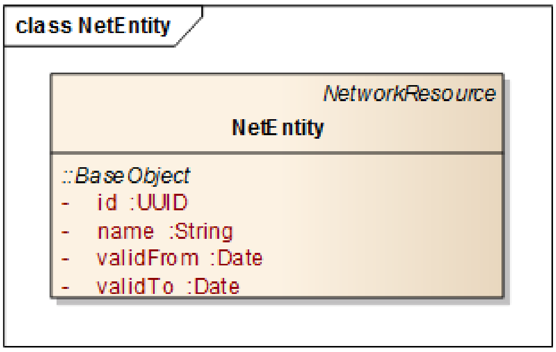
R.30100.539 The “Net Entities” package is structured in such way that these objects are associated with their location, and locations are themselves associated with the topology elements described above.



R.30100.541 *Fig. 64 - Net Entities package overview*

2.4.2 - NetEntity

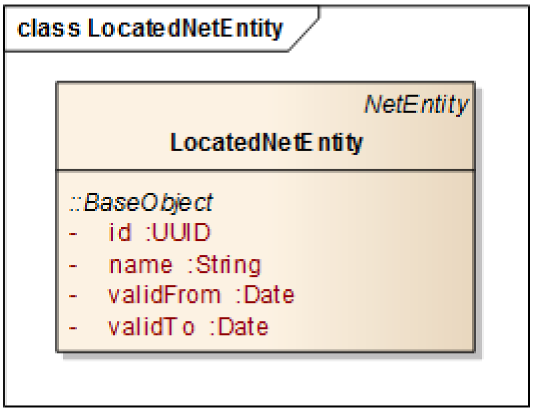
R.30100.543 "NetEntity" is a generic parent class for all information that can be associated with the network considered. Information may be, for instance: tunnels, signals, level crossings, track circuits, speed limits, etc.



R.30100.545 *Fig. 65 - NetEntity*

2.4.3 - LocatedNetEntity

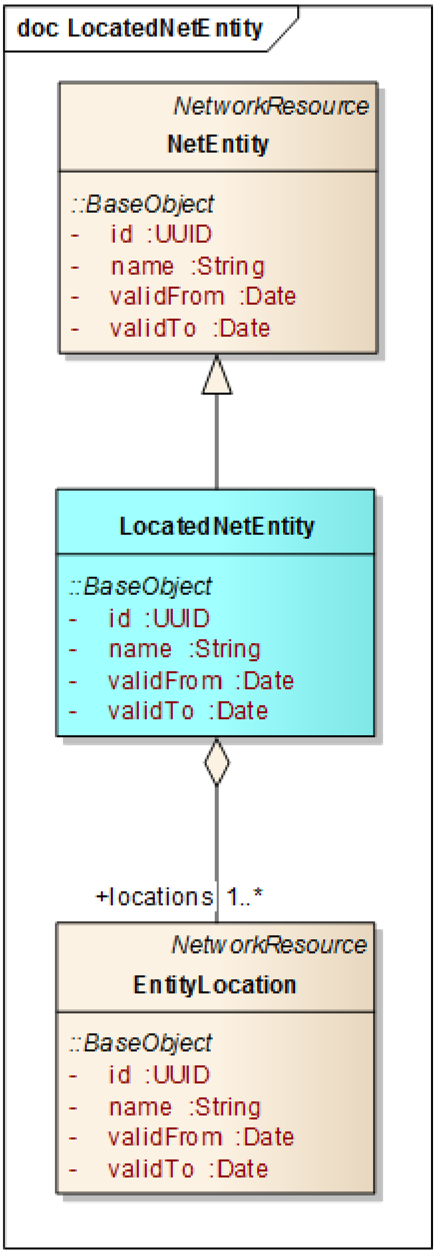
R.30100.547 The class “LocatedNetEntity” is a parent class for information that can definitely be localized, which is the case of most infrastructure-related objects.



R.30100.549 *Fig. 66 - LocatedNetEntity*

R.30100.550 Note: this class has been introduced for semantic clarification, as one may expect “UnlocatedNetEntitites” to also be introduced in the future. Possible derived classes, as shown in the Model overview (see point 1.7.3 - page 17), would be “StructureNetEntity”, “SignallingNetEntity”, “DressingNetEntity”, etc. The RailTopoModel user may create such classes, according to use cases. Further class definitions, resulting from common use cases, may be added to the present Standard in the future.

R.30100.551 "LocatedNetEntity" is a generic docking station for all relevant domain information which can be located in the context of the network in question:

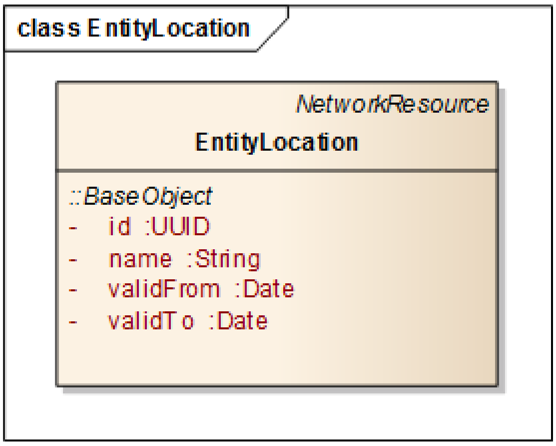


R.30100.553

Fig. 67 - LocatedNetEntity (Neighbourhood)

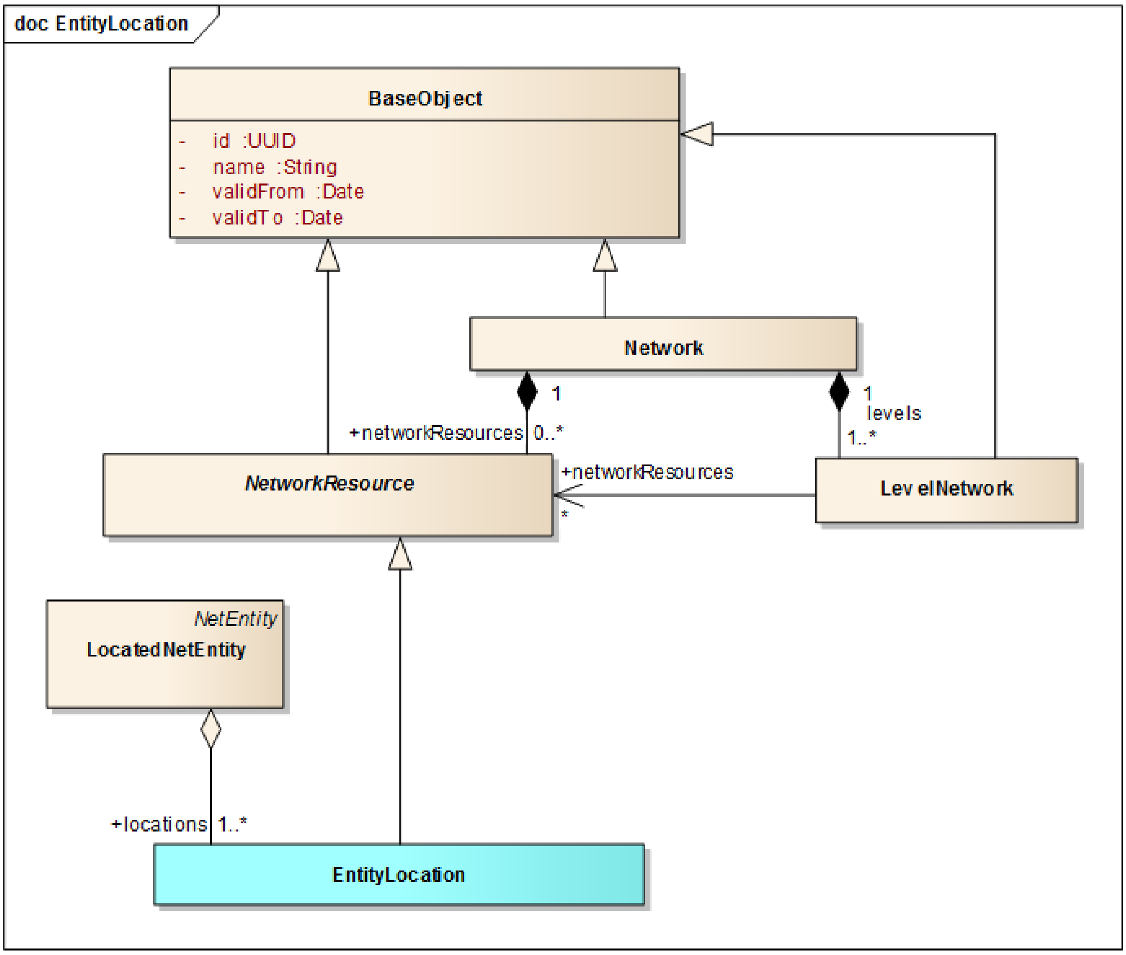
2.4.4 - EntityLocation

R.30100.555 The class "EntityLocation" defines topological and positional location information for "NetEntity" instances.



R.30100.557 *Fig. 68 - EntityLocation*

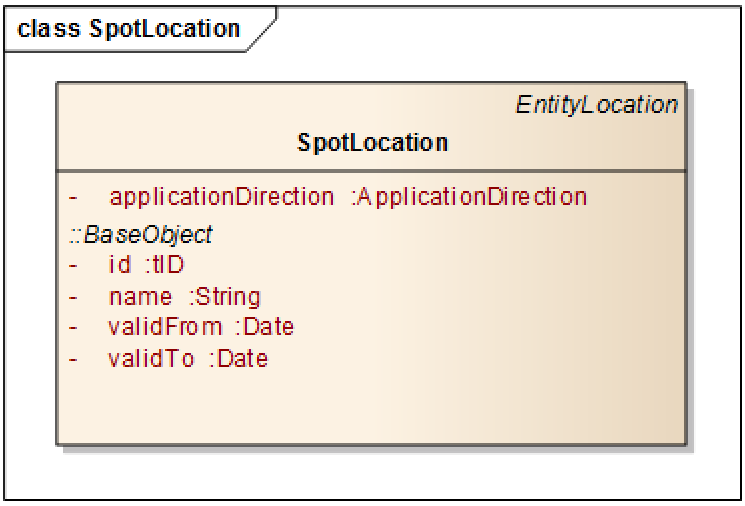
R.30100.558 The class "EntityLocation" is derived from "NetworkResource". One instance of "EntityLocation" belongs to exactly one "LocatedNetEntity" instance. When one of the "LocatedNetEntity" instances is removed, the instance of "EntityLocation" could continue to exist.



R.30100.560 *Fig. 69 - EntityLocation (Neighbourhood)*

2.4.5 - SpotLocation

R.30100.562 The class "SpotLocation" defines point location information for "LocatedNetEntity" instances in reference to one "PositioningNetElement" instance.



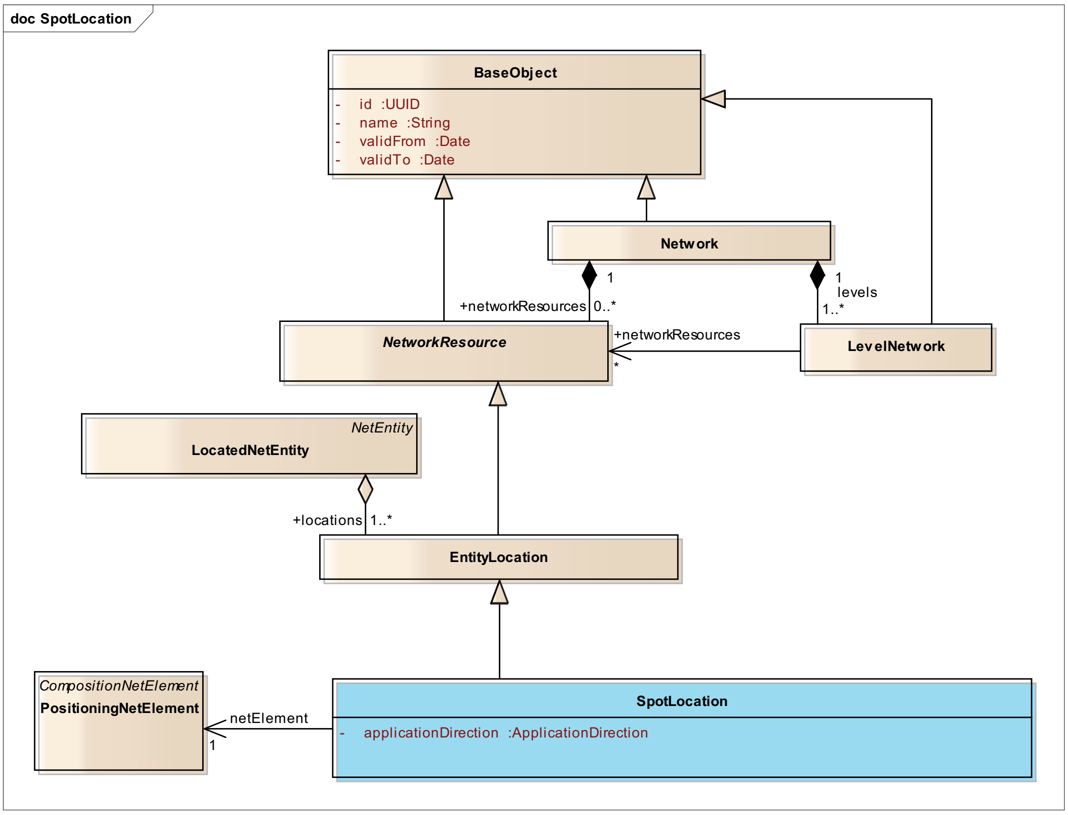
R.30100.564 *Fig. 70 - SpotLocation*

R.30100.565 The class "SpotLocation" is derived from "EntityLocation".

R.30100.566

Table 14 : SpotLocation (Attributes)

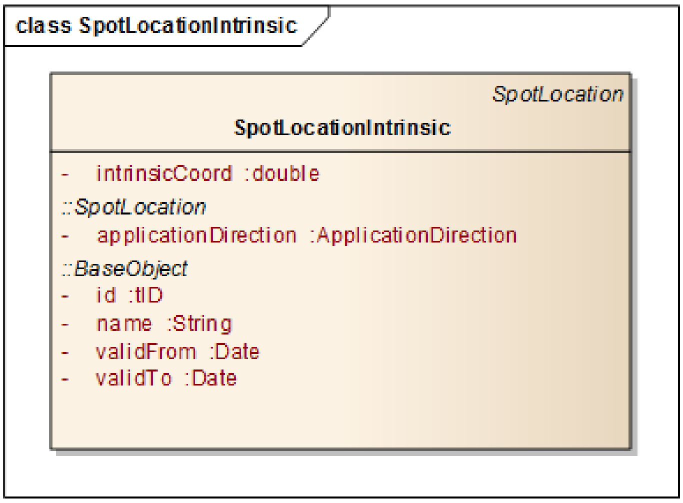
Attributes			
applicationDirection	ApplicationDirection	normal	the located object is valid in the direction of the LinearElement
		reverse	the located object is valid in the reverse direction of the LinearElement
		both	the located object is valid in both directions



R.30100.585 *Fig. 71 - SpotLocation (Neighbourhood)*

2.4.6 - SpotLocationIntrinsic

R.30100.587 The class "SpotLocationIntrinsic" defines additional Information in respect of intrinsic positioning for a "SpotLocation" instance.



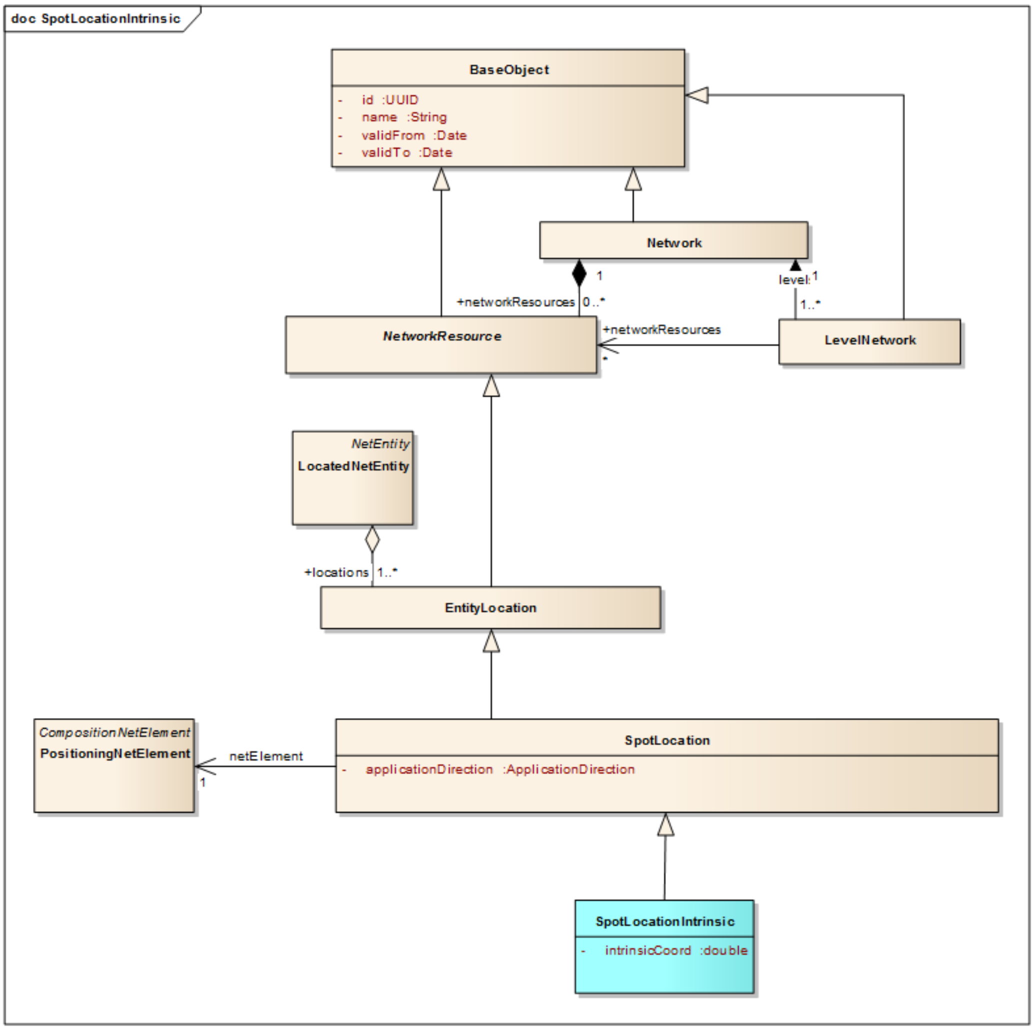
R.30100.589 *Fig. 72 - SpotLocationIntrinsic*

R.30100.590 The class "SpotLocationIntrinsic" is derived from "SpotLocation".

R.30100.591

Table 15 : SpotLocationIntrinsic (Attributes)

Attributes		
intrinsicCoord	double	Location in reference to the chosen NetElement given as value in the interval from 0 to 1.

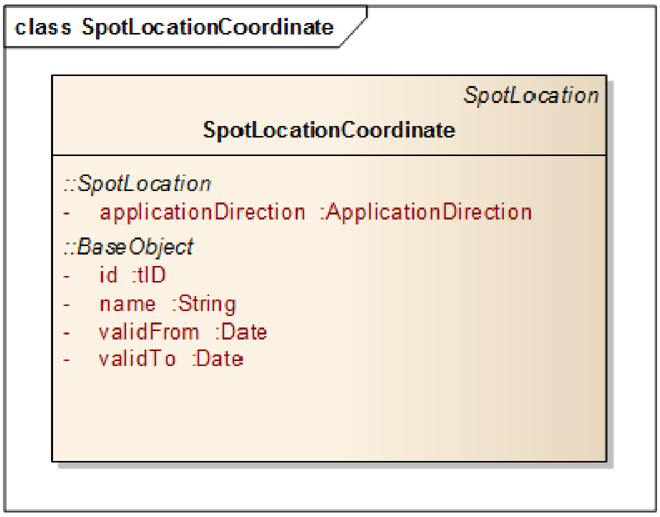


R.30100.600

Fig. 73 - SpotLocationIntrinsic (Neighbourhood)

2.4.7 - SpotLocationCoordinate

R.30100.602 The class "SpotLocationCoordinate" defines the relation between a "SpotLocation" and "PositioningSystemCoordinate".



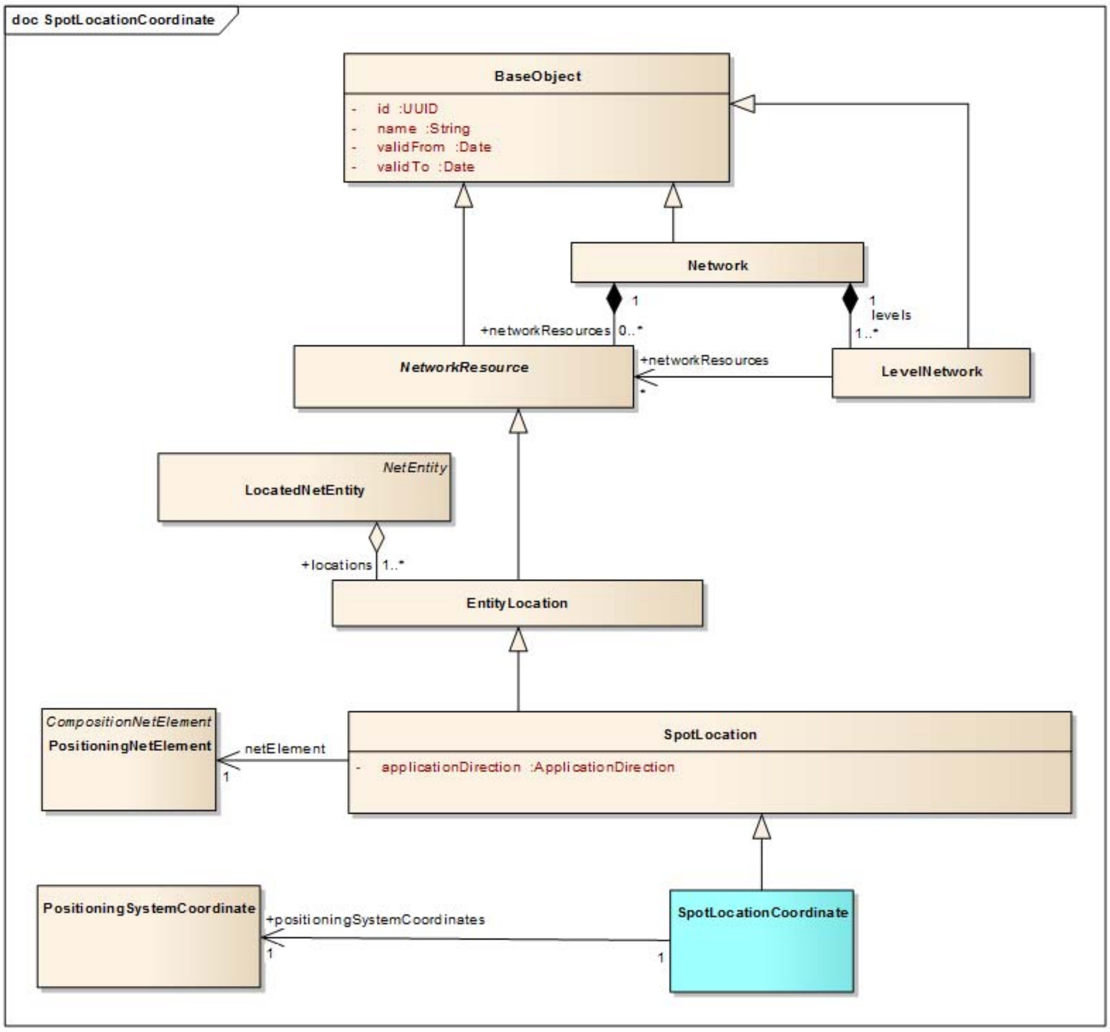
R.30100.604 *Fig. 74 - SpotLocationCoordinate*

R.30100.605

Table 16 : SpotLocationCoordinate (Attributes)

Attributes			
applicationDirection	ApplicationDirection	normal	the located object is valid in the direction of the LinearLocation
		reverse	the located object is valid in the reverse direction of LinearLocation
		both	the located object is valid in both directions

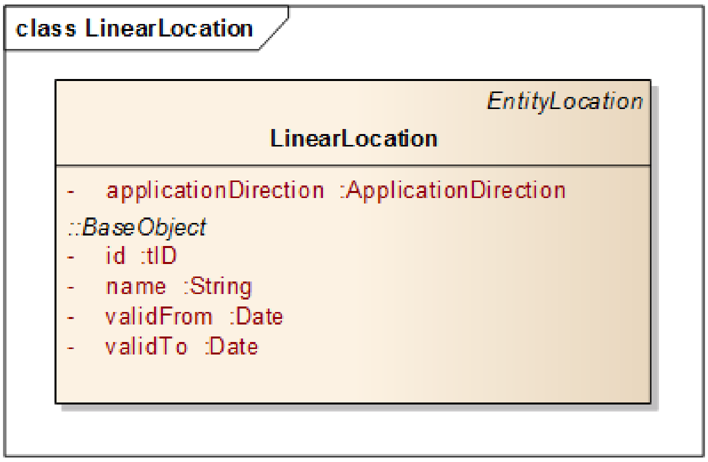
R.30100.623 The class "SpotLocationCoordinate" is derived from "SpotLocation".



R.30100.625 *Fig. 75 - SpotLocationCoordinate (Neighbourhood)*

2.4.8 - LinearLocation

R.30100.627 The class "LinearLocation" defines location information with a startpoint and an endpoint for "LocatedNetEntity" instances in reference to one or more "PositioningNetElement" instances. The set of associated "PositioningNetElement" instances is ordered.

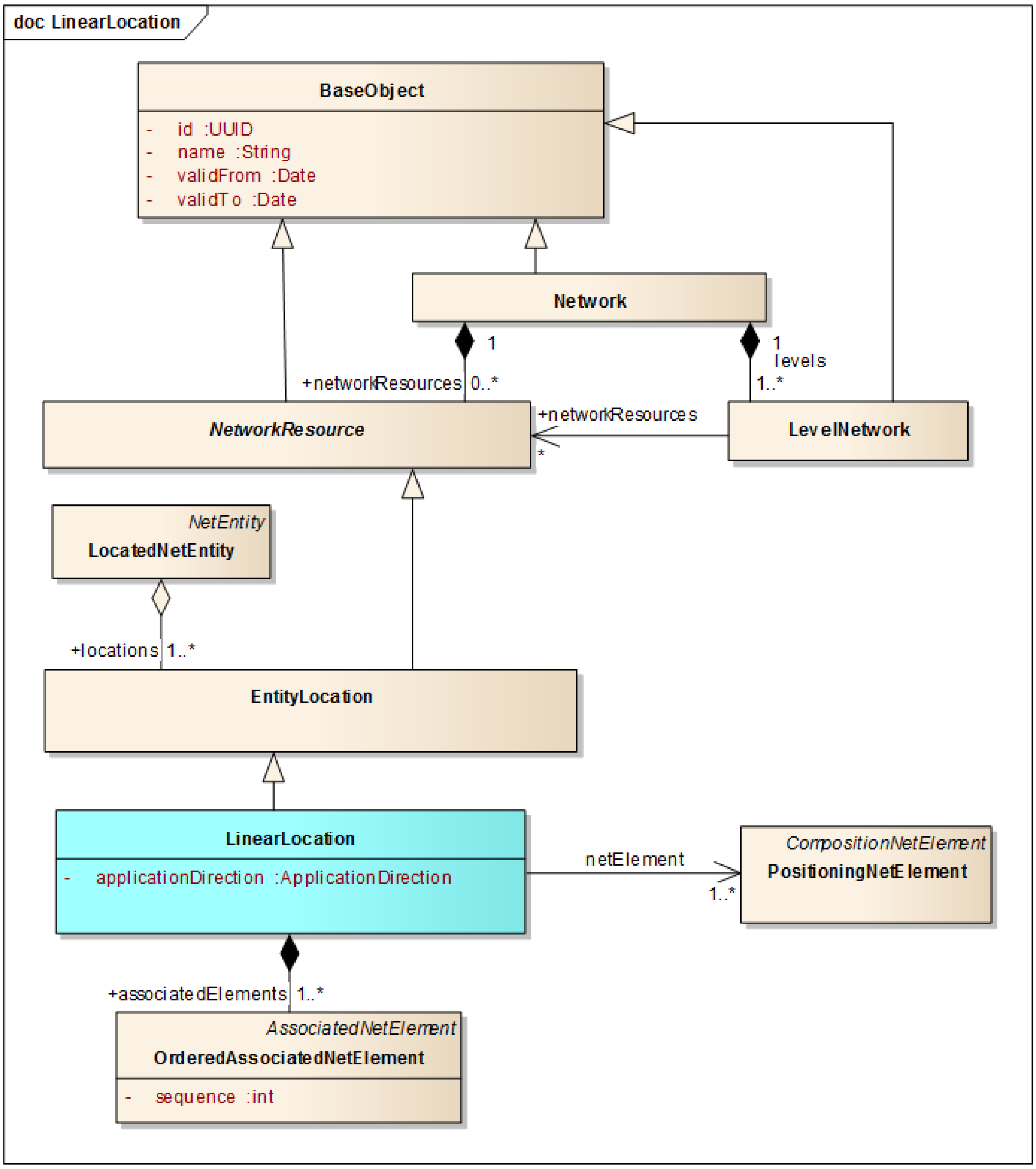


R.30100.629 *Fig. 76 - LinearLocation*

R.30100.630 The class "LinearLocation" is derived from "EntityLocation".

Table 17 : LinearLocation (Attributes)

Attributes			
applicationDirection	ApplicationDirection	normal	the located object is valid in the direction of the LinearLocation
		reverse	the located object is valid in the reverse direction of LinearLocation
		both	the located object is valid in both directions

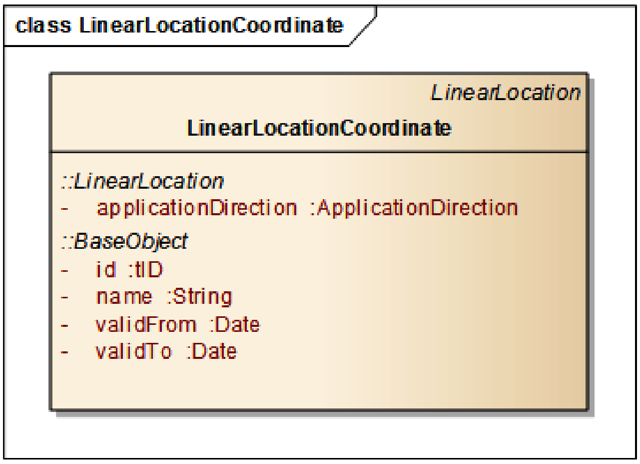


R.30100.650

Fig. 77 - LinearLocation (Neighbourhood)

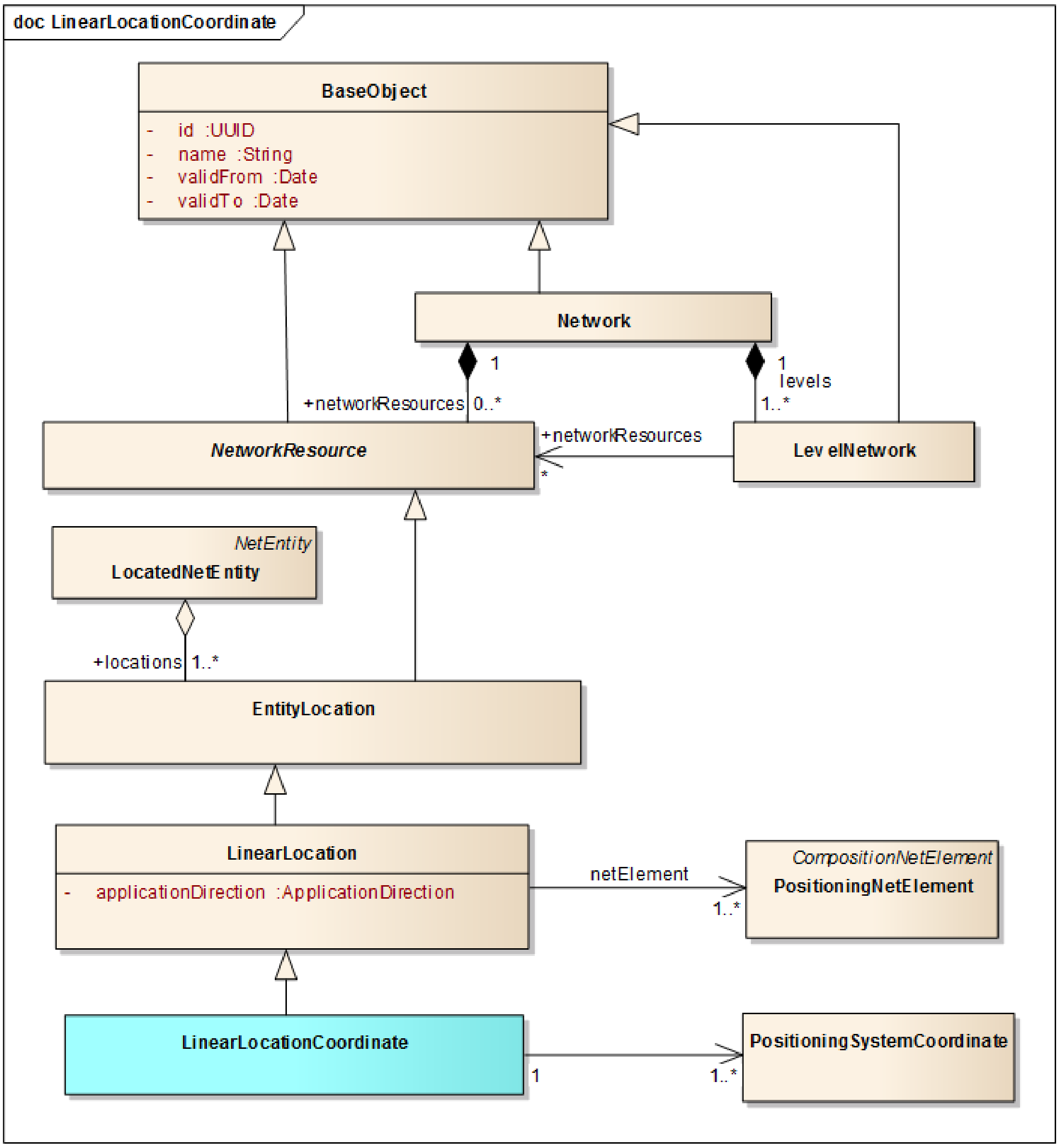
2.4.9 - LinearLocationCoordinate

R.30100.652 The class "LinearLocationCoordinate" defines the relation between a "LinearLocation" and "PositioningSystemCoordinate" instances.



R.30100.654 *Fig. 78 - LinearLocationCoordinate*

R.30100.655 The class "LinearLocationCoordinate" is derived from "LinearLocation".

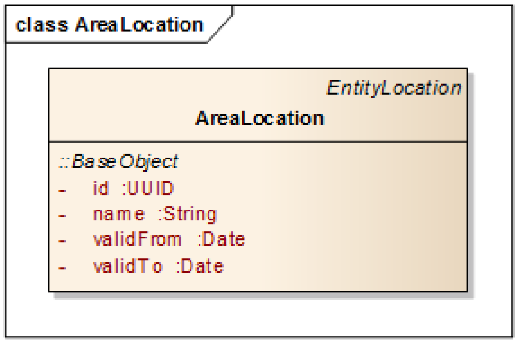


R.30100.657

Fig. 79 - LinearLocationCoordinate (Neighbourhood)

2.4.10 - AreaLocation

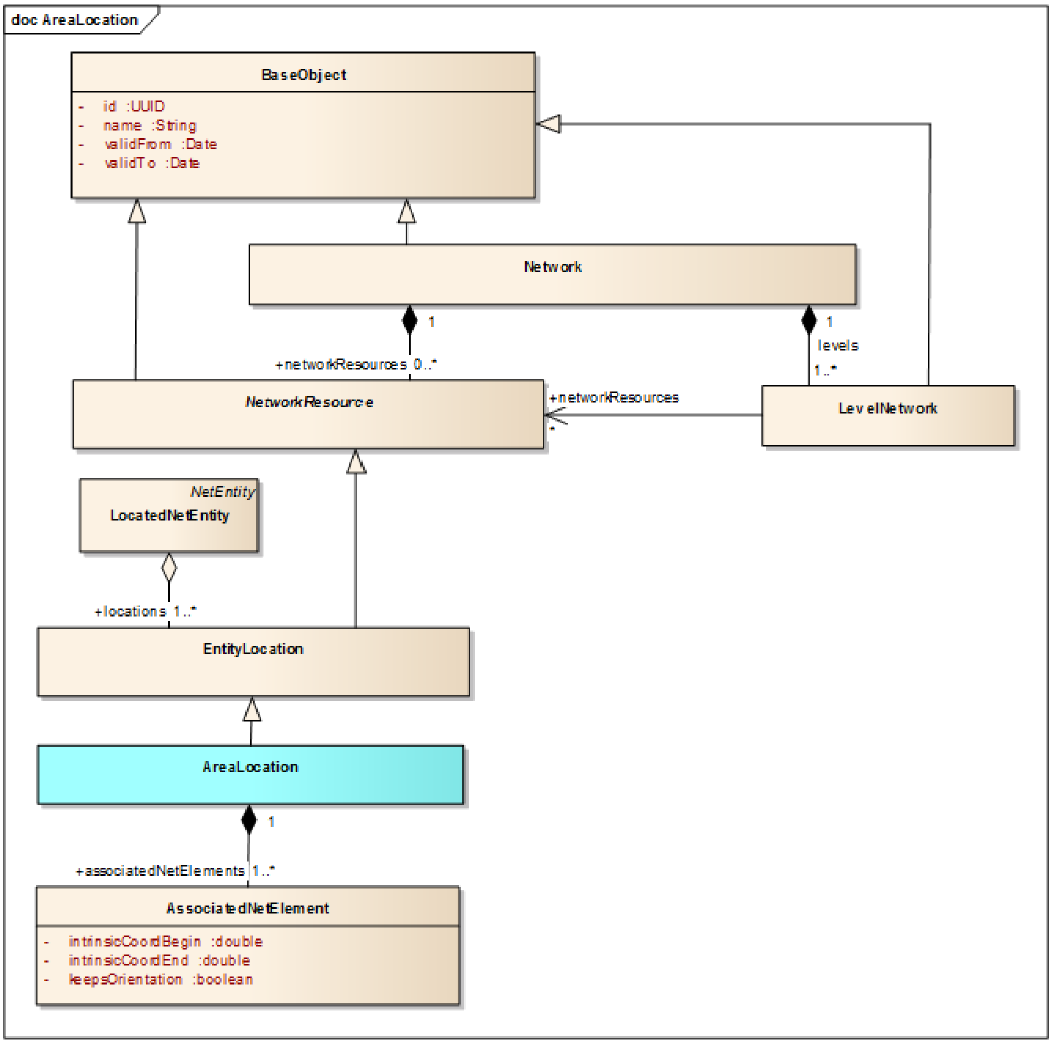
R.30100.659 The class "AreaLocation" defines a set of "AssociatedNetElement" instances which together represent an area of interest. Each “AssociatedNetElement” instance contains attributes which designate the extent of the related “PositioningNetElement” instance using intrinsic coordinates.



R.30100.661 *Fig. 80 - AreaLocation*

R.30100.662 The class "AreaLocation" is derived from "EntityLocation".

R.30100.663 Whenever an instance of "AreaLocation" is removed, all related instances of "AssociatedNetElement" are removed as well.



R.30100.665

Fig. 81 - AreaLocation (Neighbourhood)

3 - Conformance

3.1 - General

R.30100.668 RailTopoModel, as a conceptual data model, describes essential structural properties of possible implementations of railway IT systems (named “Systems” in the present section), when these Systems contain or handle information about railway infrastructure.

R.30100.669 Conformance is defined in the present section so as to ensure, inter alia:

R.30100.670 - the computational efficiency and scalability of the System;

R.30100.671 - the compatibility of the System with railML®, when relevant.

R.30100.672 To be declared conformant to the RailTopoModel, Systems must present the features required below (“shall”). Recommended (“should”) and optional (“may”) features are also mentioned, for the sake of clarity.

R.30100.673 Conformant Systems:

R.30100.674 - may include all RailTopoModel concepts, or a subset of these concepts;

R.30100.675 - may extend the RailTopoModel, e.g. with additional packages and classes;

R.30100.676 - shall not alter the concepts provided under the present IRS and their relations, irrespective of whether these concepts are required, recommended, or optional, except for the cases described below.

3.2 - Referencing techniques

R.30100.678 Linear net elements shall be used as a reference system.

R.30100.679 Locations shall be stored using intrinsic positioning.

R.30100.680 Transformation between intrinsic positioning and linear references (if used) shall be supported.

R.30100.681 Transformation between intrinsic positioning and geometric positioning (if used) shall be supported.

3.3 - Description Levels

R.30100.683 At least one of the widely used description levels (Macro, Meso, Micro) level description should be supported.

R.30100.684 Additional description levels may be implemented.

3.4 - Navigation

R.30100.686 The system shall provide topological navigation inside and between different description levels.

R.30100.687 Interaction between objects of different description levels shall be supported.

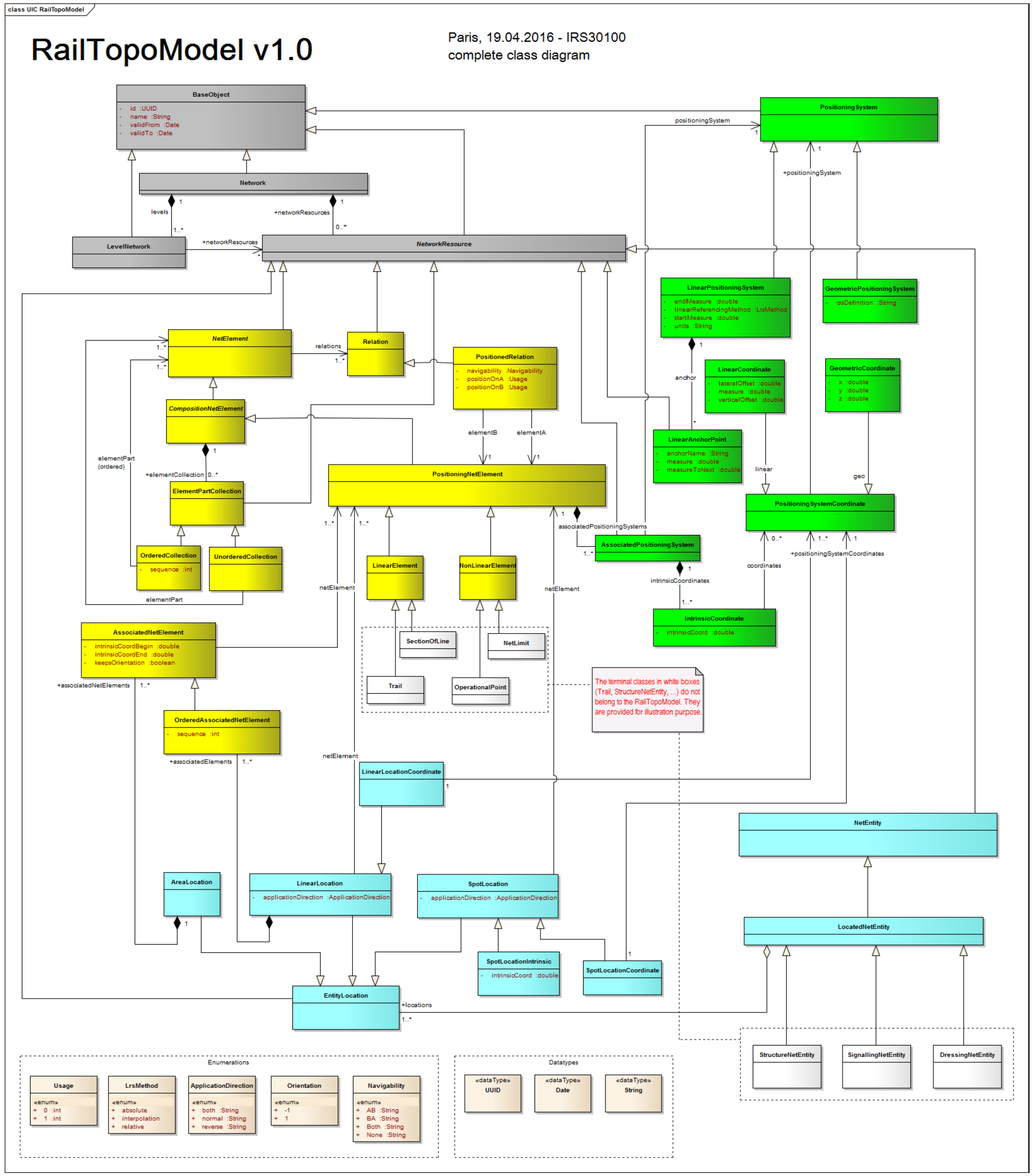
R.30100.688 A conformant System may only have one level. In such case, the aggregation mechanisms need not be developed.

3.5 - Identification of objects

R.30100.690 If a new System is created, all objects derived from “BaseObject” will, by design, inherit a unique identifier in the shape of a UUID (Leach, 2005).

R.30100.691 For legacy Systems being brought, partially or progressively, in conformance with the RailTopoModel, other identifiers may be used instead.

Appendix A - RailTopoModel complete class diagram



Warning

Printed by the International Union of Railways (UIC)
16, rue Jean Rey 75015 Paris - France, September 2016
Dépôt Légal September 2016

ISBN 978-2-7461-2513-1