TCCS SD1 - Data Model - Infrastructure

1 Table of Contents

2 Package "Infrastructure"

2.1 Header

SPT2TS-122296 - {

"$schema": "ERJU meta-model.json",

"isDefinedBy": "http://ERJU/datamodel/0.4/infrastructure",

"name": "infrastructure",

"containerStruct": "Infrastructure",

"prefix": "infra",

"intId": 1,

"version": "1.0",

"info": "Railway network topology and topography, position wrt topology, location on a map, geometry, aka. alignment",

"structs": [], "enums": []

} [⇗ Open]

2.2 Track Topology

SPT2TS-122297 - The railway system's determining feature is the network of connected tracks. The railway network topology is defined by the tracks and the connections that allow trains to travel from one track to the next.   [Content to be approved]

2.2.1 Design decisions

SPT2TS-49043 - The topology shall be clustered into so-called 'TopoArea(s)' where in subsets of track topologies are defined. This clustering allows high flexibility while handling infrastructure data for the consuming systems. For instance, a Traffic Management System responsible for three different interlockings can require several topo areas to handle operations in comparison to the topo area requirement for the interlockings. Hence, it also offers a great deal of data separation in terms of identifying relevant areas for consuming systems. [Content to be approved]

SPT2TS-49041 - The track edges shall be defined with implicit orientation direction from start to end based on their length. This implicit definition improves the understanding of track edge definition in the model and paves as a smoother way for the specification of track edge links. [Content to be approved]

SPT2TS-49040 - The track nodes are omitted from the SD1 data model since the information provided by this model

element (e.g., start and end of track edge, track navigability) can be sufficiently covered by other existing model elements (e.g., Track Edge and Track Edge Links) in the SD1 Model. Hence, it was ruled out on the grounds of redundancy and optimisation of the data volume of the model. For more information in Track Nodes, see ⊞ SPT2TS-7 - RCA Digital Map – MAP Object Catalogue [RCA.Doc.69]   [◉⬧Content to be approved]

2.2.2 Topology

SPT2TS-49039 - Topo Area: A topo area is a container for rail network topology to manage (e.g. publish) parts of the infrastructure. This lets user agree on subsets of topology that is of concern to their particular use case. Its purpose is to contain subset(s) of the network topology. Topo areas consist of a list of track edges along with track edge links. [◉⬧Content to be approved]

SPT2TS-49010 - TrackEdge: Track edge is a linear object that defines an uninterrupted stretch of a railway track without divergence or convergence. A track edge is defined along the centre line of the 2D or 3D track alignment and has a finite length. A track edge is associated with track edge links to determine navigabilities between track edges. The track edges have an implicit direction. This direction is defined with respect to their start and end locations. The default orientation direction is start to end. The start of track edge is identified as the point where length is 0 and the end of track edge is identified as the point where length is maximum. The maximum length is given by the parameter 'length' in track edge class.

The start/end of track edge shall correspond to the location of a simple point, or any form of end of track. The start/end of track edge corresponds to the tangent/secant point or the geometric start location of a simple point.
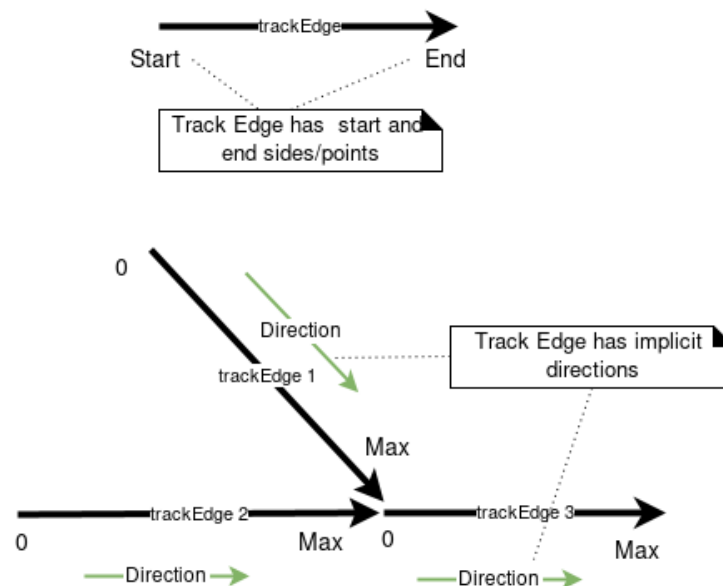

Figure 1 Topological representation of Track Edge

The following equivalent classes from other models can be referred to:

- LinearNetElement, or LinearElementWithLength at MICRO Level (the term "NetElement" is avoided to support comprehensibility and to meet ▱ SPT2TS-154) in ⊞ SPT2TS-10 - RSM - RailSystemModel
- ⊞ SPT2TS-9 - LinX4Rail CDM ->PSM (Platform Specific Data Model)
- LinearNetElement in ⊞ SPT2TS-8 - EULYNX Data Model (dataprep.eulynx.eu)
- Track Edge in ⊞ SPT2TS-7 - RCA Digital Map – MAP Object Catalogue [RCA.Doc.69]

[◉⬧Content to be approved]

SPT2TS-49008 - TrackEdgeLink: Track edge link defines a relation between two track edges along which a train can run. Track edge links determine which travel options are permitted at points and which are excluded. They are necessary to eliminate the physically impossible train movements (e.g. a movement from the left to the right branch of a point) and define the practically possible movements. A turn table can be modelled as well and would be added to the model as a functional element as soon as requirements for it are provided.
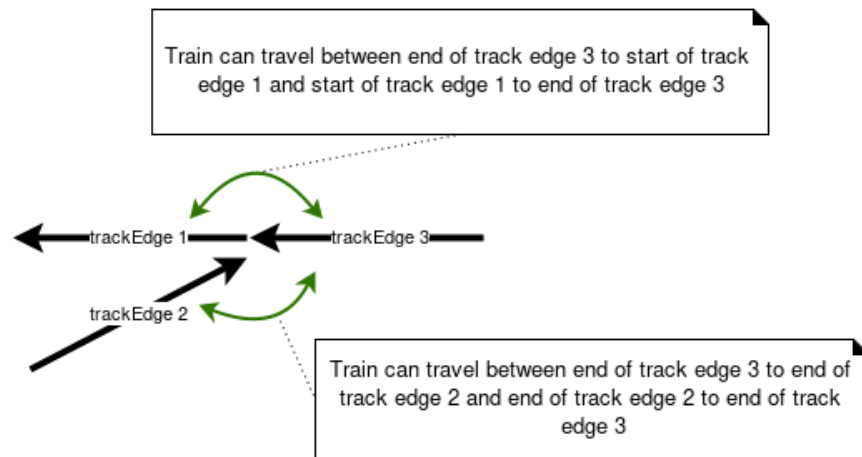

Figure 2 Track Edge Links connecting the Track Edge

The following equivalent classes from other models can be referred to:

- Positioned Relation in SPT2TS-10 - RSM - RailSystemModel
- Track Navigability in SPT2TS-7 - RCA Digital Map – MAP Object Catalogue [RCA.Doc.69]
- Net relation SPT2TS-9 - LinX4Rail CDM ->PSM (Platform Specific Data Model)

[Open]
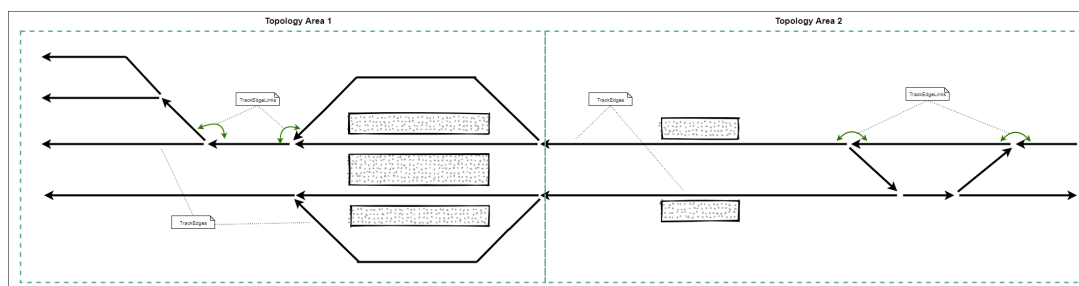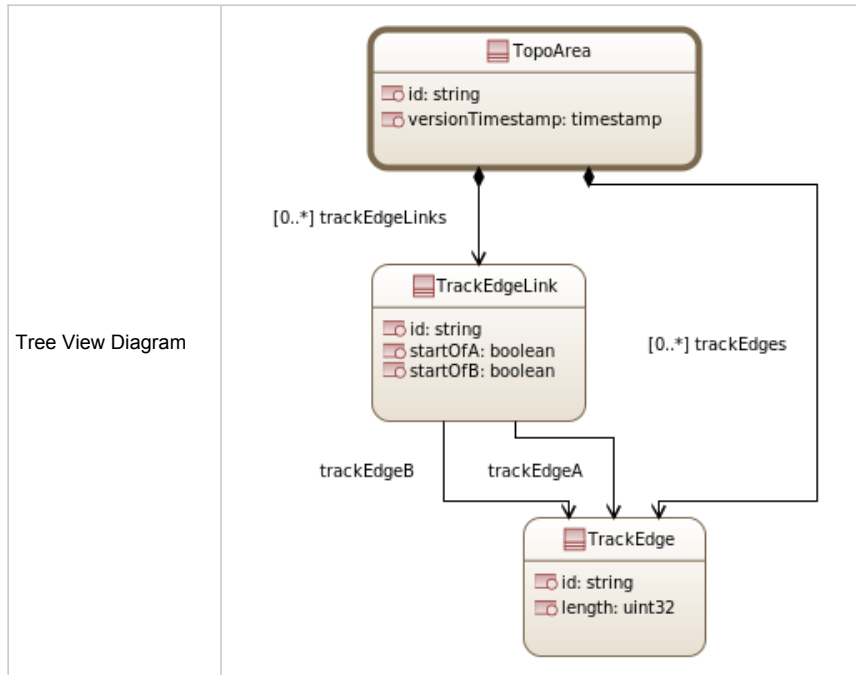
SPT2TS-49006 -

Figure 3 Representation of Topo area

Note: The purpose of the above illustration is only to portray a Topo area. It does not implicate any requirements on how the Topo areas are defined. **[Open]**

TopoArea

Tree View Diagram

SPT2TS-125434 - TopoArea

```
{
  "structs": [
  {
    "name": "TopoArea",
    "info": "Defines a container for rail network topology",
    "belongsToSubPackage": "topology",
    "attrs": [
      {"intId": 1, "name": "id", "dataType": "string", "key": "global", "info": "Identity of the object; used for referencing"},
      {"intId": 2, "name": "versionTimestamp", "dataType": "timestamp", "info": "version information which is valid since timestamp"},
      {"intId": 3, "name": "trackEdges", "composition": "TrackEdge", "multiplicity": "0..*", "sortedByKey": true, "info": "composes of track edges"},
      {"intId": 4, "name": "trackEdgeLinks", "composition": "TrackEdgeLink", "multiplicity": "0..*", "sortedByKey": true, "info": "composes of track edge links"}
    ]
  }]
}
```

SPT2TS-49005 - TrackEdge

```
{
  "structs": [
  {
    "name": "TrackEdge",
    "info": "A track edge is an uninterrupted stretch of railway track, without divergence or convergence.",
    "belongsToSubPackage": "topology",
  "ontology": {"subClassOf": "http://data.europa.eu/949/NetElement"},
    "attrs": [
      {"intId": 1, "name": "id", "dataType": "string", "key": "global", "info": "Identity of the object; used for referencing"},
      {"intId": 2, "name": "trackEdgeLength", "dataType": "uint32", "unit": "m", "exp": -3,
        "info": "Distance along the TrackEdge's track centre line. ",
        "ontology": {"subPropertyOf": "http://data.europa.eu/949/length"}}
    ]
```

```
  }]
}
```

SPT2TS-125424 - TrackEdgeLink

```
{
  "structs": [
  {
    "name": "TrackEdgeLink",
    "info": "Defines a relation between two track edges along which a train can run.",
    "belongsToSubPackage": "topology",
    "ontology": {"subClassOf": "http://data.europa.eu/949/NetRelation"},
    "attrs": [
      { "intId": 1, "name": "id", "dataType": "string", "key": "global", "info": "Identity of the object; used for referencing"},
      { "intId": 2, "name": "trackEdgeA", "reference": "TrackEdge", "info":"Connects to track edge A",
        "ontology": {"subPropertyOf": "http://data.europa.eu/949/elementA"}},
      { "intId": 3, "name": "trackEdgeB", "reference": "TrackEdge", "info":"Connects to track edge B",
        "ontology": {"subPropertyOf": "http://data.europa.eu/949/elementB"}},
      { "intId": 4, "name": "startOfA", "dataType": "boolean", "info": "True when linked to the start of track edge A, false
when to the end"},
      { "intId": 5, "name": "startOfB", "dataType": "boolean", "info": "True when linked to the start of track edge B, false
when to the end"}
    ]
  }]
}
```

SPT2TS-125435 - Direction

```
{
  "enums": [
  {
  "name": "Direction",
  "info": "List of applicable usage directions with respect to Track Edge e. g. for restrictions",
  "enumLiterals": [
    {"intId": 0, "name": "dirBoth", "info": "Applicable for 'both' travel directions"},
    {"intId": 1, "name": "dirSame", "info": "Applicable in the same direction as TrackEdge"},
    {"intId": 2, "name": "dirReverse", "info": "Applicable in the opposite direction of the TrackEdge (from end to start)"}
  ]
  }
  ]
}
```

2.2.2.1 Linking Areas

SPT2TS-123555 - From the model point of view the linking of two areas happens by using TrackEdgeLinks referencing TrackEdge of the neighbour Area.
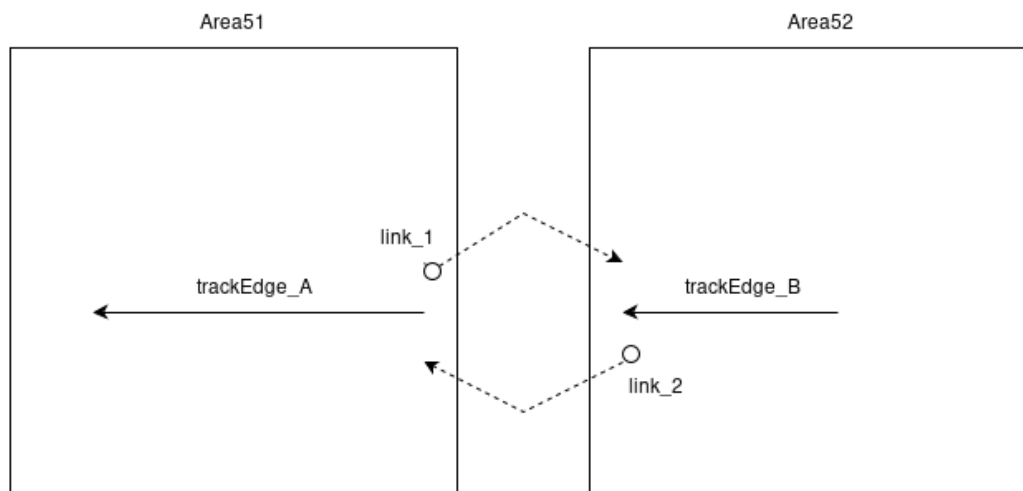
Figure 4 Linking areas

The json-serialisation for this example would be for the "Infrastructure"-object

```
{
  "topoAreas": [
  {
    "id": "Area51",
    "trackEdges": [
       {"id": "trackEdge_A", "length": 10000}
    ],
    "trackEdgeLinks": [
       {"id": "link_1", "trackEdgeA": "trackEdge_A", "trackEdgeB": "/1/1[Area52]/2[trackEdge_B]", "startOfA": true,
"startOfB": false}
    ]
  },
  {
    "id": "Area52",
    "trackEdges": [
       {"id": "trackEdge_B", "length": 10000}
    ],
    "trackEdgeLinks": [
       {"id": "link_2", "trackEdgeA": "trackEdge_B", "trackEdgeB": "/1/1[Area51]/2[trackEdge_A]", "startOfA": false,
"startOfB": true}
    ]
  }
]
}
```

As visible from the example, the linking between TrackEdges is done twice by two TrackEdgeLink-objects. This allows
bi-directional navigation and does not introduce any implications on distance-calculation, as the TrackEdgeLink has zero
length.

Also, the referenced TrackEdge from another TopoArea must be referenced with an absolute reference (see ▯ SPT2T

S-122407 - Referencing objects). [→ Open]

2.2.3 Supporting containers

Supporting classes are used only as containers for a set of attributes. They don't contain sufficient information on their own, they don't have an ID and cannot be referenced.

2.2.3.1 Track Edge Point

SPT2TS-63854 - A Track Edge Point is a simple supporting container for two attributes: trackEdge-reference and position on it. It does not have an ID and is used only in a composition relation. In most cases a functional object needs only one position-definition (e.g. a Balise). In this cases two separate attributes (trackEdge and pos) are used. In seldom cases an object needs more than one position (e.g. Crossing needs two positions). In this case an array of two TrackEdgePoints is used.

The class DirTrackEdgePoint is also a simple container for three attributes including the active direction with the same usage-pattern as TrackEdgePoint.
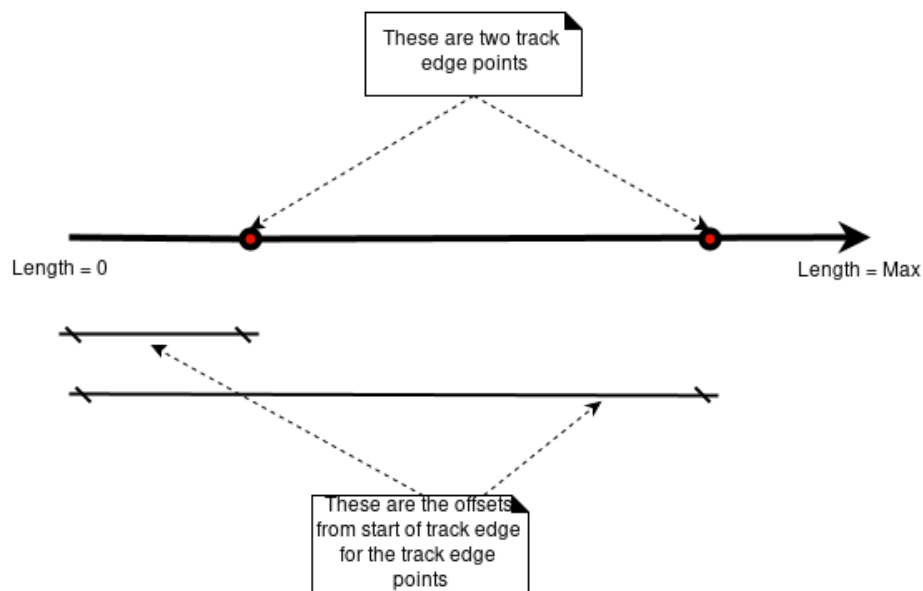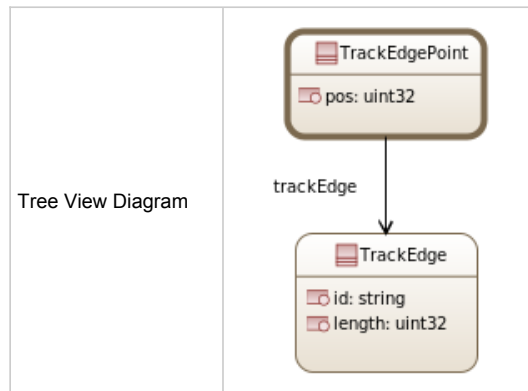

Figure 5 Illustration of Track Edge Point

Although the TrackEdgePoint contains similar information as:

- Spot Location in SPT2TS-10 - RSM - RailSystemModel
- Track Edge Point in SPT2TS-7 - RCA Digital Map – MAP Object Catalogue [RCA.Doc.69]

Its usage is different. In both upper models, the spot locations are defined separately and then used in objects by reference via IDs. Therefore, in both cases, the objects (RSM.SpotLocation and RCA.TrackEdgePoint) use an ID attribute. In this model this level of indirection is avoided, the TrackEdgePoint cannot be referenced and used directly in a composition relation. [→ Open]
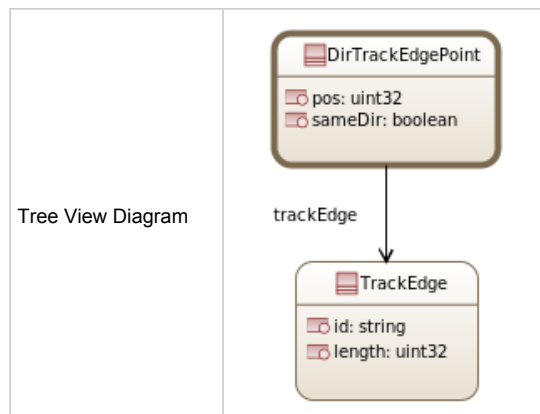
TrackEdgePoint

```
    {"intId": 2, "name": "pos", "dataType": "uint32", "unit": "m", "exp": -3, "info": "position on the associated Track Edge"
},
    {"intId": 3, "name": "sameDir", "dataType": "boolean", "info": "usage direction of the TrackEdge ; true: active in same
direction as trackEdge, false: active in opposite direction"}
  ]
 }
]
}
```

2.2.3.2 Track Edge Section

SPT2TS-63855 - A Track Edge Section is a simple container for three attributes defining part of a TrackEdge. It is used by other objects as supporting data-object (e.g. TrackEdgeGauges). Track Edge Sections have reference points defined to indicate their start and end locations. The reference points are defined using distance skips from start and end of track edges. The starting/ending point is to calculated as a distance increment/decrement from start/end of track edge respectively. Track Edge Section is always limited to one track edge i.e., they cannot be defined over multiple track edges.
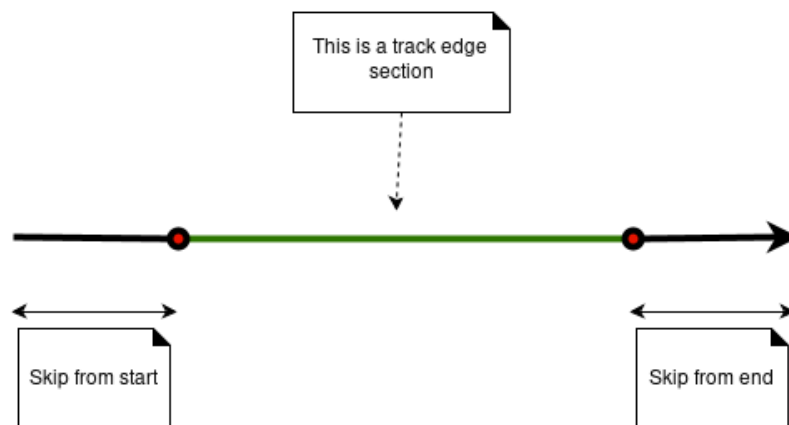
Figure 6 Illustration of Track Edge Section

The following similar classes from other models can be referred to:

- Track Edge Section in ⊞ SPT2TS-7 - RCA Digital Map – MAP Object Catalogue [RCA.Doc.69]

In RCA the TrackEdgeSection is a referenceable object having an ID. It could be defined in a separate container and then re-used in different objects by reference. In this model it is not possible - the infra.TrackEdgeSection can be used only in a composition relation. When the use case is to define areas (e.g., a contiguous track area or track area) use track edge sections as an attribute with 1..* multiplicity as in ⊟ SPT2TS-12698 - TvpSection or ▯ SPT2TS-123558 - Track Properties . [⚙ In Progress]

TrackEdgeSection
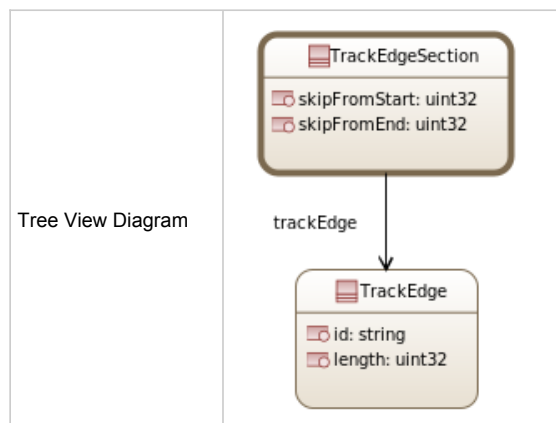
SPT2TS-49081 - TrackEdgeSection

{

  "structs": [

  {

   "name": "TrackEdgeSection",

   "info": "Defines a non-directed linear object (with a linear extension) located on a Track Edge",

   "belongsToSubPackage": "FunctionalArea",

   "attrs":[

     {"intId": 1, "name": "trackEdge", "reference": "TrackEdge", "info": "refers to track edge"},

     {"intId": 2, "name": "skipFromStart", "dataType": "uint32", "unit": "m", "exp": -3, "info": "ignore section [0..skipFromStart]."},

     {"intId": 3, "name": "skipFromEnd", "dataType": "uint32", "unit": "m", "exp": -3, "info": "ignore section [skipFromEnd, length]."}

   ]

  }]

}

DirTrackEdgeSection
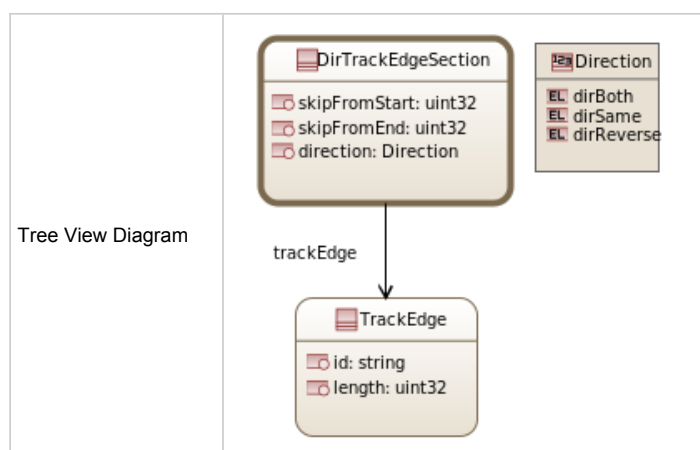
SPT2TS-125433 - DirTrackEdgeSection

{

  "structs": [

  {

```
"name": "DirTrackEdgeSection",
"info": "Defines a directed linear object (with a linear extension) located on a Track Edge",
"belongsToSubPackage": "FunctionalArea",
"attrs":[
    {"intId": 1, "name": "trackEdge", "reference": "TrackEdge", "info": "refers to track edge"},
    {"intId": 2, "name": "skipFromStart", "dataType": "uint32", "unit": "m", "exp": -3, "info": "ignore section
[0..skipFromStart]"},
    {"intId": 3, "name": "skipFromEnd", "dataType": "uint32", "unit": "m", "exp": -3, "info": "ignore section [skipFromEnd,
length]"},
    {"intId": 4, "name": "applicableDirection", "enumType": "Direction", "info": "defines the applicable direction of the
track edge"}
  ]
}]
}
```

2.2.3.3 Linked path

SPT2TS-63856 - Linked path defines a continuous path between two given reference points through different interconnected track edges. The reference points are defined using distance skips from start and end of track edges. The starting/ending point is to be calculated as a distance increment/decrement from start/end of track edge. The distance skips are defined on the track edges corresponding to the linked path. A linked path can span over different track edges building up a linear location. The linked path is only a container for directed edges, so it does not have an id and is used in other packages in composition-attributes.

The structure DirTrackEdge is only an auxiliary-container for TrackEdge-reference and used-direction. The list of DirTrackEdges in a linked path are defined in the order in which the train traverses them.
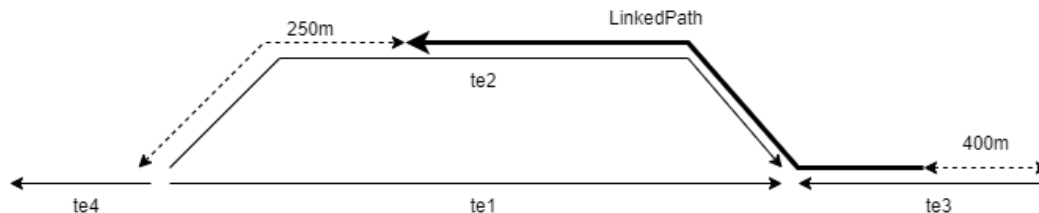

Figure 7 Illustration of Linked Path

The following equivalent classes from other models can be referred to:

- Linked path ⧉ SPT2TS-9 - LinX4Rail CDM ->PSM (Platform Specific Data Model)
- Linear Contiguous Track Area in ⧉ SPT2TS-7 - RCA Digital Map – MAP Object Catalogue [RCA.Doc.69]
- Linear Location in ⧉ SPT2TS-10 - RSM - RailSystemModel

In opposite to RCA and RSM the LinkedPath is not referenceable.

```
{
 "dirTrackEdges": [
   {"trackEdge": "te3", "sameDir": true},
   {"trackEdge": "te2", "sameDir": false}
 ],
 "skipFromPathStart": 400,
```

"skipFromPathEnd": 250

} [Open]

LinkedPath

| Tree View Diagram | |
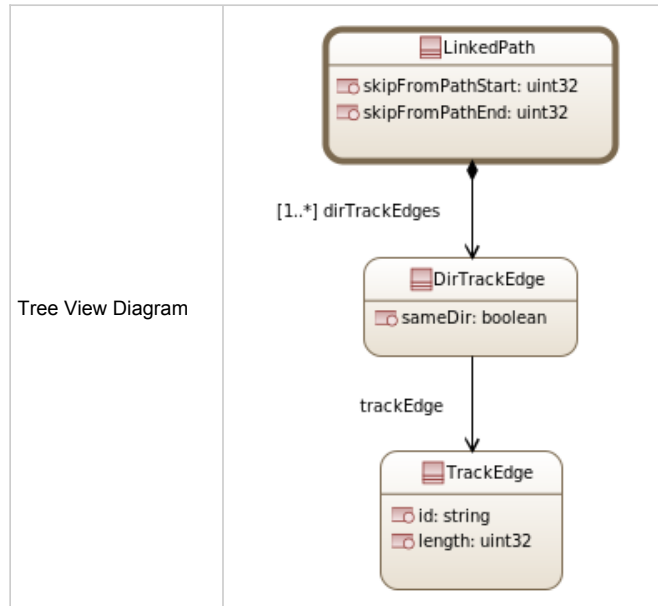
SPT2TS-49090 - DirTrackEdge

{

 "structs": [

{

  "name": "DirTrackEdge",

  "attrs": [

   {"intId": 1, "name": "trackEdge", "reference": "TrackEdge", "info": "refers to track edge"},

   {"intId": 2, "name": "sameDir", "dataType": "boolean", "info": "usage direction of the TrackEdge ; true: active in same direction as trackEdge, false: active in opposite direction"}

  ]

}]

}

SPT2TS-125513 - LinkedPath

{

 "structs": [

{

  "name": "LinkedPath",

  "info": "Defines a linked path on the topology",

  "attrs": [

   {"intId": 1, "name": "dirTrackEdges", "composition": "DirTrackEdge", "multiplicity": "1..*", "info": "composes of directed track edges", "sortedByKey": false},

   {"intId": 2, "name": "skipFromPathStart", "dataType": "uint32", "unit": "m", "exp": -3, "info": "ignore section [0..skipFromStart]"},

   {"intId": 3, "name": "skipFromPathEnd", "dataType": "uint32", "unit": "m", "exp": -3, "info": "ignore section [skipFromEnd, length]"}

```
    ]
  }
  ]
}
```

2.2.3.4 Track Area

SPT2TS-127643 - A Track Area groups an arbitrary number of Track Edge Sections. The sections don't have to be connected / adjacent to each other. The Track Area groups the sections to a logical entity, usually to illustrate a technical or functional context.

The abstract concept can be used to define topological extent for the concepts like Area Of Control. E.g., an Area of Control can be of a type Track Area which is defined for a relevant use case.

Figure 8 Illustration for Track Area

The following equivalent classes from other models can be referred to:

* Track Area in ⬚ SPT2TS-7 - RCA Digital Map – MAP Object Catalogue [RCA.Doc.69]

[🦎 Open]

SPT2TS-127642 - Track Area

```
{
 "structs": [
 {
  "name": "TrackArea",
  "info": "Defines a Track Area on the topology",
  "attrs": [
   {"intId": 1, "name": "trackEdgeSections", "composition": "TrackEdgeSection", "multiplicity": "1..*", "info": "composes of
track edge sections", "sortedByKey": false}
  ]
 }
 ]
}
```

2.2.3.5 Linked Area

SPT2TS-127644 - The Linked Area groups a number of Track Edges and Track Edge Sections, which are topologically connected to each other such that they form one or more paths. The illustrations below show different types of Linked Areas.
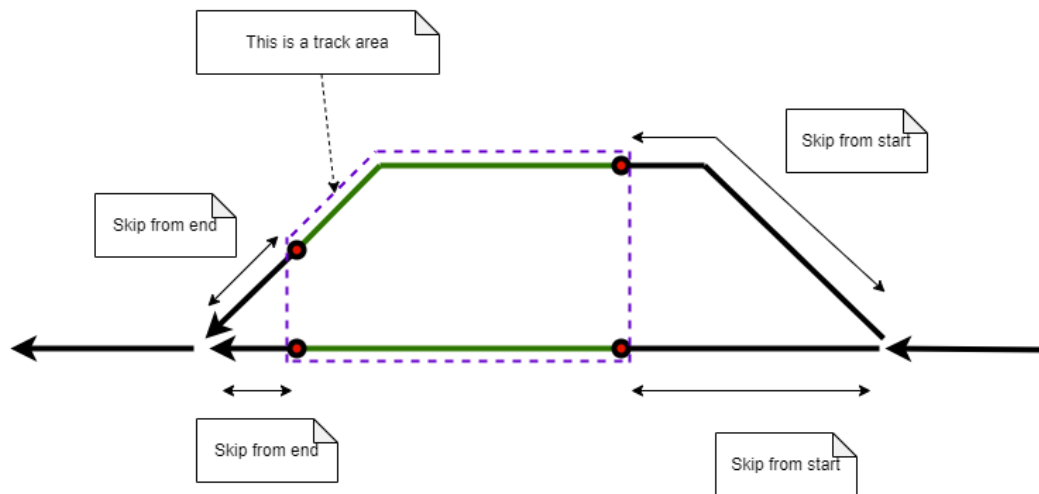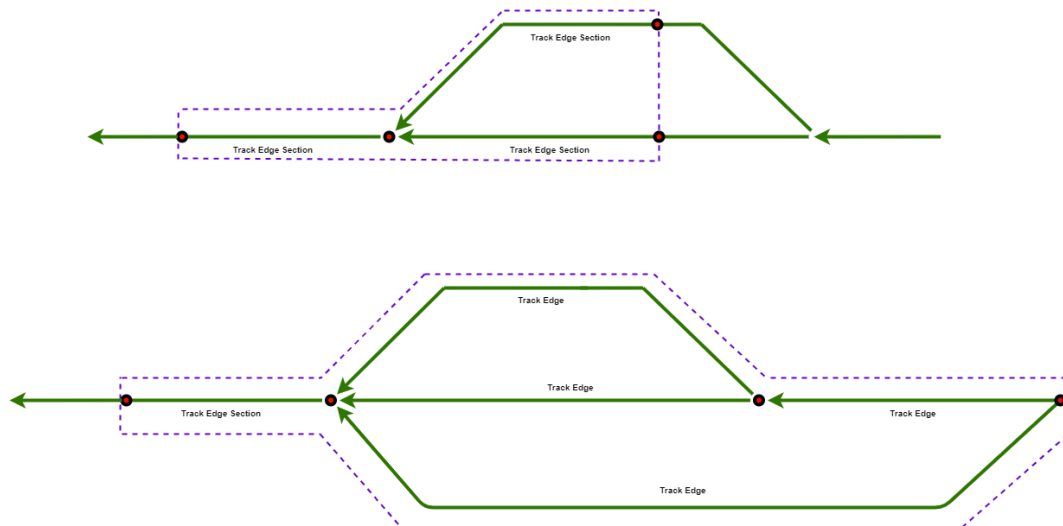

Figure 9 Illustration of Track Path

The following equivalent classes from other models can be referred to:

- Contiguous Track Area in ⬛ SPT2TS-7 *- RCA Digital Map – MAP Object Catalogue [RCA.Doc.69]*

[🔩 Open]

SPT2TS-127637 - Linked Area

```
{
 "structs": [
{
  "name": "LinkedArea",
  "attrs": [
   {"intId": 1, "name": "entireTrackEdges", "reference": "TrackEdge", "multiplicity": "*", "sortedByKey": true},
   {"intId": 2, "name": "trackEdgeSections", "composition": "TrackEdgeSection", "multiplicity": "*"}
 ]
}
]
}
```

2.3 Track Geometry

SPT2TS-49004 - Track geometry, also known as alignment, defines where the track centre line runs on the surface of the earth. This is typically achieved by defining track centre axes as a sequence of geometric parametric figures, straights, curves, clothoids, etc. It also includes cant definitions. Alternatively, track geometry is given as a suite of points in 3D space that samples the track centre line with an appropriate rate.

Coordinates fixate the track geometry on the surface of the earth and are defined in an appropriate coordinate reference system. **[🖊 Content to be approved]**

SPT2TS-49003 - The track geometry information is expected to be sufficiently consistent with the real track axis, so it fulfils all criteria of "reliable data".

Reliable Data refers to a typical characteristic of data which satisfies the qualities of being trustworthy/lower probability of incorrect information/lower fault rate of the information, current (not obsolete), complete, and accurate.

Along the life cycle and depending on data availability, several types of geometry are applied:

- design geometry is the idealised geometry as designed in a CAD tool, sometimes called "to-build" or "as-planned".
- as-built geometry describes the geometry as established after track works or tamping.
- as-is geometry is the surveyed geometry, possibly post-processed by the surveyor.

There is maximum (today national specific) tolerances which define the maximum deviation between design geometry (to-build, as-planned) and the real track axes ("as-built"). In addition, this track centre line is restored after each maintenance/tamping procedure. So to achieve maximum robustness and avoid additional error margins (i.e. from dynamical acquisition/surveys) the originally designed track geometry data is assumed to be used.

If this information is not available in a reliable (updated, accurate) way, the track geometry based on surveys ("as-is") is assumed to be used for use cases like ETCS or ATO engineering. However, use cases like maintenance (tamping) will always rely on the desired design situation ("to build").

The track geometry model shall support different levels of details for track description to support the following use cases:

1. Detailed track geometry data for exact track construction, engineering, maintenance or other use cases requiring the actual segments (such as straights, curves, transitions) of track geometry.
2. Approximate track geometry data optimised for efficient transmission and usage in real-time, e.g. for train protection systems or train operation (e.g. ATO segment profile)

[Content to be approved **]**

2.3.1 Track Edge Geometry

2.3.1.1 Input models / references

SPT2TS-64091 - The track edge geometry definition follows the definitions from SPT2TS-10 - RSM - RailSystemModel and SPT2TS-7 - RCA Digital Map – MAP Object Catalogue [RCA.Doc.69] . **[** Content to be approved **]**

2.3.1.2 Design decisions

SPT2TS-49017 -

The track geometry definition shall ensure that the geometric primitive elements are specified for each track edge, because

- it ensures that no backwards search along the track edges is required to identify the specific values (such as gradient, radius, etc.).
- "Breaking up" segments that straddle a track edge is computationally hard and should not be the responsibility of the data consumer.
- Assigning segments to areas and subdividing areas is computationally easier

[Content to be approved **]**

SPT2TS-49016 - The track geometry definition shall ensure that the entire extent (0..length) of the track edge is populated with the required geometric primitive elements. **[** Content to be approved **]**

SPT2TS-64092 - The track edge geometry has a list of one or more contiguous, i.e. abutting segments. A track segment is a parametrised geometric figure, i.e. line, arc or transition. **[**⊙🔍Content to be approved **]**

SPT2TS-49014 - The list of segments is ordered from start to end of the track edge. **[**⊙🔍Content to be approved **]**

SPT2TS-49038 - A segment is assigned to exactly one track edge geometry; a segment never straddles two track edges. **[**⊙🔍Content to be approved **]**

SPT2TS-49036 - Redundant transmission of start- and end-values (e.g. position, azimuth, radius,..) is avoided, which reduces data volume and the need to check plausibility by the data receiver (the next segment starts where the previous one ended). Instead,

- Initial values of a segment (e.g. position, azimuth, radius, ..) are always informed.
- Final/end values (e.g. azimuth, radius) of a segment are defined by the initial values of the next segment
- to describe the complete edge, a final segment with length=0 is required to define the end azimuth of a transition curve, if the last track edge ends with a transition curve.

[⊙🔍Content to be approved **]**

SPT2TS-49034 - The position along the track p informs where the segment starts (offset). Position p is the distance a train will travel along the 2D projection of the track on a horizontal mapping plane. The 3D projection of the track can be derived depending on the data consumer needs using 2D values and with required corrections. 2D was choosen as compromise to minimise the efforts spent for 3D data provisioning/engineering. **[**⊙🔍Content to be approved **]**

SPT2TS-123551 - The length of a segment is defined by the position of the next segment minus the position of the current segment (p_{n+1} - $p_{n)}$ **[**⊙🔍Content to be approved **]**

SPT2TS-49031 - Azimuth is the clockwise bearing with respect to the geographic north. **[**⊙🔍Content to be approved **]**

SPT2TS-49028 - Cant is commonly expressed as supereelevation, i.e. the height difference between the inner and outer rail along the curve because it's a convenient way of acquiring the measure in the field. Defining which is the inner and outer rail can be ambivalent but the left and right rail is well defined when given with respect to the orientation of the track edge. The cant here represents applied cant. **[**⊙🔍Content to be approved **]**

SPT2TS-49025 - The position on a map (coordinates) isn't given (moved to another part of the model) **[**⊙🔍Content to be approved **]**

SPT2TS-49045 -
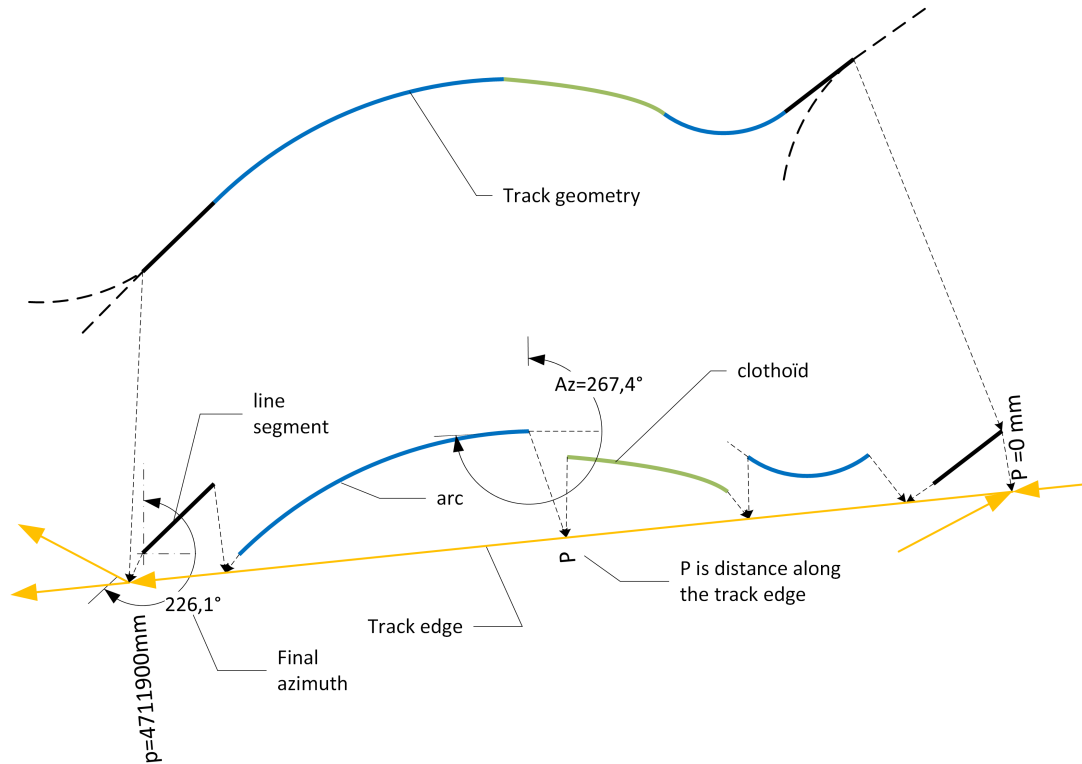
Figure 10 Top view of exact track geometry model. The exact geometry of a track edge is a list of track segments, ordered in the direction of the track edge (right to left in this figure). The position and azimuth of each segment are given.

The following equivalent classes from other models can be referred to:

- Horizontal alignment, Vertical alignment, and Lateral inclinations in SPT2TS-10 - RSM - RailSystemModel
- Curve, Gradient, and Cant in SPT2TS-7 - RCA Digital Map – MAP Object Catalogue [RCA.Doc.69]
- Horizontal alignment, Vertical alignment, and Lateral inclinations in SPT2TS-8 - EULYNX Data Model (dataprep.eulynx.eu)
- SPT2TS-9 - LinX4Rail CDM ->PSM (Platform Specific Data Model)

[Content to be approved]

GeometryArea
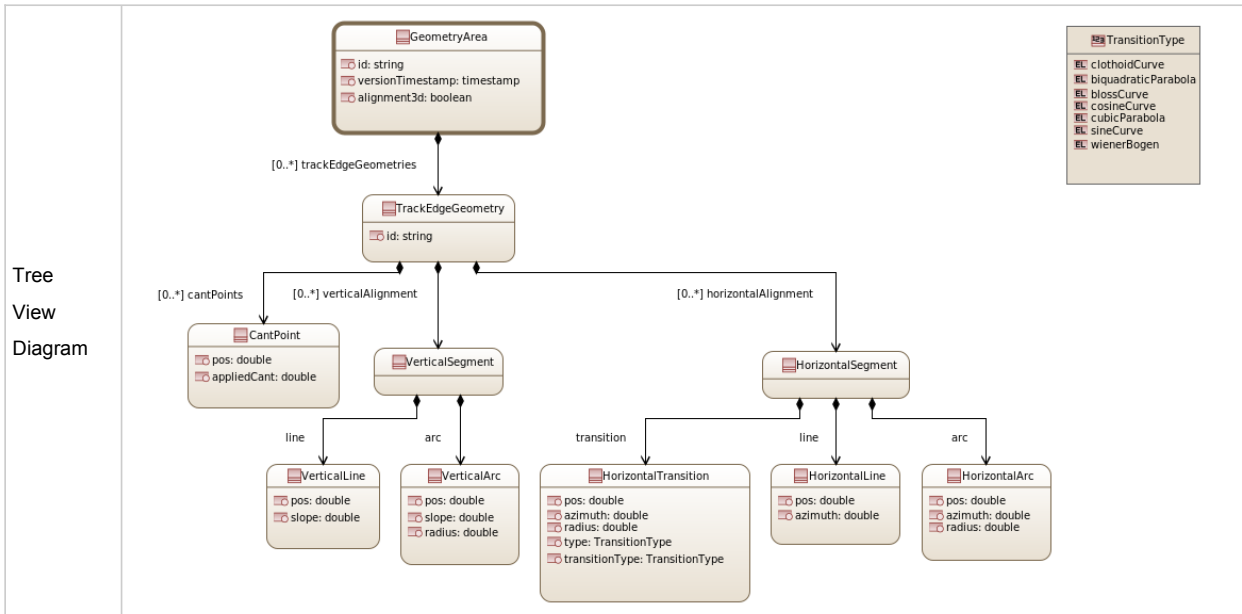
SPT2TS-49044 - GeometryArea

```
{
  "structs": [
  {
    "name": "GeometryArea",
    "attrs": [
      {"intId": 1, "name": "id", "dataType": "string", "key": "global", "sameKeyAs": "TopoArea", "info": "Identity of the object;
used for referencing"},
      {"intId": 2, "name": "versionTimestamp", "dataType": "timestamp", "info": "version information which is valid since
timestamp"},
      {"intId": 3, "name": "alignment3d", "dataType": "boolean",
       "info": "if true, all positions and length calculated on 3d line. if false - on 2d projection"},
      {"intId": 4, "name": "trackEdgeGeometries", "composition": "TrackEdgeGeometry",
       "multiplicity":  "0..*", "sortedByKey": true, "info": "composes of track edge geometries"}
    ]
  }]
}
```

SPT2TS-125443 - TrackEdgeGeometry

```
{
  "structs": [
  {
    "name": "TrackEdgeGeometry",
    "info": "Defines the geometry of the track edge centre line as a list of parametrised curves. The geometry is smooth;
no discontinuities in azimuth or inclination.",
    "belongsToSubPackage": "geometry",
    "attrs": [
      {"intId": 1, "name": "id", "dataType": "string", "key": "global", "sameKeyAs": "TrackEdge",
       "info": "Identity of the object; used for referencing"},
      {"intId": 2, "name": "horizontalAlignment", "composition": "HorizontalSegment",
       "multiplicity": "0..*", "info": "Sequence of geometric primitives, i.e. arc, line or clothoid, that compose this track
geometry", "sortedByKey": false},
      {"intId": 3, "name": "verticalAlignment", "composition": "VerticalSegment",
       "multiplicity": "0..*", "info": "Sequence of geometric primitives, i.e. arc, line or clothoid, that compose this track
```

```
geometry", "sortedByKey": false},
    {"intId": 4, "name": "cantPoints", "composition": "CantPoint", "multiplicity": "0..*", "info": "trackEdge must be completely
defined, at least at beginning and end. Empty means undefined", "sortedByKey": false}
  ]
}]
}
```

SPT2TS-125444 - HorizontalSegment

```
{
  "structs": [
  {
    "name": "HorizontalSegment",
    "info": "A curvi-linear geometric primitive, mathematically represents a horizontal segment of the centre line of a
railway track",
    "belongsToSubPackage": "geometry",
    "union": true,
    "attrs": [
      {"intId": 1, "name": "horizontalSegmentLine", "composition": "HorizontalLine", "info": "composes of lines"},
      {"intId": 2, "name": "horizontalSegmentArc", "composition": "HorizontalArc", "info": "composes of arcs"},
      {"intId": 3, "name": "horizontalSegmentTransition", "composition": "HorizontalTransition", "info": "composes of
transition curves"}
    ]
}]
}
```

SPT2TS-125445 - HorizontalLine

```
{
  "structs": [
  {
    "name": "HorizontalLine",
    "info": "Geometric primitive for a horizontal linear segment",
    "belongsToSubPackage": "geometry",
    "attrs": [
      {"intId": 1, "name": "trackGeometryPos", "dataType": "double", "unit": "m", "info": "position on the associated Track
Edge"},
      { "intId": 2, "name": "azimuth", "dataType": "double", "unit": "degree", "range": "0..360", "info": "The azimuth at pos, i.e.
bearing measured clockwise with respect to geographic north"}
    ]
}]
}
```

SPT2TS-125437 - HorizontalArc

```
{
  "structs": [
  {
    "name": "HorizontalArc",
    "info": "Geometric primitive for a horizontal arc of circle",
    "belongsToSubPackage": "geometry",
    "attrs": [
      {"intId": 1, "name": "trackGeometryPos", "dataType": "double", "unit": "m", "info": "position on the associated Track
Edge"},
      {"intId": 2, "name": "azimuth", "dataType": "double", "unit": "degree", "range": "0..360", "info": "The azimuth at pos,
i.e. bearing measured clockwise with respect to geographic north"},
      {"intId": 3, "name": "radius", "dataType": "double", "unit": "m",
```

```
      "info": "radius at pos, negative-counter-clockwise/left-curve, positive-clockwise/right-curve"}
    ]
  }]
}
```

SPT2TS-125438 - HorizontalTransition

```
{
  "structs": [
  {
    "name": "HorizontalTransition",
    "info": "Geometric primitive for a horizontal transition curve, a clothoid.",
    "belongsToSubPackage": "geometry",
    "attrs": [
      {"intId": 1, "name": "trackGeometryPos", "dataType": "double", "unit": "m", "info": "position on the associated Track
Edge"},
      {"intId": 2, "name": "azimuth", "dataType": "double", "unit": "degree", "range": "0..360",
       "info": "The azimuth at pos, i.e. bearing measured clockwise with respect to geographic north"},
      {"intId": 3, "name": "radius", "dataType": "double", "unit": "m",
       "info": "radius at pos, negative-counter-clockwise/left-curve, positive-clockwise/right-curve"},
      {"intId": 4, "name": "transitionType", "enumType": "TransitionType", "info": "defines different transition types"}
    ]
  }],
  "enums": [
  {
    "name": "TransitionType",
    "enumLiterals": [
      {"intId": 0, "name": "clothoidCurve"},
      {"intId": 1, "name": "biquadraticParabola"},
      {"intId": 2, "name": "blossCurve"},
      {"intId": 3, "name": "cosineCurve"},
      {"intId": 4, "name": "cubicParabola"},
      {"intId": 5, "name": "sineCurve"},
      {"intId": 6, "name": "wienerBogen"}
    ]
  }
  ]
}
```

SPT2TS-125439 - VerticalSegment

```
{
  "structs": [
  {
    "name": "VerticalSegment",
    "info": "A curvi-linear geometric primitive, mathematically represents a vertical segment of the centre line of a railway
track",
    "belongsToSubPackage": "geometry",
    "union": true,
    "attrs": [
      {"intId": 1, "name": "verticalSegmentLine", "composition": "VerticalLine", "info": "composes of lines"},
      {"intId": 2, "name": "verticalSegmentArc", "composition": "VerticalArc", "info": "composes of arcs"}
    ]
  }]
}
```

SPT2TS-125440 - VerticalLine

```
{
  "structs": [
  {
    "name": "VerticalLine",
    "info": "Geometric primitive for vertical linear segment",
    "belongsToSubPackage": "geometry",
    "attrs": [
      {"intId": 1, "name": "trackGeometryPos", "dataType": "double", "unit": "m", "info": "position on the associated Track
Edge"},
      {"intId": 2, "name": "slope", "dataType": "double", "unit": "permill",
       "info": "The vertical slope at pos, between the horizon and the track centre line. Viewed in the direction of the track
edge. Positive for rising slope and negative for falling slope."}
    ]
  }]
}
```

SPT2TS-125441 - VerticalArc

```
{
  "structs": [
  {
    "name": "VerticalArc",
    "info": "Geometric primitive for a vertical arc.",
    "belongsToSubPackage": "geometry",
    "attrs": [
      {"intId": 1, "name": "trackGeometryPos", "dataType": "double", "unit": "m", "info": "position on the associated Track
Edge"},
      {"intId": 2, "name": "slope", "dataType": "double", "unit": "permill",
       "info": "The vertical slope at pos, between the horizon and the track centre line. Viewed in the direction of the track
edge. Positive for rising slope and negative for falling slope."},
      {"intId": 3, "name": "radius", "dataType": "double", "unit": "m",
       "info": "radius at pos, negative-counter-clockwise/sag-curve, positive-clockwise/crest-curve"}
    ]
  }]
}
```

SPT2TS-125442 - CantPoint

```
{
  "structs": [
  {
    "name": "CantPoint",
    "info": "Cant at a given location on the track.",
    "attrs": [
      {"intId": 1, "name": "trackGeometryPos", "dataType": "double", "unit": "m",
       "info": "position on the associated track edge"},
      {"intId": 2, "name": "appliedCant", "dataType": "double", "unit": "m",
       "info": "Height of the right rail minus the left rail. Right/left with respect to the direction of the track edge.
Rollangle=(arcsin(cant/gauge)), gauge is distance between middle of the railheads, e.g. 1500mm for EU-standard gauge
'1435mm'"}
    ]
  }
  ]
}
```

SPT2TS-49042 - Example "Track Edge Geometry":
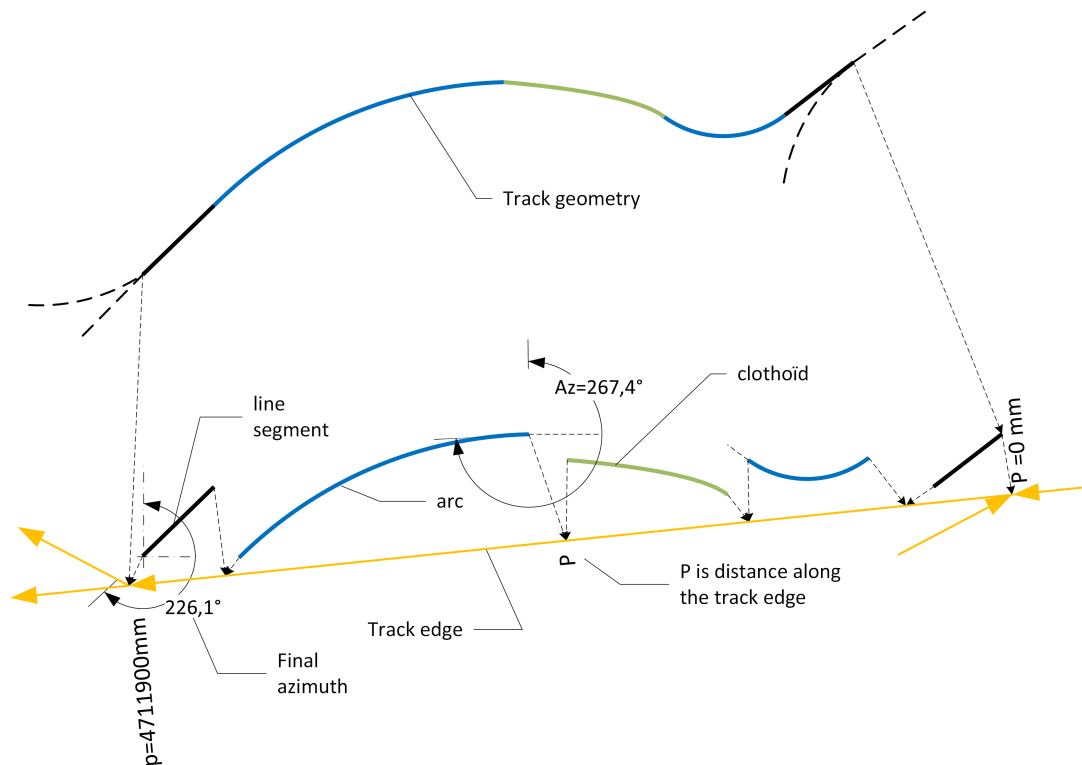
Figure 11 Illustration of Track Edge Geometry

```
{
  "id": "Area51",
  "trackEdgeGeometries": [
   {
    "id": "track1",
    "horizontalSegments": [
      {"horizontalSegmentLine": {"trackGeometryPos": 0, "radius": 0, "azimuth": 270000}},
      {"horizontalSegmentArc": {"trackGeometryPos": 51000, "radius": 480000, "azimuth": 270000}},
      {"horizontalSegmentTransition": {"trackGeometryPos": 151000, "radius": -480000, "azimuth": 170000, "type":
"clothoid"}},
      {"horizontalSegmentArc": {"trackGeometryPos": 251000, "radius": -1480000, "azimuth": 267400}},
      {"horizontalSegmentLine": {"trackGeometryPos": 651000, "radius": 0, "azimuth": 216400}}
    ]
   }
  ]
}
```

[🚀 Open]

SPT2-TransversalSystems
TCCS SD1 - Data Model_10_INFRA (rev. 407574)

2.3.2 Track Edge Sampled Geometry

2.3.2.1 Input models / References

SPT2TS-64093 - The track edge geometry definition follows the definitions ⊞ SPT2TS-9 - LinX4Rail CDM ->PSM (Platform Specific Data Model)

The following equivalent classes from other models can be referred to:

- ⊞ SPT2TS-9 - LinX4Rail CDM ->PSM (Platform Specific Data Model)

[↪ Open]

2.3.2.2 Design Decisions

SPT2TS-49013 - Sampled track geometry represents the track geometry as a list of sampled discrete points. This is typically used when the exact geometry isn't available or not required for the specific use case (e.g ATO Segment Profile). The sampling rate can't be prescribed but be such as to satisfy current use cases. By increasing the sampling rate (e.g. 10cm sample point distance), a very exact representation is possible as well if needed.
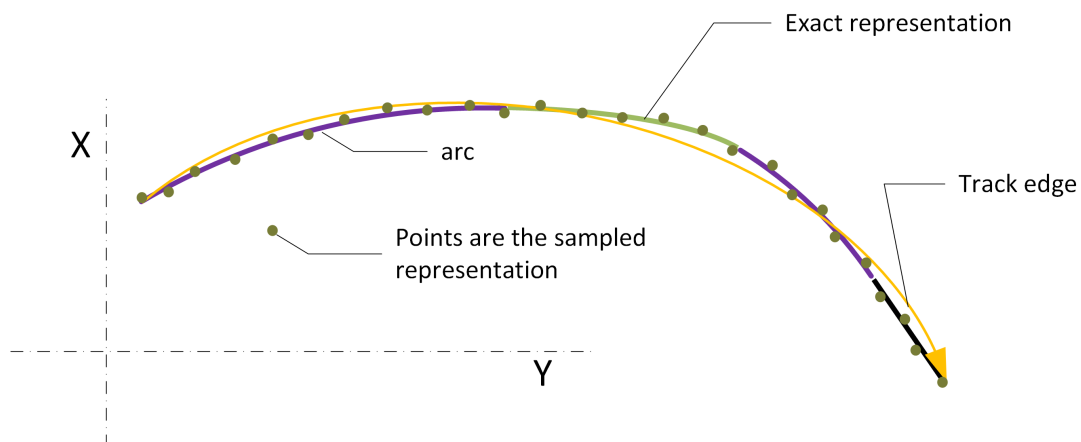
Figure 12: Top view of a sampled track geometry. A track edge is associated with a list of points, each of which is a sample of the track centre line.

[🔍 Content to be approved]

SPT2TS-49012 - The decision of Track Edge Geometry (see 🔖 SPT2TS-49033 - Design decisions)are also valid for the sampled track geometry, though the amount of information is reduced (e.g. azimuth, slope not required for simple model and not added). **[🔍 Content to be approved]**

SampledGeometryArea

24 | Page
2024-10-31 08:34

| Tree View Diagram | |
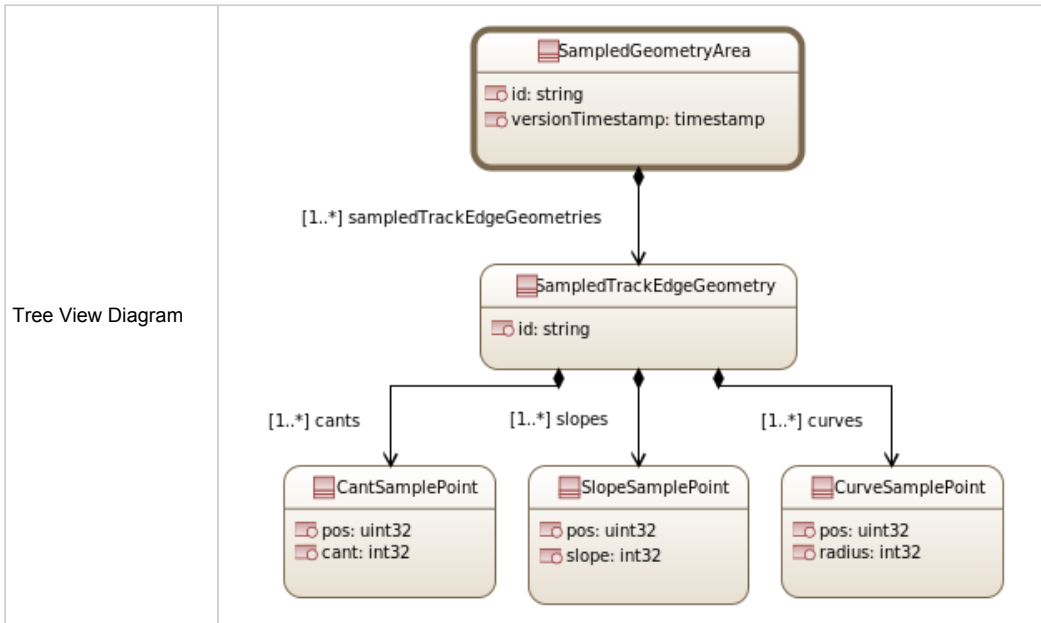
SPT2TS-49009 - SampledGeometryArea

```
{
  "structs": [
  {
    "name": "SampledGeometryArea",
    "attrs": [
      {"intId": 1, "name": "id", "dataType": "string", "key": "global", "sameKeyAs": "TopoArea", "info": "Identity of the object;
used for referencing"},
      {"intId": 2, "name": "versionTimestamp", "dataType": "timestamp", "info": "version information which is valid since
timestamp"},
      {"intId": 3, "name": "sampledTrackEdgeGeometries", "composition": "SampledTrackEdgeGeometry", "multiplicity":
"1..*", "sortedByKey": true, "info": "composes of sampled track edge geometries"}
    ]
  }]
}
```

SPT2TS-125447 - SampledTrackEdgeGeometry

```
{
  "structs": [
  {
    "name": "SampledTrackEdgeGeometry",
    "attrs": [
      {"intId": 1, "name": "id", "dataType": "string", "key": "global", "sameKeyAs": "TrackEdge",
        "info": "Identity of the object; used for referencing"},
      {"intId": 2, "name": "slopes", "composition": "SlopeSamplePoint", "multiplicity": "1..*", "sortedByKey": false, "info":
"composes of slope sample points"},
      {"intId": 3, "name": "curves", "composition": "CurveSamplePoint", "multiplicity": "1..*", "sortedByKey": false, "info":
"composes of curve sample points"},
      {"intId": 4, "name": "cants", "composition": "CantSamplePoint", "multiplicity": "1..*", "sortedByKey": false, "info":
"composes of cant sample points"}
    ]
  }]
}
```

SPT2TS-125448 - SlopeSamplePoint

```
{
  "structs": [
  {
    "name": "SlopeSamplePoint",
    "info": "Slope (gradient) of the track at a given location on the track.",
    "attrs": [
      {"intId": 1, "name": "pos", "dataType": "uint32", "unit": "m", "exp": -3,
       "info": "position on the associated TrackEdge"},
      {"intId": 2, "name": "samplePointSlope", "dataType": "int32", "unit": "permill", "exp": -1,
        "info": "The vertical slope at pos, between the horizon and the track centre line. Viewed in the direction of the track edge. Positive for rising slope and negative for falling slope."}
    ]
  }]
}
```

SPT2TS-125449 - CantSamplePoint

```
{
  "structs": [
  {
    "name": "CantSamplePoint",
    "info": "Cant(superelevation) of the track at a given location on the track.",
    "attrs": [
      {"intId": 1, "name": "pos", "dataType": "uint32", "unit": "m", "exp": -3,
       "info": "position on the associated TrackEdge"},
      {"intId": 2, "name": "samplePointCant", "dataType": "int32", "unit": "m", "exp": -3,
        "info": "Defines appliedCant as the Height of the right rail minus the left rail. Right/left with respect to the direction of the track edge. Rollangle=(arcsin(cant/gauge), gauge is distance between centers of rail-heads, e.g. 1500mm for EU-standard gauge."}
    ]
  }]
}
```

SPT2TS-125446 - CurveSamplePoint

```
{
  "structs": [
  {
    "name": "CurveSamplePoint",
    "info": "Curve (radius) of the track at a given location on the track.",
    "attrs": [
      {"intId": 1, "name": "pos", "dataType": "uint32", "unit": "m", "exp": -3,
       "info": "position on the associated TrackEdge"},
      {"intId": 2, "name": "sampleCurveRadius", "dataType": "int32", "unit": "m", "exp": -3,
       "info": "radius at pos, negative-counter-clockwise/left-curve, positive-clockwise/right-curve"}
    ]
  }]
}
```

SPT2TS-49007 - Example "Track Edge Sampled Geometry":
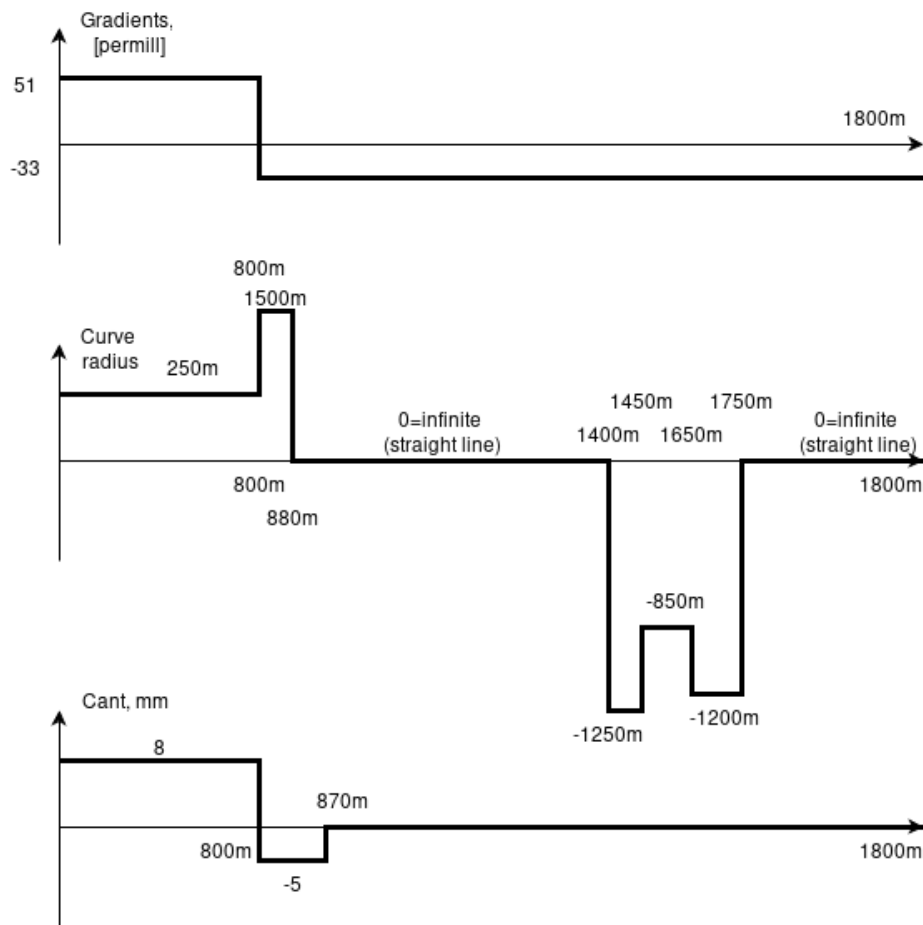
Figure 13 Example for a track edge geometry.

```
{
  "id": "Area51",
  "trackEdgeGeometries": [
   {
     "id": "track1",
     "slopes": [
       {"pos": 0, "samplePointSlope": 51000},
       {"pos": 800000, "samplePointSlope": -33000}
     ],
     "curves": [
       {"pos": 0, "sampleCurveRadius": 250000}},
       {"pos": 800000, "sampleCurveRadius": 1500000},
       {"pos": 880000, "sampleCurveRadius": 0},
       {"pos": 1400000, "sampleCurveRadius": -1250000},
       {"pos": 1450000, "sampleCurveRadius": -850000},
       {"pos": 1650000, "sampleCurveRadius": -1200000},
       {"pos": 1750000, "sampleCurveRadius": 0}
     ],
     "cants": [
```

```
    {"pos": 0, "samplePointCant": 8},

    {"pos": 800000, "samplePointCant": -5},

    {"pos": 870000, "samplePointCant": 0}

  ]

 },

 ... other trackEdgeGeometries

]
```

} [🟢 Open]

2.4 Point, Crossing, Derailer

SPT2TS-63857 - Point, crossing and derailer are track assets / field elements. They contain movable components which are equipped with actuators in order to set / change the positions of the field element.
Points are controlled field elements which represent a branching point in the track network wherein exists a possibility to continue either on the right or on the left branch:

- Simple points (with and without movable frogs)
- Single- and double-slip crossings

In a simple point, a branch track branches off from a main track. A simple point offers two driving possibilities i.e. one along main track and one along the branch track. The required direction of travel can be set using the point machines.

A single or double slip crossing are between two tracks without grade of separation and with possibility to switch between straight passing track and turning track.

Derailer is a safety device attached to a rail, which when operated can cause a derailment of a train that is making an unauthorised movement. A derailer prevents rail vehicles from driving on the track beyond the point secured by the derailer.
Derailers are intended to prevent accidents caused by cold movements, for example rolling of parked rail vehicles which were not secured against unintentional movements.

A track crossing between two tracks without grade of separation and without possibility to switch between straight passing track and turning track. In other words, diamond crossing is the overlapping/intersection of two tracks at the same level without the possibility of changing tracks. There are two possible types of a diamond crossing, one with movable tongues (switched diamond crossing) and one without movable parts (non-switched diamond crossing)
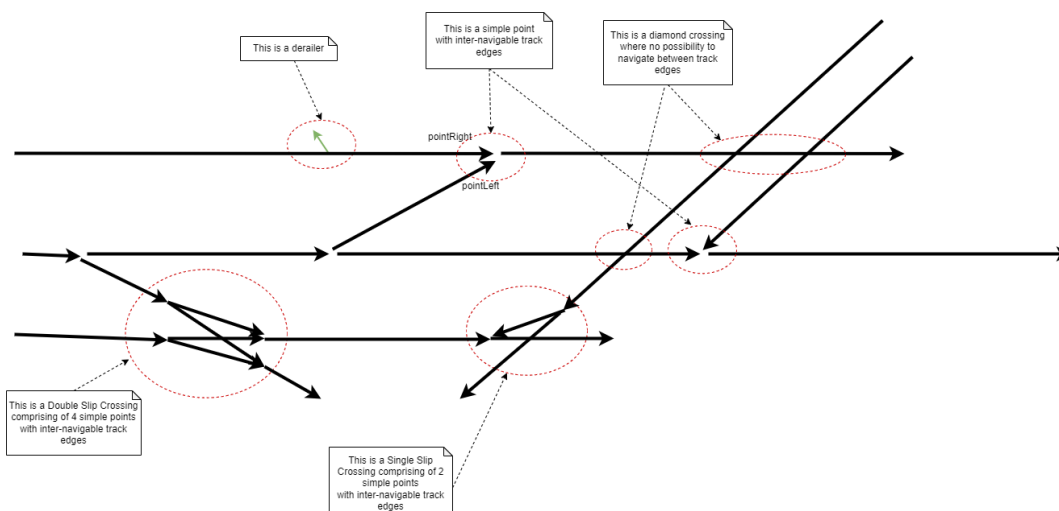
Figure 14 Meaning of attributes in classes

The following equivalent classes from other models can be referred to:

- Point, Crossing, and Derailer in ⊞ SPT2TS-7 - RCA Digital Map – MAP Object Catalogue [RCA.Doc.69]
- Crossing, Derailer, Buffer Stop, Rail Switch in ⊞ SPT2TS-9 - LinX4Rail CDM ->PSM (Platform Specific Data Model)
- Crossing and Turnout in ⊞ SPT2TS-10 - RSM - RailSystemModel

[⇥ Open]

2.4.1 Design Decisions

SPT2TS-123574 - In case of crossing remains the question, how this functional element is represented in the topology Map:

1. without splitting the track edges (the Crossing-object defines location of the crossing on the track edge)
2. by splitting TrackEdges at the crossing and introducing TrackEdgeLinks with usableDir == None.

The first option was selected as much more compact, without loss of information. The crossing in this case contains two TrackEdgePoints marking the crossing locations on TrackEdges, rather than defining a list of TrackEdgeLinks, which would be the case for second option. **[⊙ Content to be approved]**

SimplePoint
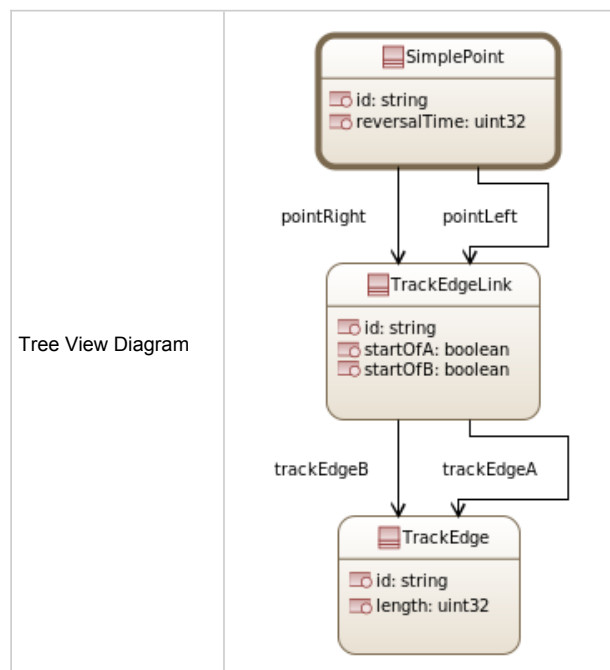
SPT2TS-49048 - SimplePoint

```
{
  "structs": [
  {
    "name": "SimplePoint",
    "info": "Defines the physical track asset Simple Point",
    "belongsToSubPackage": "FunctionalArea",
    "attrs": [
      {"intId": 1, "name": "id", "dataType": "string", "key": "global", "info": "Identity of the object; used for referencing ; struct
```

ure refers to EULYNX EU.SAS.77"},

    {"intId": 2, "name": "reversalTime", "dataType": "uint32", "unit": "s", "exp": -1, "info": "Defines the time in seconds to switch from one position to the other ; 0 means unknown"},

    {"intId": 3, "name": "pointLeft", "reference": "TrackEdgeLink", "info": "Track Edge Link referencing the left side of the point in the facing direction of the simple point"},

    {"intId": 4, "name": "pointRight", "reference": "TrackEdgeLink", "info": "Track Edge Link referencing the right side of the point in the facing direction of the simple point"}

  ]

 }]

}

SlipCrossing

Tree View Diagram
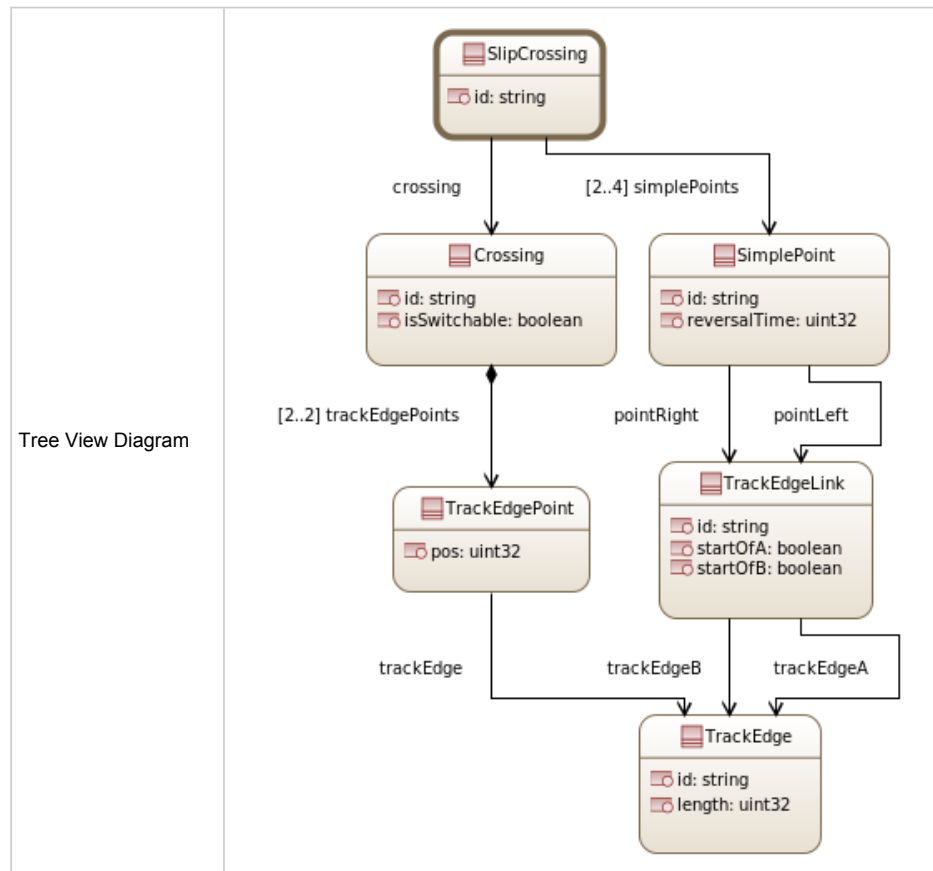
SPT2TS-125450 - SlipCrossing

{

 "structs": [

 {

  "name": "SlipCrossing",

  "info": "Defines the physical track asset Slip Crossing ; 2 instances simple point define a single slip crossing and 4 instances of simple point defined a double slip crossing ; Reversal time is the maximum of reversal times of simple points",

  "belongsToSubPackage": "FunctionalArea",

  "attrs":[

    {"intId": 1, "name": "id", "dataType": "string", "key": "global", "info": "Identity of the object; used for referencing"},

    {"intId": 2, "name": "simplePoints", "reference": "SimplePoint", "multiplicity":"2..4", "info":"Consists of simple points"},

    {"intId": 3, "name": "crossing", "reference": "Crossing","info":"Consists of crossing"}

  ]

 }]

}

Derailer

Tree View Diagram
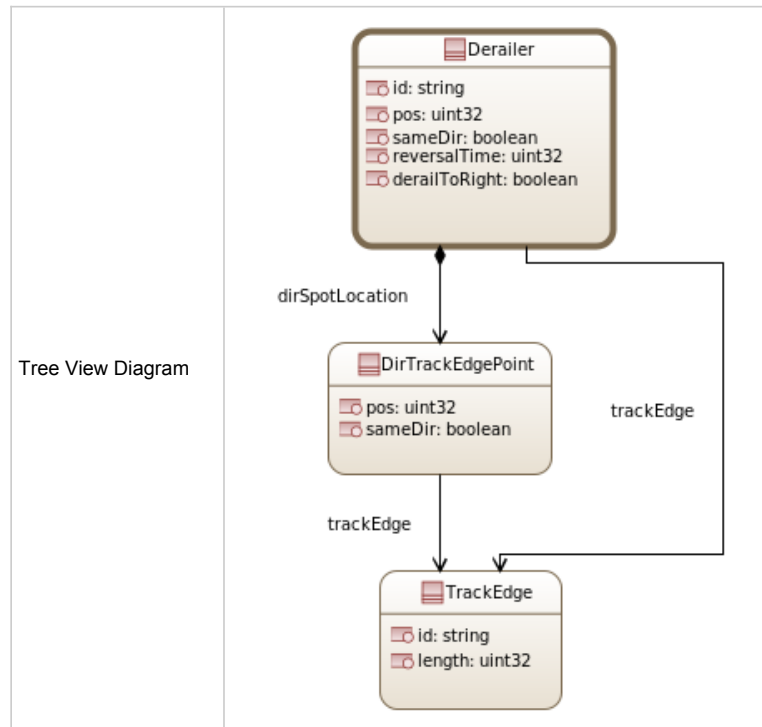
SPT2TS-125451 - Derailer

```
{
  "structs": [
  {
    "name": "Derailer",
    "info": "Defines the physical track asset Derailer",
    "belongsToSubPackage": "FunctionalArea",
    "attrs": [
        {"intId": 1, "name": "id", "dataType": "string", "key": "global", "info": "Identity of the object; used for referencing"},
        {"intId": 2, "name": "dirSpotLocation", "composition": "DirTrackEdgePoint", "info": "defines position on TrackEdge"},
        {"intId": 3, "name": "reversalTime", "dataType": "uint32", "unit": "s", "exp": -1, "info": "Defines the time in seconds to switch from one position to the other ; 0 means unknown"},
        { "intId": 4, "name": "derailToRight", "dataType": "boolean", "info": "if false = derailToLeft, the side at which the train will be derailed"}
    ]
  }]
}
```

Crossing

Tree View Diagram	
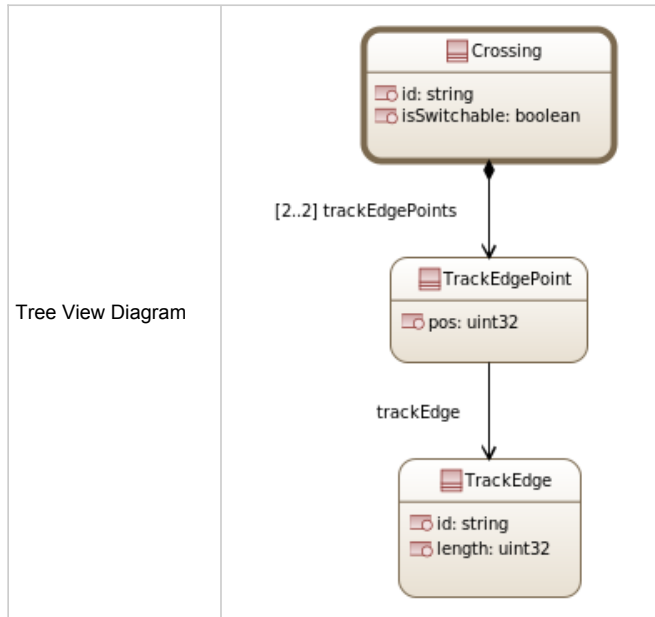
SPT2TS-125452 - Crossing

```
{
 "structs": [
 {
  "name": "Crossing",
  "info": "Defines the physical track asset Crossing without possibility to switch between two track edges",
  "belongsToSubPackage": "FunctionalArea",
  "attrs": [
     {"intId": 1, "name": "id", "dataType": "string", "key": "global", "info": "Identity of the object; used for referencing"},
     {"intId": 2, "name": "trackEdgePoints", "composition": "TrackEdgePoint","multiplicity":"2", "info": "composes of track edge points"},
     { "intId": 3, "name": "isSwitchable", "dataType": "boolean", "info": "True: If the crossing is switchable ; False: If the crossing is not switchable ; https://commons.wikimedia.org/wiki/Category:Switched_diamond_crossings "}
  ]
 }]
}
```

2.5 Balise (Group)

SPT2TS-63858 - Balises (or Eurobalises in the context of ETCS) are technical devices on the railway track bed that store information (telegram) and transmit it to rail vehicles passing the location of the Balise. The balises shall be represented using a Track Edge Point on the topology and are not associated with a direction.
Balises are clustered in groups and each balise group can consist a maximum of 8 Balises. Orientation direction of the balises are explicitly provided.

The following equivalent classes from other models can be referred to:

- Balise and Balise Group in SPT2TS-7 - RCA Digital Map – MAP Object Catalogue [RCA.Doc.69]
- Balise and Balise Group in SPT2TS-9 - LinX4Rail CDM ->PSM (Platform Specific Data Model)
- Balise and Balise Group in https://eulynx.eu/index.php/dataprep
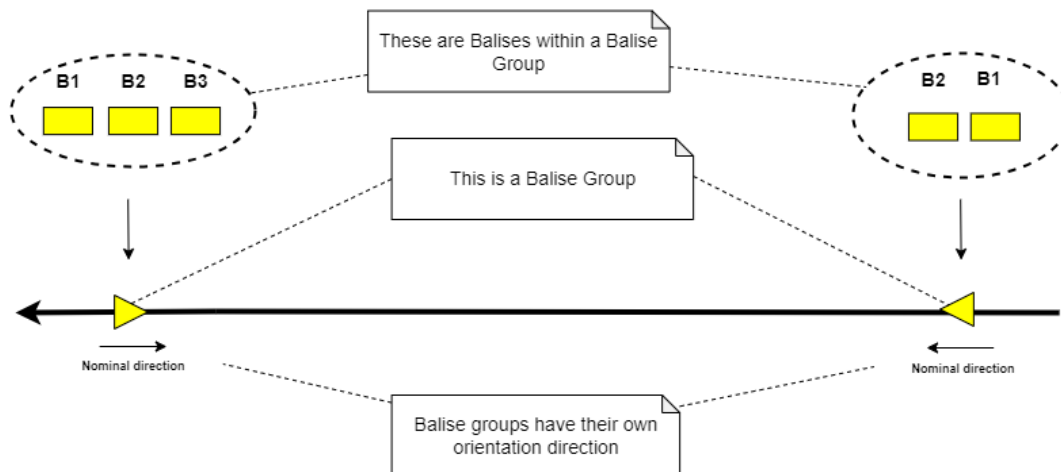
Figure 15 Balises and Balise Groups

[🔗 Open]

BaliseGroup

Tree View Diagram
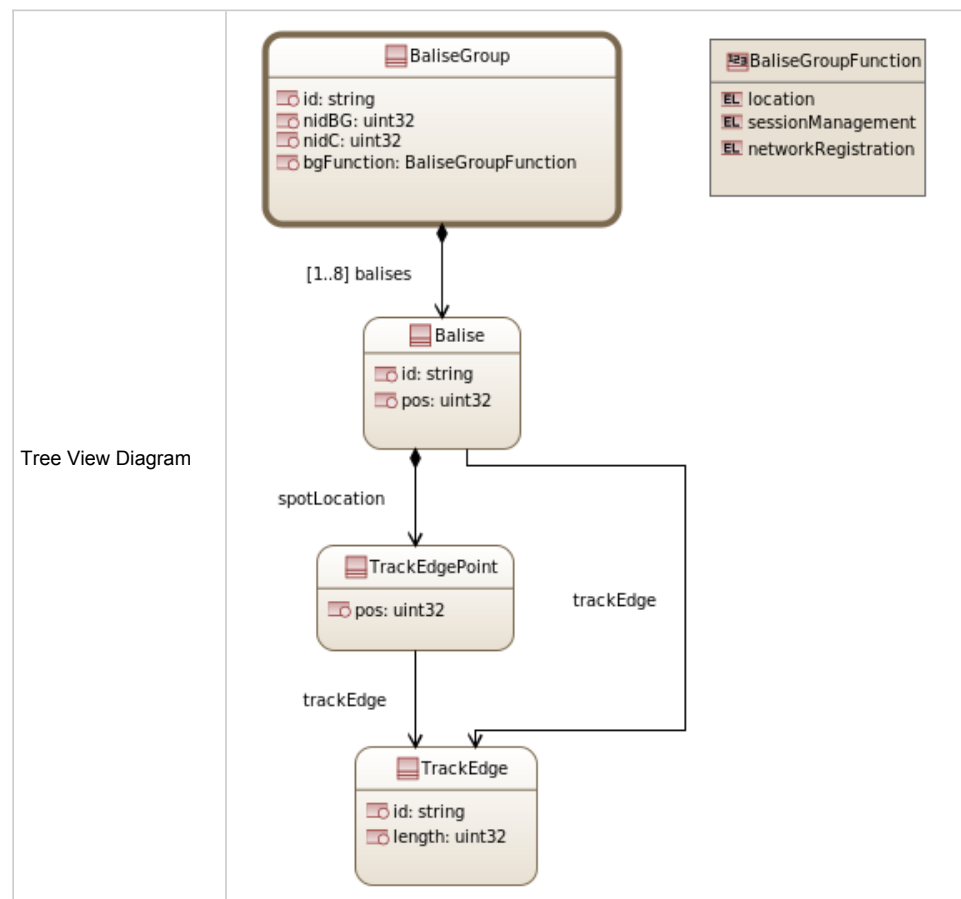
SPT2TS-125471 - BaliseGroup

```
{
  "structs": [
  {
    "name": "BaliseGroup",
    "info": "Defines a group of balises.",
```

```
    "belongsToSubPackage": "FunctionalArea",
    "attrs":[
        {"intId": 1, "name": "id", "dataType": "string", "key": "global", "info": "Identity of the object; used for referencing"},
        {"intId": 2, "name": "nid_bg", "dataType": "uint32", "range": "0..16383", "info": "national identifier of the balise
group"},
        {"intId": 3, "name": "nid_c", "dataType": "uint32", "range": "0..1023", "info": "country identifier of the balise group"},
        {"intId": 4, "name": "balises", "composition": "Balise", "multiplicity": "1..8", "info": "Defines the balises within the
balise group. ETCS-positionInGroup is equal to the position this array"},
        {"intId": 5, "name": "bgFunction",  "enumType": "BaliseGroupFunction", "info": "to be clarified, if derivable from
baliseTelegram"},
        {"intId": 6, "name": "linkOrientationDirection", "enumType": "Direction", "info": "defines linking orientation direction
along the TrackEdge"}
    ]
 },
 {
    "name": "Balise",
    "info": "Defines a balise.",
    "belongsToSubPackage": "FunctionalArea",
    "attrs":[
        {"intId": 1, "name": "id", "dataType": "string", "key": "global", "info": "Identity of the object; used for referencing"},
        {"intId": 2, "name": "spotLocation", "composition": "TrackEdgePoint", "info": "location on TrackEdge"}
    ]
 }
],
"enums": [
{
  "name": "BaliseGroupFunction",
  "enumLiterals": [
    {"intId": 0, "name": "location", "info": "has a location function"},
    {"intId": 1,"name": "sessionManagement", "info": "has a session management function"},
    {"intId": 2, "name": "networkRegistration", "info": "has a network registration function"}
  ]
 }]
}
```

2.5.1 Example for balise group

```
{
 "id": "bg2",
 "balises": [
   {"id": "b1", "spotLocation": {"trackEdge": "te1", "pos": 1000}},   <--- with position in group == 0
   {"id": "b2", "spotLocation": {"trackEdge": "te1", "pos": 4000}},   <--- with position in group == 1
   {"id": "b3", "spotLocation": {"trackEdge": "te1", "pos": 7000}},   <--- with position in group == 2
   {"id": "b4", "spotLocation": {"trackEdge": "te1", "pos": 10000}}    <--- with position in group == 3
 ]
}
```

2.6 Track Properties

SPT2TS-63859 - Track properties is a container for different classes which define properties for a section of track. The following properties are defined as a part of it:

1. Track Condition: Defines the type of the condition valid for a section of track.
2. Track Edge Loading Gauge: Defines the loading gauge applicable for a section of track.
3. Track Edge Gauge: Defines the gauge (i.e., 1000mm, 1435mm, ...) applicable for a section of track.
4. Current Limitation: Defines the applicable current limitation in Amperes for a section of track.
5. Permitted Braking Distance: Defines the permitted braking distance for the train to come to stop before danger areas like Level Crossing, unplanned Radio holes, etc
6. Traction System: Defines the equipped traction system on the track.

The following equivalent classes from other models can be referred to:

- Track Condition, Loading Gauge, Gauge, Traction System, Current Limitation, and Permitted Braking Distance in SPT2TS-7 - RCA Digital Map – MAP Object Catalogue [RCA.Doc.69]
- Loading Gauge, Gauge, Voltage, Current Limitation, and Permitted Braking Distance in SPT2TS-9 - LinX4Rail CDM ->PSM (Platform Specific Data Model)

[Open]

PropertiesArea
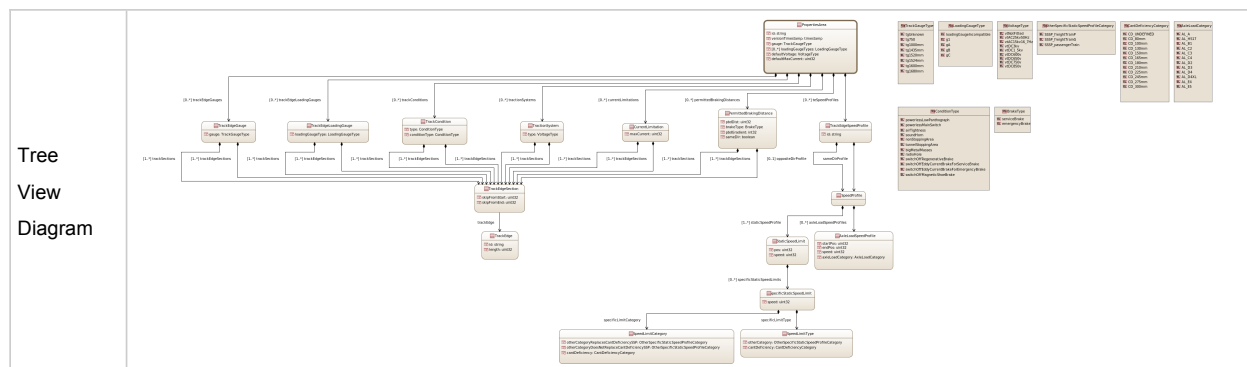
SPT2TS-63833 - TrackProperties enumerations

```
{
  "enums": [
    {
      "name": "LoadingGaugeType",
      "info": "is coming from ETCS packets 11,12 mLoadingGauge",
      "enumLiterals": [
        {"intId": 0, "name": "loadingGaugeIncompatible", "info": "The train does not fit to any of the interoperable loading
gauge profiles"},
        {"intId": 1, "name": "g1", "info": "belongs to type g1"},
        {"intId": 2, "name": "gA", "info": "belongs to type gA"},
        {"intId": 3, "name": "gB", "info": "belongs to type gB"},
        {"intId": 4, "name": "gC", "info": "belongs to type gC"}
      ]
    },
    {
```

```
      "name": "ConditionType",
      "info": "List of track conditions",
      "enumLiterals": [
        { "intId": 0, "name": "powerlessLowPanthograph", "info": "Powerless section, lower Pantograph"},
        { "intId": 1, "name": "powerlessMainSwitch", "info": "Powerless section, switch off main power switch"},
        { "intId": 2, "name": "airTightness", "info": "defines a air tightness section"},
        { "intId": 3, "name": "soundHorn", "info": "defines a sound horn section"},
        { "intId": 4, "name": "nonStoppingArea", "info": "defines a non stopping area"},
        { "intId": 5, "name": "tunnelStoppingArea", "info": "defines a tunnel stopping area"},
        { "intId": 6, "name": "bigMetalMasses", "info": "Big metal masses, ignore on-board integrity check alarm of balise
transmission"},
        { "intId": 7, "name": "radioHole", "info": "stop supervision of the loss of safe radio connection"},
        { "intId": 8, "name": "switchOffRegenerativeBrake", "info": "defines a switch Off Regenerative Brake sections"},
        { "intId": 9, "name": "switchOffEddyCurrentBrakeForServiceBrake", "info": "defines a switch Off Eddy Current Brake
sections"},
        { "intId": 10, "name": "switchOffEddyCurrentBrakeForEmergencyBrake", "info": "defines a switch Off Eddy Current
Emergency Brake sections"},
        { "intId": 11, "name": "switchOffMagneticShoeBrake", "info": "defines a switch Off Magnetic Shoe Brake sections"}

      ]
    },
    {
        "name": "BrakeType",
        "enumLiterals": [
            { "intId": 0, "name": "serviceBrake", "info": "defines service brake"},
            { "intId": 1, "name": "emergencyBrake", "info": "defines emergency brake"}
        ]
    },
    {
      "name": "VoltageType",
      "info":"List defining Voltage types",
      "enumLiterals": [
        {"intId": 0, "name": "vtNotFitted", "info": "Line not fitted (not electrified) with any traction system"},
        {"intId": 1, "name": "vtAC25kv50Hz", "info": "AC 25kV 50Hz",
         "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/energy-supply-systems/rinf/AC10"}},
        {"intId": 2, "name": "vtAC15kv16_7Hz", "info": "AC 15kV 16.7Hz",
         "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/energy-supply-systems/rinf/AC20"}},
        {"intId": 3, "name": "vtDC3kv", "info": "DC 3kV",
         "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/energy-supply-systems/rinf/DC30"}},
        {"intId": 4, "name": "vtDC1_5kv", "info": "DC 1.5kV",
         "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/energy-supply-systems/rinf/DC40"}},
        {"intId": 5, "name": "vtDC600v", "info": "DC 600V",
         "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/energy-supply-systems/rinf/DC80"}},
        {"intId": 6, "name": "vtDC650v", "info": "DC 650V",
         "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/energy-supply-systems/rinf/DC70"}},
        {"intId": 7, "name": "vtDC750v", "info": "DC 750V",
         "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/energy-supply-systems/rinf/DC60"}},
        {"intId": 8, "name": "vtDC850v", "info": "DC 850V",
         "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/energy-supply-systems/rinf/DC90"}}
      ]
    },
    {
      "name": "TrackGaugeType",
```

```json
      "enumLiterals": [
        {"intId": 0, "name": "tgUnknown", "info": "Line not fitted (not electrified) with any traction system"},
        {"intId": 1, "name": "tg750", "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/nominal-track-gauges/rinf/10"}, "info":"gauge type 750mm"},
        {"intId": 2, "name": "tg1000mm", "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/nominal-track-gauges/rinf/20"}, "info":"gauge type 1000mm"},
        {"intId": 3, "name": "tg1435mm", "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/nominal-track-gauges/rinf/30"}, "info":"gauge type 1435mm"},
        {"intId": 4, "name": "tg1520mm", "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/nominal-track-gauges/rinf/40"}, "info":"gauge type 1520mm"},
        {"intId": 5, "name": "tg1524mm", "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/nominal-track-gauges/rinf/50"}, "info":"gauge type 1524mm"},
        {"intId": 6, "name": "tg1600mm", "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/nominal-track-gauges/rinf/60"}, "info":"gauge type 1600mm"},
        {"intId": 7, "name": "tg1680mm", "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/nominal-track-gauges/rinf/70"}, "info":"gauge type 1680mm"}
      ]
  }]
}
```

SPT2TS-125457 - PropertiesArea

```json
{
  "structs": [
  {
    "name": "PropertiesArea",
    "info": "the object itself contains default-values for the area. Exceptions are in trackConditions etc.",
    "attrs": [
      {"intId": 1, "name": "id", "dataType": "string", "key": "global", "sameKeyAs": "TopoArea", "info": "Identity of the object; used for referencing"},
      {"intId": 2, "name": "versionTimestamp", "dataType": "timestamp", "info": "version information which is valid since timestamp"},
      {"intId": 3, "name": "gauge", "enumType": "TrackGaugeType", "info": "default gauge in the area. Deviations are covered in trackEdgeGauges"},
      {"intId": 4, "name": "loadingGaugeTypes", "enumType": "LoadingGaugeType", "multiplicity": "*", "info": "list of supported loading gauges per default"},
      {"intId": 5, "name": "defaultVoltage", "enumType": "VoltageType", "info": "defines default Voltage"},
      {"intId": 6, "name": "defaultMaxCurrent", "dataType": "uint32", "unit": "A", "info": "used in ATO & ETCS.packet40"},
      {"intId": 7, "name": "teSpeedProfiles", "composition": "TrackEdgeSpeedProfile", "multiplicity": "*", "sortedByKey": true, "info": "composes of track edge speed profiles"},
      {"intId": 8, "name": "trackEdgeGauges", "composition": "TrackEdgeGauge", "multiplicity": "*", "info": "composes of track edge gauges"},
      {"intId": 9, "name": "trackEdgeLoadingGauges", "composition": "TrackEdgeLoadingGauge", "multiplicity": "*", "info": "composes of track edge loading gauges"},
      {"intId": 10, "name": "trackConditions", "composition": "TrackCondition", "multiplicity": "*", "info": "composes of track conditions"},
      {"intId": 11, "name": "currentLimitations", "composition": "CurrentLimitation", "multiplicity": "*", "info": "composes of current limitation"},
      {"intId": 12, "name": "permittedBrakingDistances", "composition": "PermittedBrakingDistance", "multiplicity": "*", "info": "composes of permitted braking distances"},
      {"intId": 13, "name": "tractionSystems", "composition": "TractionSystem", "multiplicity": "*", "info": "composes of traction systems"}
    ]
  }]
```

```
}
```

SPT2TS-125458 - TrackEdgeGauge

```
{
  "structs": [
  {
    "name": "TrackEdgeGauge",
    "info": "inner distance between two rails, e.g. 1432 for EU-standard-rails",
    "attrs": [
      {"intId": 1, "name": "trackEdgeSections", "composition": "TrackEdgeSection", "multiplicity": "1..*", "info": "composes
of track edge sections"},
      {"intId": 2, "name": "gauge", "enumType": "TrackGaugeType", "info": "defines different track gauges"}
    ]
  }]
}
```

SPT2TS-125459 - TrackEdgeLoadingGauge

```
{
  "structs": [
  {
    "name": "TrackEdgeLoadingGauge",
    "attrs": [
      {"intId": 1, "name": "trackEdgeSections", "composition": "TrackEdgeSection", "multiplicity": "1..*", "info": "composes
of track edge sections"},
      {"intId": 2, "name": "loadingGaugeType", "enumType": "LoadingGaugeType", "info": "defines different loafing
gauges"}
    ]
  }]
}
```

SPT2TS-125453 - TrackCondition

```
{
  "structs": [
  {
    "name": "TrackCondition",
    "info": "Defines the TrackConditions for a section of track",
    "attrs": [
      {"intId": 1, "name": "trackEdgeSections", "composition": "TrackEdgeSection", "multiplicity": "1..*", "info": "composes
of track edge sections"},
      {"intId": 2, "name": "conditionType", "enumType": "ConditionType", "info": "defines different track conditions"}
    ]
  }]
}
```

SPT2TS-125454 - CurrentLimitation

```
{
  "structs": [
  {
    "name": "CurrentLimitation",
    "info": "Defines max current for a list of track sections",
    "attrs": [
      {"intId": 1, "name": "trackEdgeSections", "composition": "TrackEdgeSection", "multiplicity": "1..*", "info": "composes
of track edge sections"},
      {"intId": 2, "name": "maxCurrent", "dataType": "uint32", "unit": "A", "info": "defines max current value"}
    ]
```

```
    }]
}
```

SPT2TS-125455 - PermittedBrakingDistance

```
{
  "structs": [
  {
    "name": "PermittedBrakingDistance",
    "info": "Defines max allowed braking distance for a list of track sections",
    "attrs": [
      {"intId": 1, "name": "trackEdgeSections", "composition": "TrackEdgeSection", "multiplicity": "1..*", "info": "composes
of track edge sections"},
      {"intId": 2, "name": "distance", "dataType": "uint32", "unit": "m", "info": "defines the applicable permitted braking
distance"},
      {"intId": 3, "name": "brakeType", "enumType": "BrakeType" , "info": "defines the applicable brake type for braking"},
      {"intId": 4, "name": "gradientValue", "dataType": "int32", "unit": "permill", "exp": -1, "info": "defines the gradient
associated with the permitted braking distance"},
      { "intId": 5, "name": "sameDir", "dataType": "boolean", "info": "usage direction of the TrackEdge ; true: active in same
direction as trackEdge, false: active in opposite direction"}
    ]
  }]
}
```

SPT2TS-125456 - TractionSystem

```
{
  "structs": [
  {
    "name": "TractionSystem",
    "info": "Defines the equipped traction system on the track",
    "attrs": [
      {"intId": 1, "name": "trackEdgeSections", "composition": "TrackEdgeSection", "multiplicity": "1..*", "info": "composes
of track edge sections"},
      {"intId": 2, "name": "voltageType", "enumType": "VoltageType", "info": "defines the different voltage types"}
    ]
  }]
}
```

2.7 Track usage area

TrackUsageArea
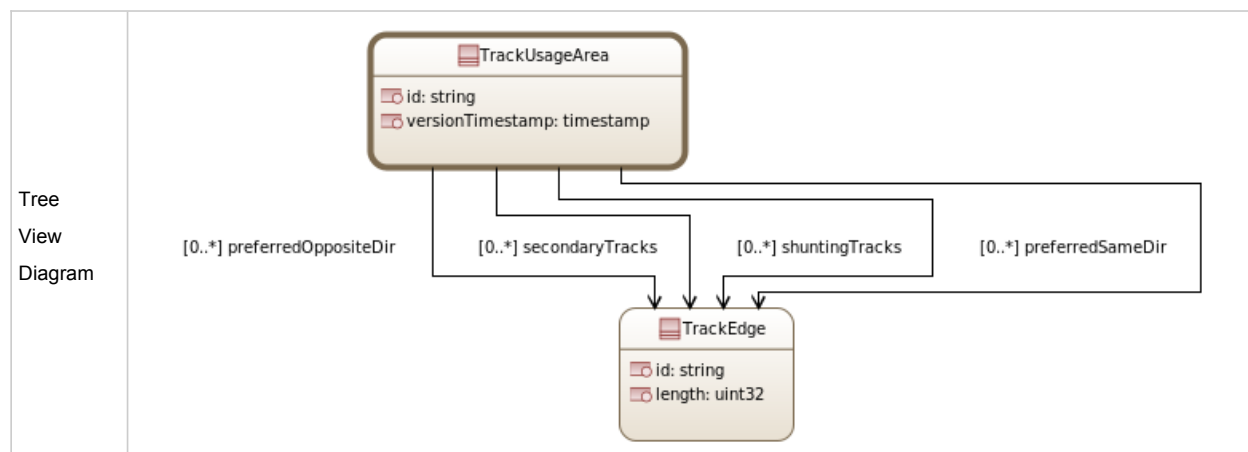
SPT2TS-125460 - TrackUsageArea

{
"structs": [
 {
   "name": "TrackUsageArea",
   "info": "Preferred usage specification, most usable for TMS and planning",
   "attrs": [
     {"intId": 1, "name": "id", "dataType": "string", "key": "global", "sameKeyAs": "TopoArea", "info": "Identity of the object; used for referencing"},
     {"intId": 2, "name": "versionTimestamp", "dataType": "timestamp","info": "version information which is valid since timestamp"},
     {"intId": 3, "name": "secondaryTracks", "reference": "TrackEdge", "multiplicity": "*", "sortedByKey": true, "info": "all tracks except 'shuntingTracks' and 'secondaryTracks' are 'mainTracks'"},
     {"intId": 4, "name": "shuntingTracks", "reference": "TrackEdge", "multiplicity": "*", "sortedByKey": true, "info": "all tracks except 'shuntingTracks' and 'secondaryTracks' are 'mainTracks'"},
     {"intId": 5, "name": "preferredSameDir", "reference": "TrackEdge", "multiplicity": "*", "sortedByKey": true, "info": "track edges with preferred direction identical to the TrackEdge-Direction"},
     {"intId": 6, "name": "preferredOppositeDir", "reference": "TrackEdge", "multiplicity": "*", "sortedByKey": true, "info": "track edges with preferred direction opposite to the TrackEdge-Direction"}
   ]
 }
]
}

2.8 Speed Profiles

SPT2TS-63860 - Speed profiles represent a base object that defines the speed restrictions for a section of track. Specific speed profiles are defined as an extension to this base object.

The following different kinds of speed restrictions exist:

1. The Static Speed Profile (SSP) is a description of the fixed speed restrictions of a given section of track. The speed restrictions can be related to, e.g. maximum line speed, curves, points, tunnel profiles, and bridges.

    1. The specific SSP categories, as defined in ETCS Subset 026, are decomposed into two types:

        - The "Cant Deficiency" SSP categories: the cant deficiency value assigned to one category shall define the maximum speed, determined by suspension design, at which a particular train can traverse a curve. It thus can be used to set a specific speed limit in a curve with regard to this category.
        - The "other specific" SSP categories: group all other specific SSP categories corresponding to the other international train categories.

2. The Axle load speed profiles define restrictive speed profiles for sections of track based on the different axle load categories defined in ETCS Subset 026.
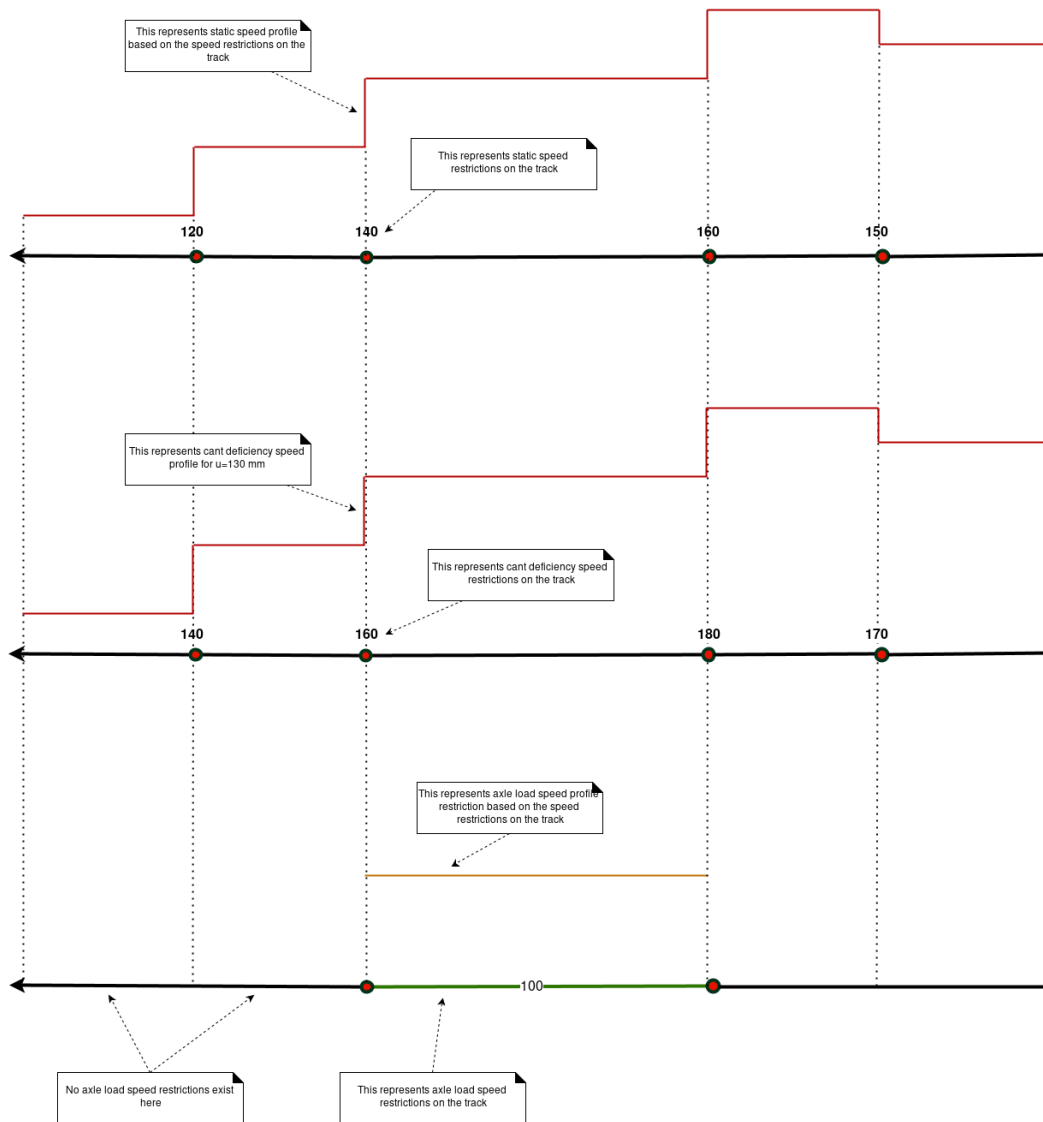
Figure 16 Illustration of different Speed Profiles

The following equivalent classes from other models can be referred to:

- Speed Profiles in SPT2TS-7 - RCA Digital Map – MAP Object Catalogue [RCA.Doc.69]

[Open]

TrackEdgeSpeedProfile
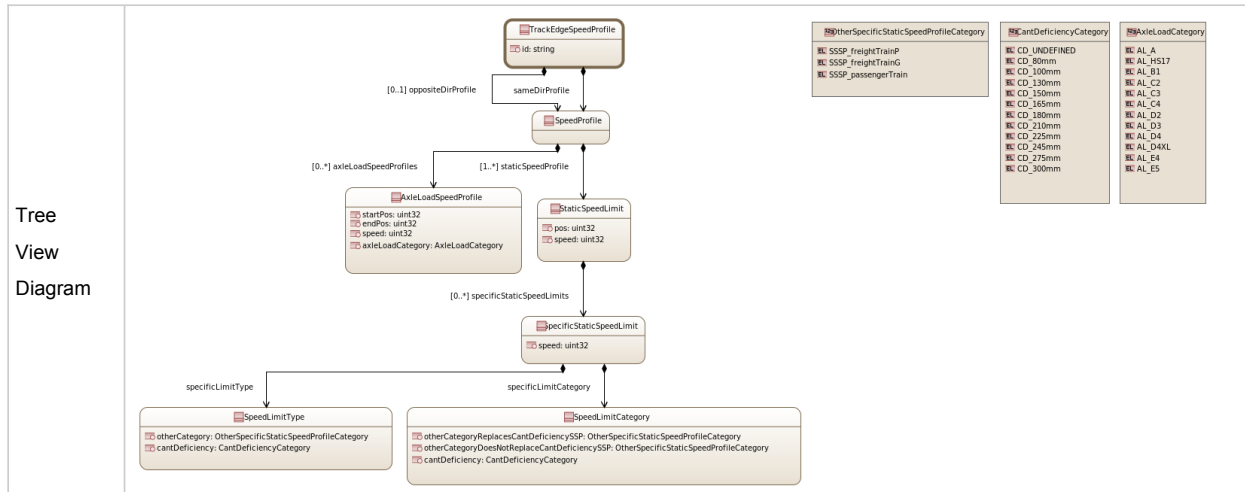
SPT2TS-49086 - TrackEdgeSpeedProfile

```
{
  "structs": [
  {
    "name": "TrackEdgeSpeedProfile",
    "info": "Defines the permitted speeds for a Track edge in both directions",
    "belongsToSubPackage": "PropertiesArea",
    "attrs": [
      { "intId": 1, "name": "id", "dataType": "string", "key": "global", "sameKeyAs": "TrackEdge", "info": "Identity of the object; used for referencing"},
      { "intId": 2, "name": "sameDirProfile", "composition": "SpeedProfile", "multiplicity": "1", "info": "speed profile valid in the same direction as the trackEdge"},
      { "intId": 3, "name": "oppositeDirProfile", "composition": "SpeedProfile", "multiplicity": "0..1", "info": "valid in the opposite direction as the TrackEdge. Skip if the same in both directions."}
    ]
  }]
}
```

SPT2TS-125463 - SpeedProfile

```
{
  "structs": [
  {
    "name": "SpeedProfile",
    "attrs": [
      { "intId": 1, "name": "staticSpeedProfile", "composition": "StaticSpeedLimit", "multiplicity": "1..*", "info": "composes of static speed profiles"},
      { "intId": 2, "name": "axleLoadSpeedProfiles", "composition": "AxleLoadSpeedProfile", "multiplicity": "0..*", "info": "composes of axle load speed profiles"}
    ]
  }]
}
```

SPT2TS-125464 - StaticSpeedLimit

```
{
  "structs": [
  {
    "name": "StaticSpeedLimit",
    "info": "Defines the static speed profile for a section of track",
    "belongsToSubPackage": "PropertiesArea",
    "attrs": [
      {"intId": 1, "name": "pos", "dataType": "uint32", "unit": "m", "exp": -3, "info": "position on the associated TrackEdge"},
      {"intId": 2, "name": "speed", "dataType": "uint32", "unit": "km/h", "info": "Defines the applicable speed for the section of track ; Resolution min 5 kmph"},
      {"intId": 3, "name": "specificStaticSpeedLimits", "composition": "SpecificStaticSpeedLimit", "multiplicity": "0..*", "info": "composes of specific static speed profiles"}
    ]
  }]
}
```

SPT2TS-125465 - SpecificStaticSpeedLimit

```
{
  "structs": [
  {
    "name": "SpecificStaticSpeedLimit",
    "info": "Defines the specific static speed profile for a section of track",
    "belongsToSubPackage": "PropertiesArea",
    "attrs": [
      {"intId": 1, "name": "speed", "dataType": "uint32", "unit": "km/h", "info": "Defines the applicable speed for the section of track ; Resolution min 5 kmph"},
      {"intId": 2, "name": "specificLimitCategory", "composition": "SpeedLimitCategory", "info": "defines different specific speed limit categories"}
    ]
  }]
}
```

SPT2TS-125461 - SpeedLimitCategory

```
{
  "structs": [
  {
    "name": "SpeedLimitCategory",
    "union": true,
    "attrs": [
      {"intId": 1, "name": "otherCategoryReplacesCantDeficiencySSP", "enumType": "OtherSpecificStaticSpeedProfileCategory", "info": "is of type other Category Replaces Cant Deficiency SSP"},
      {"intId": 2, "name": "otherCategoryDoesNotReplaceCantDeficiencySSP", "enumType": "OtherSpecificStaticSpeedProfileCategory", "info": "is of type other Category Replaces Cant Deficiency SSP"},
      {"intId": 3, "name": "cantDeficiencyCategory", "enumType": "CantDeficiencyCategory", "info": "is of type Cant Deficiency Category"}
    ]
```

```
  }]
 }
```

SPT2TS-125462 - AxleLoadSpeedProfile

```
{
  "structs": [
  {
    "name": "AxleLoadSpeedProfile",
    "info": "Defines the axle load speed profile for a section of track",
    "belongsToSubPackage": "PropertiesArea",
    "attrs": [
      {"intId": 1, "name": "startPos", "dataType": "uint32", "unit": "m", "exp": -3, "info": "start position on the associated
TrackEdge"},
      {"intId": 2, "name": "endPos", "dataType": "uint32", "unit": "m", "exp": -3, "info": "end position on the associated
TrackEdge"},
      {"intId": 3, "name": "speed", "dataType": "uint32", "unit": "km/h", "info": "Defines the applicable speed for the section
of track ; Resolution min 5 kmph"},
      {"intId": 4,"name": "axleLoadCategory", "enumType": "AxleLoadCategory", "info": "defines different axle load
categories"}
    ]
  }
]}
```

SPT2TS-126748 - SpeedProfile enumerations

```
{
"enums": [
{
    "name": "TrainEndApplicability",
    "info": "List of train end applicability types",
    "enumLiterals": [
      { "intId": 0, "name": "lengthDelay", "info": "Train length delay on validity end point of profile element"},
      { "intId": 1, "name": "noLengthDelay", "info": "No Train length delay on validity end point of profile element."}
    ]
  },
  {
    "name": "OtherSpecificStaticSpeedProfileCategory",
    "info": "List of other specific static speed profile categories ",
    "enumLiterals": [
      { "intId": 0, "name": "SSSP_freightTrainP", "info": "Other SSSP applies to Freight train of type P"},
      { "intId": 1, "name": "SSSP_freightTrainG", "info": "Other SSSP applies to Freight train of type G"},
      { "intId": 2, "name": "SSSP_passengerTrain", "info": "Other SSSP applies to Passenger train"}
    ]
  },
  {
    "name": "CantDeficiencyCategory",
    "info": "List of cant deficiency speed profile categories",
```

```
    "enumLiterals": [
        { "intId": 0, "name": "CD_Undefined", "info": "annotates inactive cant deficiency"},
        { "intId": 1, "name": "CD_80mm", "info": "Applicable for trains capable of compensating cant deficiency of 80mm"
},
        { "intId": 2, "name": "CD_100mm", "info": "Applicable for trains capable of compensating cant deficiency of
100mm"},
        { "intId": 3, "name": "CD_130mm", "info": "Applicable for trains capable of compensating cant deficiency of
130mm"},
        { "intId": 4, "name": "CD_150mm", "info": "Applicable for trains capable of compensating cant deficiency of
150mm"},
        { "intId": 5, "name": "CD_165mm", "info": "Applicable for trains capable of compensating cant deficiency of
165mm"},
        { "intId": 6, "name": "CD_180mm", "info": "Applicable for trains capable of compensating cant deficiency of
180mm"},
        { "intId": 7, "name": "CD_210mm", "info": "Applicable for trains capable of compensating cant deficiency of
210mm"},
        { "intId": 8, "name": "CD_225mm", "info": "Applicable for trains capable of compensating cant deficiency of
225mm"},
        { "intId": 9, "name": "CD_245mm", "info": "Applicable for trains capable of compensating cant deficiency of
245mm"},
        { "intId": 10, "name": "CD_275mm", "info": "Applicable for trains capable of compensating cant deficiency of
275mm"},
        { "intId": 11, "name": "CD_300mm", "info": "Applicable for trains capable of compensating cant deficiency of
300mm"}
    ]
  },
  {
    "name": "AxleLoadCategory",
    "info": "List of axle load speed profile categories",
    "enumLiterals": [
        { "intId": 0, "name": "AL_A", "info": "Applicable for trains falling under axle load category A"},
        { "intId": 1, "name": "AL_HS17", "info": "Applicable for trains falling under axle load category HS17"},
        { "intId": 2, "name": "AL_B1", "info": "Applicable for trains falling under axle load category B1"},
        { "intId": 3, "name": "AL_B2", "info": "Applicable for trains falling under axle load category B2"},
        { "intId": 4, "name": "AL_C2", "info": "Applicable for trains falling under axle load category C2"},
        { "intId": 5, "name": "AL_C3", "info": "Applicable for trains falling under axle load category C3"},
        { "intId": 6, "name": "AL_C4", "info": "Applicable for trains falling under axle load category C4"},
        { "intId": 7, "name": "AL_D2", "info": "Applicable for trains falling under axle load category D2"},
        { "intId": 8, "name": "AL_D3", "info": "Applicable for trains falling under axle load category D3"},
        { "intId": 9, "name": "AL_D4", "info": "Applicable for trains falling under axle load category D4"},
        { "intId": 10, "name": "AL_D4XL", "info": "Applicable for trains falling under axle load category D4XL"},
        { "intId": 11, "name": "AL_E4", "info": "Applicable for trains falling under axle load category E4"},
        { "intId": 12, "name": "AL_E5", "info": "Applicable for trains falling under axle load category E5"}
    ]
```

```
    }
  ]
}
```

2.8.1 Applicable direction

The speed profile in the same direction as the TrackEdge is assigned to TrackEdgeSpeedProfile.sameDirSpeedProfile. The speed profile in the opposite direction, if it is different from the "same direction" is assigned to TrackEdgeSpeedProfile.oppositeDirSpeedProfile. The SpeedProfile is a sampled speed curve, where the speed value defined in some position "pos" is valid in the "increasing" direction. This approach is used for both: same direction and opposite direction speed profiles. If you want to traverse the opposite direction from the End to Begin of the TrackEdge ensure proper adjustment of the speed-change-positions.
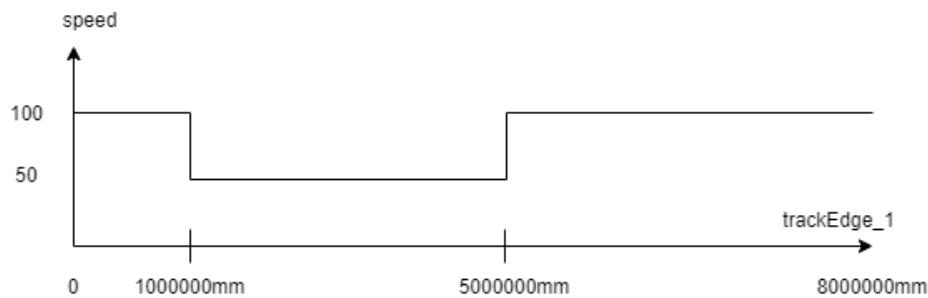

Figure 17 Speed profile in opposite direction.

The data object representing the speed profile for the opposite direction:

```
{
  "trackEdge": "trackEdge_1",
  "oppositeDirProfile": {
    "staticSpeedProfile": [
      {"pos": 0, "speed": 100},
      {"pos": 1000000, "speed": 50},
      {"pos": 5000000, "speed": 100}
    ]
  }
}
```

This means, that in the section [5000, 8000] the speed 100 is valid for trains moving in opposite direction of the trackEdge_1.
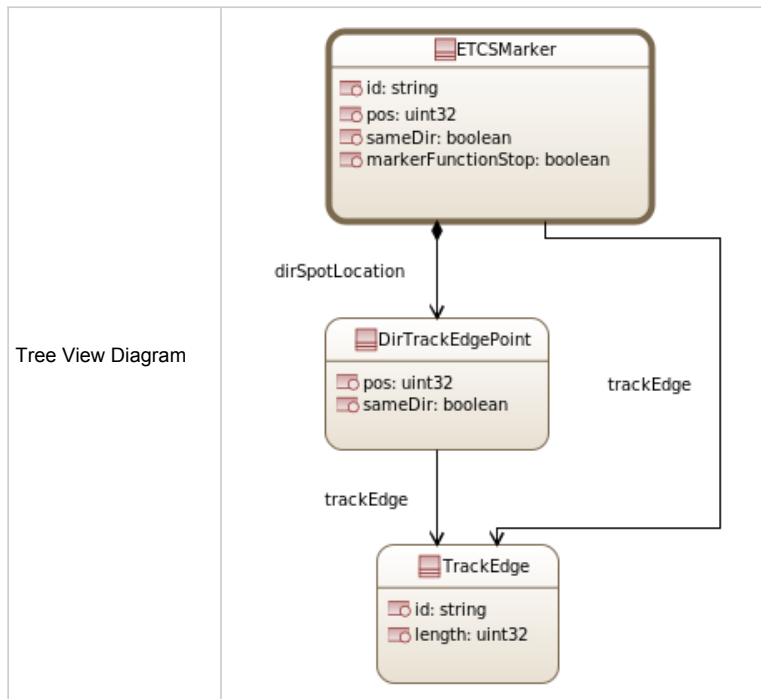
2.9 ETCS Marker

SPT2TS-64095 - Defines ETCS stop and location markers.

The following equivalent classes from other models can be referred to:

- Signal in ⊡ SPT2TS-7 - RCA Digital Map – MAP Object Catalogue [RCA.Doc.69]

[↪ Open]

ETCSMarker

Tree View Diagram	

SPT2TS-49088 - ETCSMarker

{

  "structs" : [

  {

   "name": "ETCSMarker",

   "info": "Defines the ETCS Marker Boards ",

   "belongsToSubPackage": "FunctionalArea",

   "attrs": [

        {"intId": 1, "name": "id", "dataType": "string", "key": "global", "info": "Identity of the object; used for referencing"},

        {"intId": 2, "name": "dirSpotLocation", "composition": "DirTrackEdgePoint"},

        {"intId": 3, "name": "markerFunctionStop", "dataType": "boolean", "info": "true=stop, false=location (when the ETCS marker is a location marker)"}
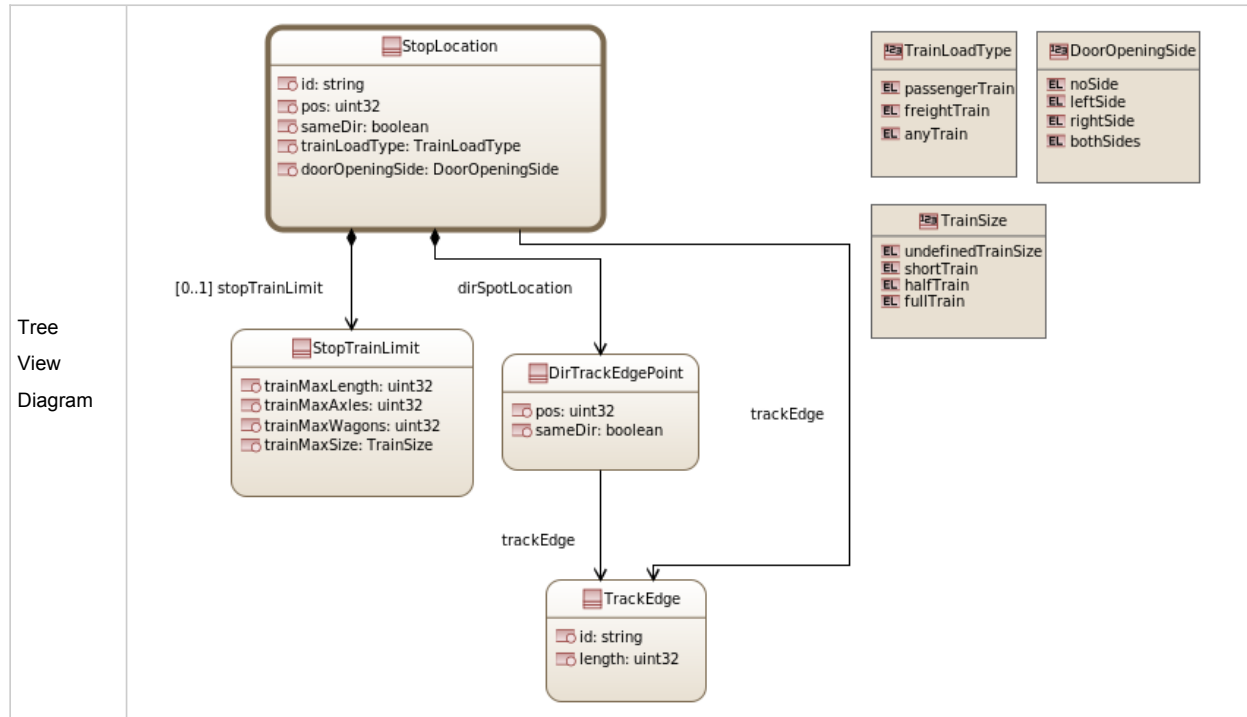
   ]

  }]

}

2.10 Stop Location

SPT2TS-64097 - Defines different stop locations of trains with respect to train size, wagons, axles, or length. The TMS is allowed to stop trains only at StopLocations.

The following equivalent classes from other models can be referred to:

- Signal in ⊞ SPT2TS-7 - RCA Digital Map – MAP Object Catalogue [RCA.Doc.69]

[↗ Open]

StopLocation

Tree View Diagram	

SPT2TS-49102 - StopLocation

```
{
  "structs": [
  {
    "name": "StopLocation",
    "info": "Defines the stop locations",
    "belongsToSubPackage": "FunctionalArea",
    "attrs": [
        {"intId": 1, "name": "id", "dataType": "string", "key": "global", "info": "Identity of the object; used for referencing"},
        {"intId": 2, "name": "dirSpotLocation", "composition": "DirTrackEdgePoint"},
        {"intId": 3, "name": "stopTrainLimit", "composition": "StopTrainLimit", "multiplicity": "0..1", "info": "composes of stop train limits"},
        {"intId": 4, "name": "trainLoadType", "enumType": "TrainLoadType", "info": "composes of train load types"},
        {"intId": 5, "name": "doorOpeningSide", "enumType": "DoorOpeningSide", "info": "defines different train opening sides"}
    ]
  }]
}
```

SPT2TS-125466 - StopTrainLimit

```
{
  "structs": [
  {
    "name": "StopTrainLimit",
    "attrs": [
      {"intId": 1, "name": "trainMaxLength", "dataType": "uint32", "unit": "m", "info": "use zero if not defined"},
      {"intId": 2, "name": "trainMaxAxles", "dataType": "uint32", "info": "use zero if not defined"},
```

```
      {"intId": 3, "name": "trainMaxWagons", "dataType": "uint32", "info": "including Locomotives, use zero if not defined"},
      {"intId": 4, "name": "trainMaxSize", "enumType": "TrainSize", "info": "use undefinedTrainSize if not defined"}
    ]
  }
]}
```

SPT2TS-126749 - StopLocation enumerations

```
{
"enums": [
  {
    "name": "TrainSize",
    "info":"List defining different train sizes",
    "enumLiterals": [
      {"intId": 0, "name": "undefinedTrainSize", "info": "train size undefined"},
      {"intId": 1, "name": "shortTrain", "info": "defines a short train"},
      {"intId": 2, "name": "halfTrain", "info": "defines a half train"},
      {"intId": 3, "name": "fullTrain", "info": "defines a full train"}
    ]
  },
  {
    "name": "TrainLoadType",
    "enumLiterals": [
      {"intId": 0, "name": "passengerTrain", "info": "defines a passenger train"},
      {"intId": 1, "name": "freightTrain", "info": "defines a freight train"},
      {"intId": 2, "name": "anyTrain", "info": "defines any train"}
    ]
  },
  {
    "name": "DoorOpeningSide",
    "enumLiterals": [
      {"intId": 0, "name": "noSide" , "info": "doors do not open in any side"},
      {"intId": 1, "name": "leftSide" , "info": "doors on the left side"},
      {"intId": 2, "name": "rightSide" , "info": "doors on the right side"},
      {"intId": 3, "name": "bothSides" , "info": "doors on both sides"}
    ]
  }]
}
```
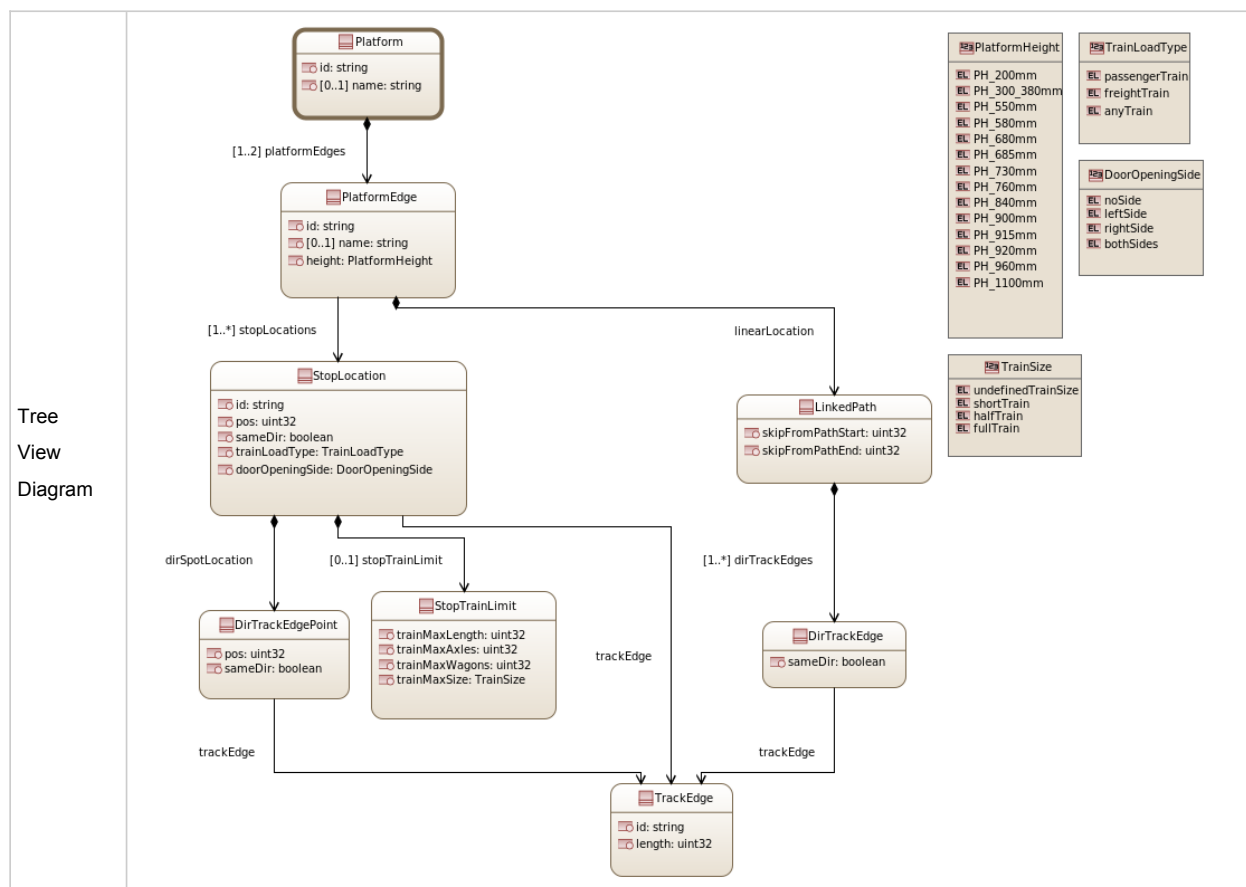
2.11 Platform

SPT2TS-64099 - Defines the civil infrastructure object platform

The following equivalent classes from other models can be referred to:

- Platform Edge in ⧉ SPT2TS-7 - RCA Digital Map – MAP Object Catalogue [RCA.Doc.69]
- Platform in ⧉ SPT2TS-9 - LinX4Rail CDM ->PSM (Platform Specific Data Model)

[↗ Open]

Platform

Tree View Diagram

SPT2TS-49092 - PlatformEdge

{
"structs": [
 {
   "name": "PlatformEdge",
    "info": "Platform for the purpose of RINF is understood as a platform edge. A platform concerns only the part of the structure neighbouring to the track (interfaced with trains).",
   "ontology": {"equivalentClass": "http://data.europa.eu/949/Platform"},
   "belongsToSubPackage": "FunctionalArea",
   "attrs": [
        {"intId": 1, "name": "platformId", "dataType": "string", "key": "global", "info": "Identity of the object; used for referencing"},

```
        {"intId": 2, "name": "name", "dataType": "string", "ontology": {"equivalentProperty": "http://data.europa.eu/949/platf
ormId"}, "info": "User-friendly name, only if different from id", "multiplicity": "0..1"},
        {"intId": 3, "name": "linearLocation", "composition": "LinkedPath", "info": "composes of linked path"},
        {"intId": 4, "name": "stopLocations", "reference": "StopLocation", "multiplicity": "1..*", "info": "add only, if
stopLocation can be used on that PlatformEdge for passenger/freight exchange", "sortedByKey": true},
        {"intId": 5, "name": "height", "enumType": "PlatformHeight", "ontology": {"equivalentProperty": "http://data.europa.e
u/949/platformHeight"}, "info": "defines different platform heights"}
    ]
  }]
}
```

SPT2TS-125467 - Platform

```
{
  "structs": [
  {
    "name": "Platform",
    "info":"Defines the container object platform",
    "belongsToSubPackage": "FunctionalArea",
    "attrs": [
        {"intId": 1, "name": "id", "dataType": "string", "key": "global", "info": "Identity of the object; used for referencing"},
        {"intId": 2, "name": "name", "dataType": "string","info":"User-friendly name, only if different from id", "multiplicity":
"0..1"},
        {"intId": 3, "name": "platformEdges", "composition": "PlatformEdge", "multiplicity": "1..*", "info": "composes of
platform edges"}
    ]
  }
]}
```

SPT2TS-126750 - Platform enumerations

```
{
"enums": [
  {
    "name": "PlatformHeight",
    "info": "List of nominal height of platform above rail level (refer to TSI infrastructure)",
    "enumLiterals": [
        { "intId": 0, "name": "PH_200mm", "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/platform-
heights/rinf/60"}, "info": "platform height 200mm"},
        { "intId": 1, "name": "PH_300_380mm", "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/platform-
heights/rinf/50"}, "info": "platform height 300-380mm"},
        { "intId": 2, "name": "PH_550mm", "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/platform-
heights/rinf/30"}, "info": "platform height 550mm"},
        { "intId": 3, "name": "PH_580mm", "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/platform-
heights/rinf/70"}, "info": "platform height 580mm"},
        { "intId": 4, "name": "PH_680mm", "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/platform-
heights/rinf/80"}, "info": "platform height 680mm"},
        { "intId": 5, "name": "PH_685mm", "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/platform-
```

heights/rinf/90"}, "info": "platform height 685mm"},

    { "intId": 6, "name": "PH_730mm", "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/platform-heights/rinf/100"}, "info": "platform height 730mm"},

    { "intId": 7, "name": "PH_760mm", "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/platform-heights/rinf/40"}, "info": "platform height 760mm"},

    { "intId": 8, "name": "PH_840mm", "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/platform-heights/rinf/110"}, "info": "platform height 840mm"},

    { "intId": 9, "name": "PH_900mm", "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/platform-heights/rinf/120"}, "info": "platform height 900mm"},

    { "intId": 10, "name": "PH_915mm", "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/platform-heights/rinf/130"}, "info": "platform height 915mm"},

    { "intId": 11, "name": "PH_920mm", "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/platform-heights/rinf/140"}, "info": "platform height 920mm"},

    { "intId": 12, "name": "PH_960mm", "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/platform-heights/rinf/150"}, "info": "platform height 960mm"},

    { "intId": 13, "name": "PH_1100mm", "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/platform-heights/rinf/160"}, "info": "platform height 1100mm"}

   ]

  }

]

}

SPT2TS-125512 - Example
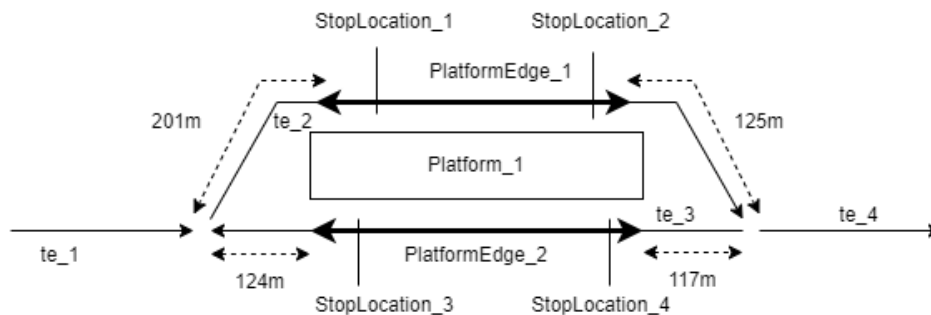
Example for the Platform-object:

{

 "id": "Platform_1",

 "name": "Berlin_Gate 1&2",

 "platformEdges": [

  {

   "id": "PlatformEdge_1",

   "name": "Gate 1",

   "linearLocation": {"dirTrackEdges": [{"trackEdge": "te_2", "sameDir": false}], "skipFromPathStart": 125, "skipFromPathEnd": 201},

   "stopLocations": ["StopLocation_1", "StopLocation_2"],

   "height": "PH_960mm"

  },

```
  {
    "id": "PlatformEdge_2",
    "name": "Gate 2",
    "linearLocation": {"dirTrackEdges": [{"trackEdge": "te_3", "sameDir": true}], "skipFromPathStart": 117,
"skipFromPathEnd": 124},
    "stopLocations": ["StopLocation_3", "StopLocation_4"],
    "height": "PH_960mm"
  }
 ]
}
```
[⊙⚲Content to be approved **]**

2.12 Operational Point

SPT2TS-63863 - An operational point is an area of the railway network, aggregated into a geographical point (with geographical coordinates), that has a distinctive operational function.

Note:

1. Although an Operational Point is illustrated as an area, it is merely just a set of characteristics belonging to a particular location.
2. The centre point is not always located in the centre of the operational point, and corresponding tolerances can be foreseen.

Use Case: Systems which are responsible for planning railway operations might require an overview on this level to determine required points (specific to an operational point) on the topology for timetable planning, journey profile definition, movement permission definition, etc.

The illustration below depicts an Operational Point with its operational characteristics. The definition of Operational Point might not be apt, and it is only defined for pictorial and understanding purposes.
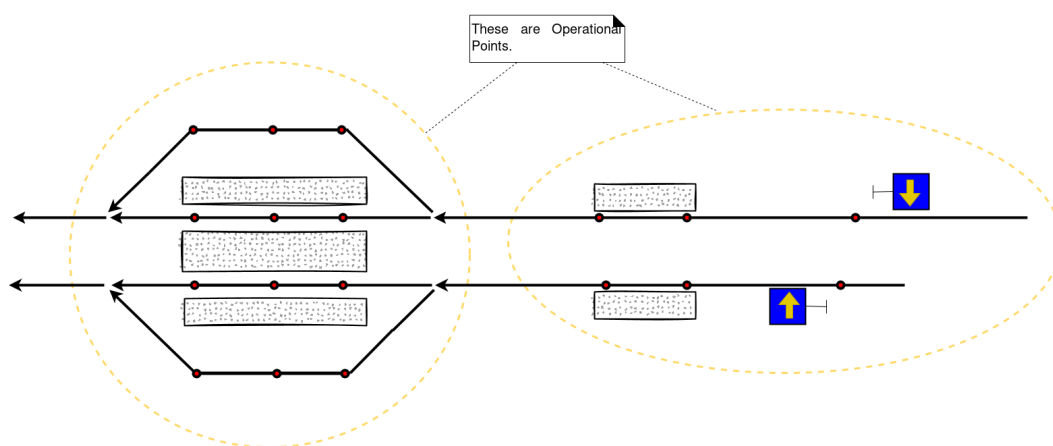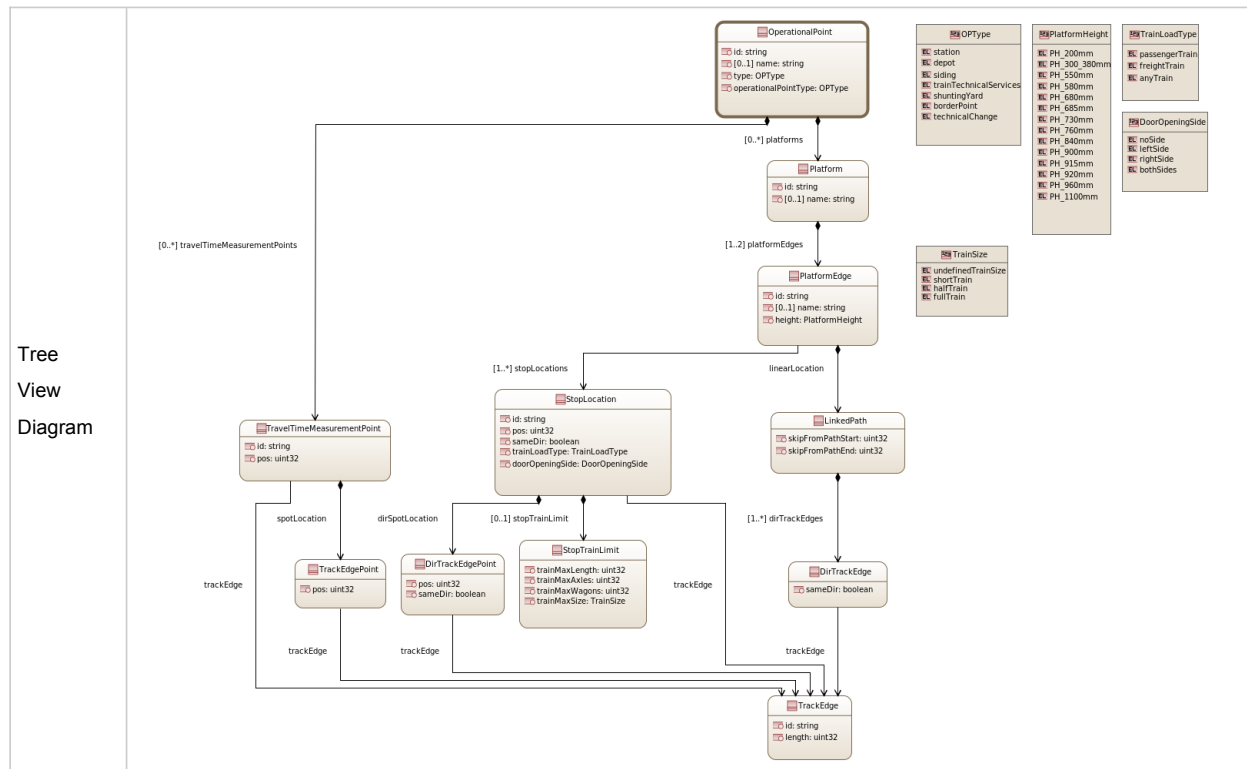

Figure 18 Explanation for Operational Points

The following equivalent classes from other models can be referred to:

- Operational Point in ⊞ SPT2TS-7 - RCA Digital Map – MAP Object Catalogue [RCA.Doc.69]
- Operational Point in ⊞ SPT2TS-64108 - Guide on the application of the common specifications of the register of Infrast...

[➦ Open **]**

OperationalPoint

SPT2TS-49105 - OperationalPoint

{

  "structs": [

  {

    "name": "OperationalPoint",

    "info": "An operational point (OP) means any location for train service operations, where train services may begin and end or change route, and where passenger or freight services may be provided; operational point also means any location at boundaries between Member States or infrastructure managers. In https://eur-lex.europa.eu/eli/reg_impl/2019/773/oj 2.1.2 Principal locations (stations, yards, junctions, freight terminals).",

    "ontology": {"subClassOf": "http://data.europa.eu/949/OperationalPoint"},

    "belongsToSubPackage": "FunctionalArea",

    "attrs":[

      {"intId": 1, "name": "uopId", "dataType": "string", "key": "global",

       "ontology": {"subPropertyOf": "http://data.europa.eu/949/uopid"}, "info": "Identity of the object; used for referencing"
},

      {"intId": 2, "name": "opName", "dataType": "string", "info": "User-friendly name, only if different from id", "multiplicity": "0..1",

       "ontology": {"subPropertyOf": "http://data.europa.eu/949/opName"}},

      {"intId": 3, "name": "travelTimeMeasurementPoints", "composition": "TravelTimeMeasurementPoint", "multiplicity":"0..*", "sortedByKey": true, "info": "used to measure/define timestamps for passing trains. Stopping-timingPoints can be found via platforms.platformEdges.stopLocations and timingPoint.stopLocation"},

      {"intId": 4, "name": "operationalPointType", "enumType": "OPType", "info": "Defines the type of Operational Point",

       "ontology": {"subPropertyOf": "http://data.europa.eu/949/opType"}},

      {"intId": 5, "name": "platforms", "composition": "Platform", "multiplicity":"0..*", "sortedByKey": false, "info": "the

sequence of platforms is according to the real location, so neighbours are the same in array and reality"}

    ]

}

]}

SPT2TS-126751 - OperationalPoint enumerations

```
{
"enums": [
    {
        "name": "OPType",
        "info": "List of types for Operational Point",
        "enumLiterals": [
            { "intId": 0, "name": "station", "info": "station with functions, supports international  and/or national railway traffic",
              "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/op-types/rinf/10"}, "info": "is of type station"},
            { "intId": 1, "name": "depot", "info": "group of tracks used by depot or workshop for maintenance",
              "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/op-types/rinf/50"}, "info": "is of type depot"},
            { "intId": 2, "name": "siding", "info": "operational point that describes the embranchment located on the main line
that leads to the siding",
               "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/op-types/rinf/130"}, "info": "is of type siding"},
            { "intId": 3, "name": "trainTechnicalServices", "info": "group of tracks for servicing trains (parking, washing, etc.)",
               "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/op-types/rinf/60"}, "info": "is of type train
technical services"},
            { "intId": 4, "name": "shuntingYard", "info": "group of tracks used for shunting trains, mostly related to freight
traffic",
               "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/op-types/rinf/100"}, "info": "is of type shunting
yard"},
            { "intId": 5, "name": "borderPoint", "info": "located in the point where a border between Major States or IMs meets
a railway line",
               "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/op-types/rinf/90"}, "info": "is of type border point"
},
            { "intId": 6, "name": "technicalChange", "info": "to describe a change of CCS or a type of contact line or a Gauge
changeover facility",
               "ontology": {"owl:sameAs": "http://data.europa.eu/949/concepts/op-types/rinf/110"}, "info": "is of type technical
change"}
        ]
    }
]
}
```

SPT2TS-125468 - TravelTimeMeasurementPoint

```
{
  "structs": [
  {
   "name": "TravelTimeMeasurementPoint",
   "info": "normally the passing time through a OperationalPoint is measured at the middle of it",
   "attrs": [
```

```
    {"intId": 1, "name": "id", "dataType": "string", "key": "local", "info": "Identity of the object; used for referencing"},
    {"intId": 2, "name": "spotLocation", "composition": "TrackEdgePoint"}
  ]
}]
}
```

2.13 Timing Point

SPT2TS-63864 - A Timing Point is an ATO-aspect. It is a fixed point on the infrastructure which is associated with the appropriate time. This time may be an arrival time, departure time, or a passing time.

The fixed points on the infrastructure can be further elaborated as follows:

- All the ETCS signals where the routes may end or begin and at signals covering danger areas, e.g., tunnels or buffer stops.
- At platforms, e.g., In the center and both ends of the platform
- At stop posts, e.g., based on train lengths (200m, 300m, …), train size (short, medium, full), Number of Wagons, Number of Axles

In addition, the Timing Points have stopping accuracy information i.e., tolerances which enable trains to come to an accurate stop at Timing Points. To ensure precise stopping, additional requirements on the engineering of balises before Timing Points of relevance are also foreseen.

Use Case: For ATO, Timing Points are referenced in Segment Profiles and Journey Profiles. The Segment Profiles provides the location of the Timing Point and the Journey Profiles provide the time and operational aspects (Stop/Pass/Skip) of the Timing Point. The illustration below depicts a Timing Point with respect to Operational Points.
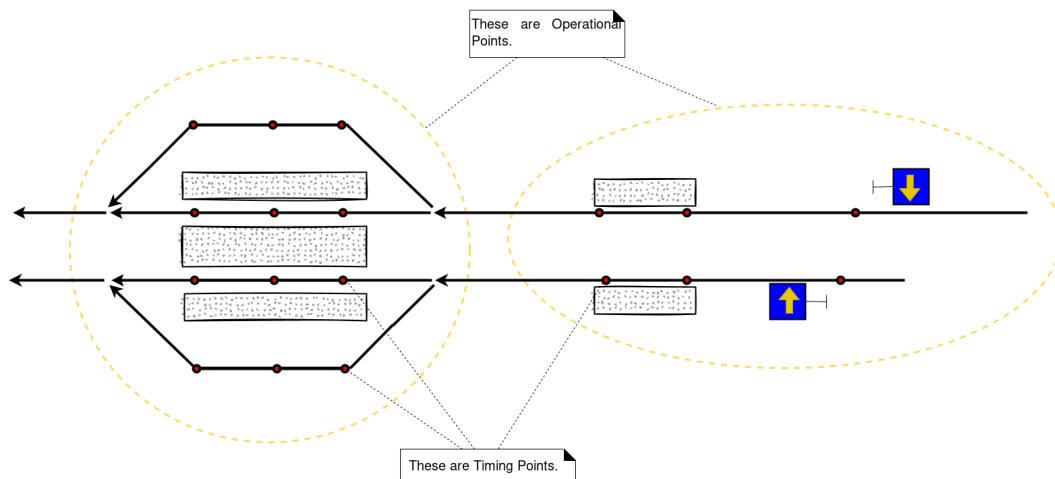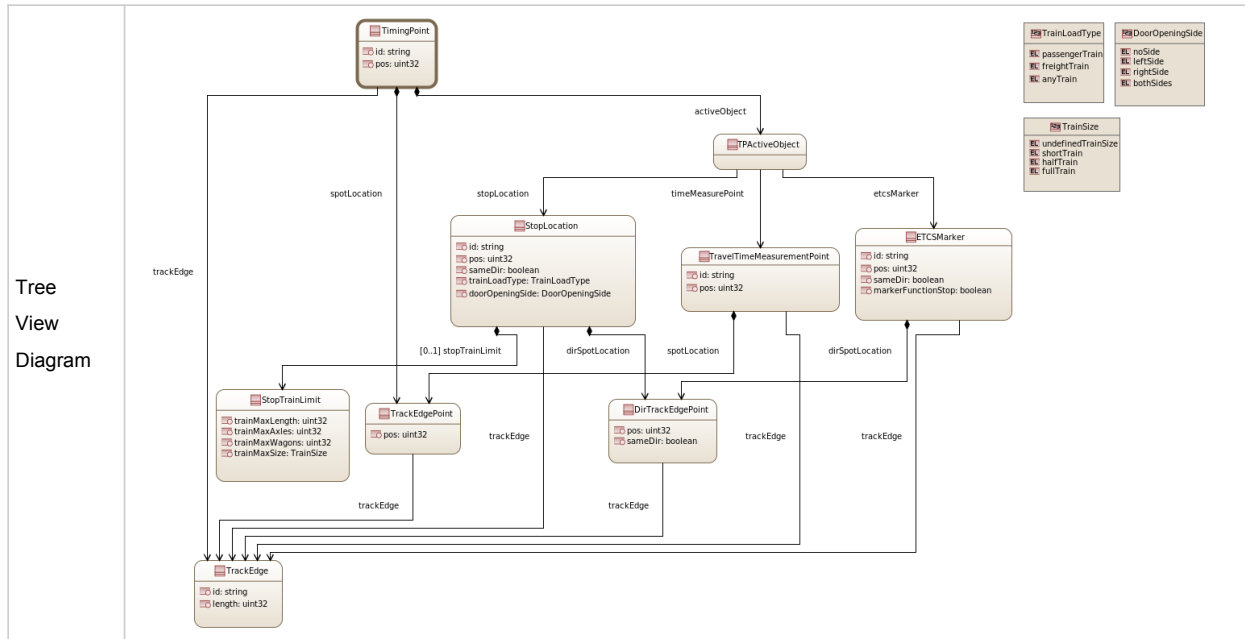

Figure 19 Explanation for Operational Points and Timing Points.

The following equivalent classes from other models can be referred to:

- Timing Point in SPT2TS-7 - RCA Digital Map – MAP Object Catalogue [RCA.Doc.69]
- Timing Point in SPT2TS-9 - LinX4Rail CDM ->PSM (Platform Specific Data Model)

[Open]

TimingPoint

SPT2TS-125469 - TimingPoint

```
{
  "structs": [
  {
   "name": "TimingPoint",
   "info": "Defines a fixed point on the infrastructure",
   "belongsToSubPackage": "FunctionalArea",
   "attrs":
   [
      {"intId": 1, "name": "id", "dataType": "string", "key": "global", "info": "Identity of the object; used for referencing"},
      {"intId": 2, "name": "spotLocation", "composition": "TrackEdgePoint"},
      {"intId": 3, "name": "activeObject", "composition": "TPActiveObject", "info": "composes of active timing points
object"}
   ]
},
{
  "name": "TPActiveObject",
  "info": "A functional object infront of which a TimingPoint is located and active",
  "union": true,
  "attrs": [
   {"intId": 1, "name": "etcsMarker", "reference": "ETCSMarker", "info": "refers to ETCS marker"},
   {"intId": 2, "name": "stopLocation", "reference": "StopLocation", "info": "refers to stop location"},
   {"intId": 3, "name": "timeMeasurePoint", "reference": "TravelTimeMeasurementPoint", "info": "refers to travel time
measurement point"}
  ]
}
]
}
```

2.14 Tunnel

SPT2TS-64100 - The object tunnel provides a physical representation of the tunnel infrastructure. A tunnel consists Tunnel Tube(s) and Tunnel Portal(s)

The following equivalent classes from other models can be referred to:

- Tunnel in ⬛ SPT2TS-7 - RCA Digital Map – MAP Object Catalogue [RCA.Doc.69]
- Tunnel in ⬛ SPT2TS-9 - LinX4Rail CDM ->PSM (Platform Specific Data Model)

[🔄 Open]

Tunnel
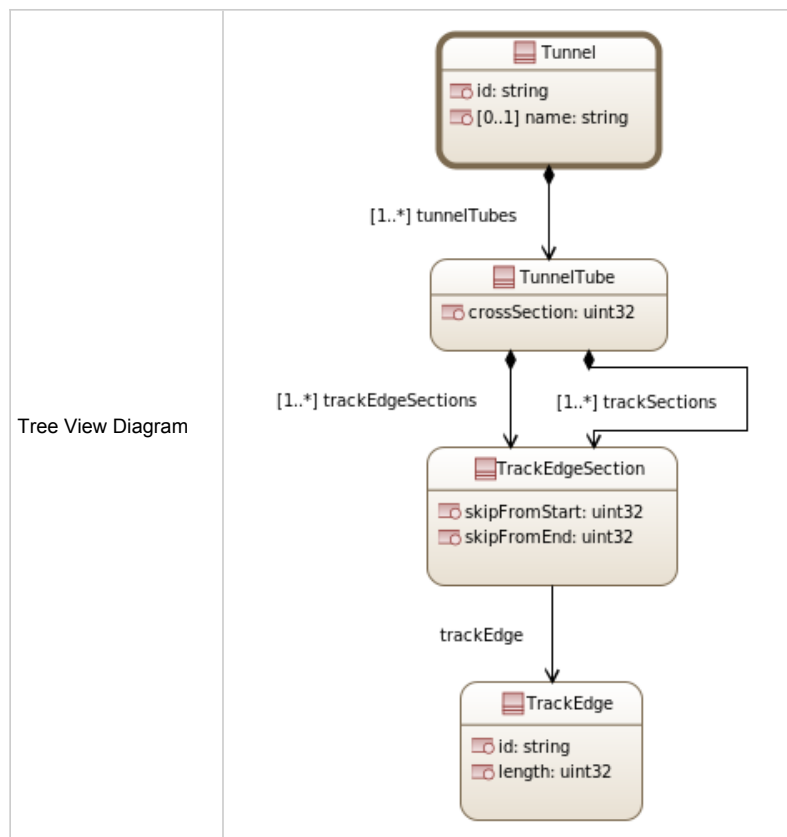
SPT2TS-48967 - Tunnel

{
"structs" : [
{
   "name": "Tunnel",
   "info": "A railway tunnel is an excavation or a construction around the track provided to allow the railway to pass for example higher land, buildings or water.",
    "ontology": {"equivalentClass": "http://data.europa.eu/949/Tunnel"},
   "belongsToSubPackage": "FunctionalArea",
   "attrs": [
        {"intId": 1, "name": "tunnelIdentification", "dataType": "string", "key": "global", "info": "Identity of the object; used for referencing"},
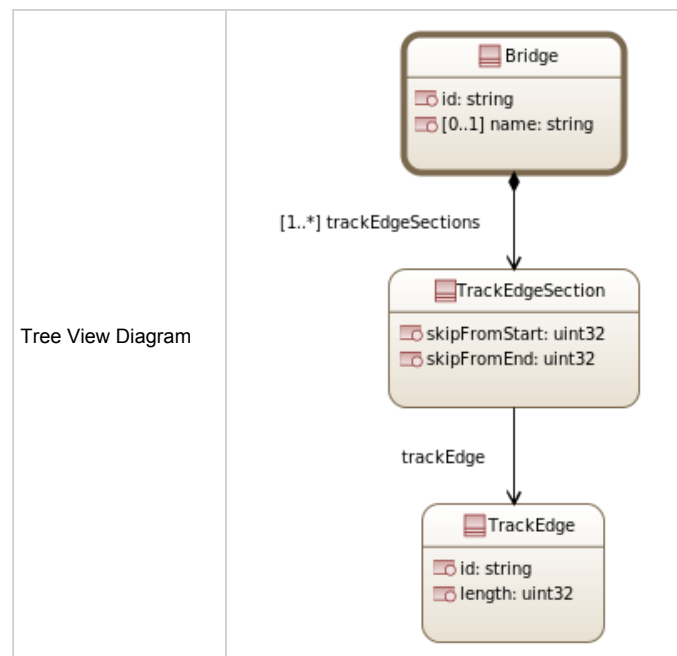
{"intId": 2, "name": "name", "dataType": "string","info": "User-friendly name, only if different from id", "multiplicity": "0..1"},

{"intId": 3, "name": "tunnelTubes", "composition": "TunnelTube", "multiplicity":"1..*", "info": "composes of tunnel tubes"}

    ]

 },

{

  "name": "TunnelTube",

  "info": "Defines the tunnels tubes within a tunnel. Main Use-Case identify emergency management with rescue into neighbour tube",

  "belongsToSubPackage": "FunctionalArea",

  "attrs": [

    {"intId": 1, "name": "trackArea", "composition": "TrackArea", "info": "composes of track area"},

    {"intId": 2, "name": "crossSection", "dataType": "uint32", "unit": "m2", "exp": -3, "info": "for calculation of air-resistance ; Standard Cross section dimensions corresponding to ATO Tunnel types will be defined later"}

    ]

 }

]

}

2.15 Bridge

SPT2TS-64102 - The object bridge provides a physical representation of the bridge infrastructure. **[Open]**

Bridge

Tree View Diagram

SPT2TS-48975 - Bridge

{

"structs": [
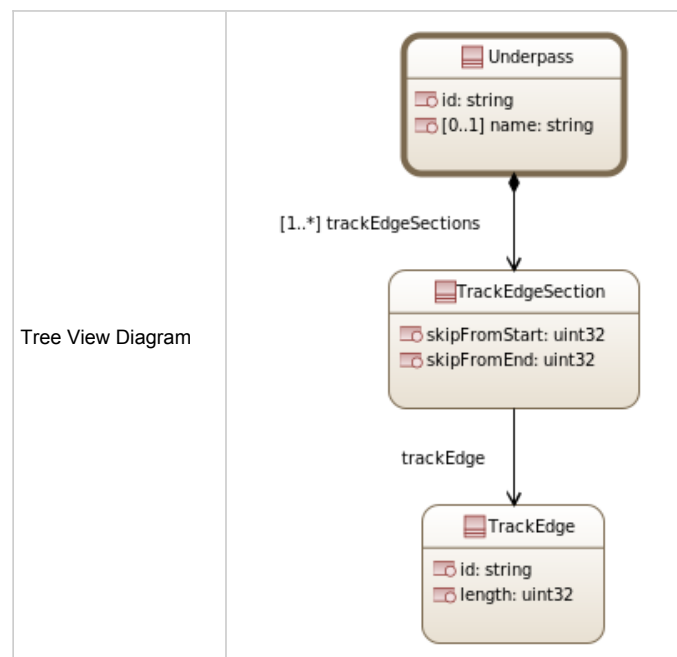
{

```
    "name": "Bridge",
    "info": "Defines the civil infrastructure object bridge",
    "belongsToSubPackage": "FunctionalArea",
    "attrs": [
        {"intId": 1, "name": "id", "dataType": "string", "key": "global", "info": "Identity of the object; used for referencing"},
        {"intId": 2, "name": "name", "dataType": "string","info": "User-friendly name, only if different from id", "multiplicity":
"0..1"},
        {"intId": 3, "name": "trackArea", "composition": "TrackArea", "info": "composes of track area"}
    ]
  }
]
}
```

2.16 Underpass

SPT2TS-64103 - The object underpass provides a physical representation of the underpass infrastructure. [↪ Open]

Underpass

Tree View Diagram

SPT2TS-48976 - Underpass

```
{
"structs": [
{
    "name": "Underpass",
    "info": "Defines the civil infrastructure object Underpass",
    "belongsToSubPackage": "FunctionalArea",
    "attrs": [
        {"intId": 1, "name": "id", "dataType": "string", "key": "global", "info": "Identity of the object; used for referencing"},
        {"intId": 2, "name": "name", "dataType": "string","info": "User-friendly name, only if different from id", "multiplicity":
"0..1"},
```

        {"intId": 3, "name": "trackArea", "composition": "TrackArea", "info": "composes of track area"}

      ]

    }

  ]

}

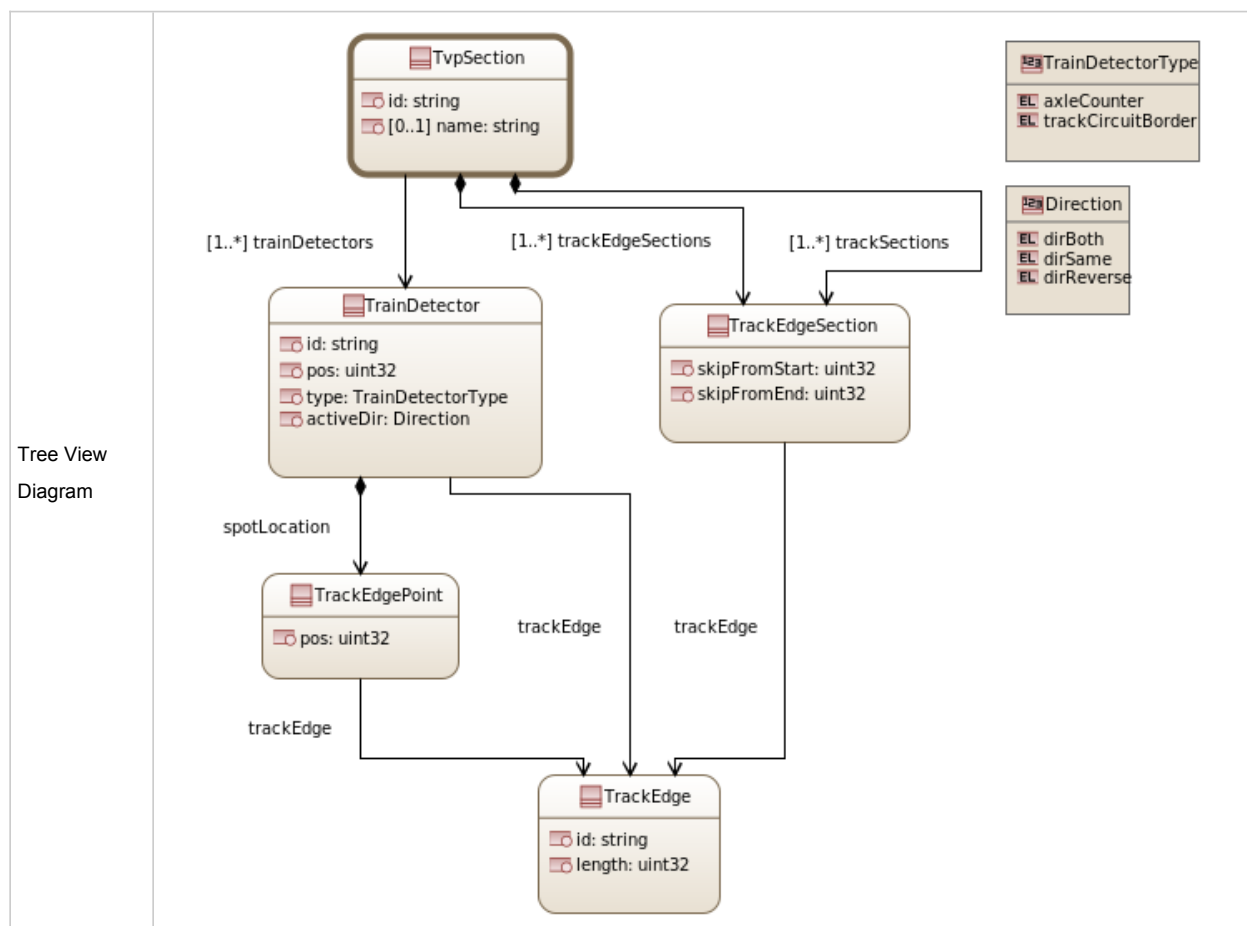### 2.17 Track Vacancy Proving Section (TVPS)

SPT2TS-64104 - Defines a section along the track for which the occupancy by a movable object can be technically detected. Each track vacancy proving section has a train detectors that are used to detect movable objects on the track.

The following equivalent classes from other models can be referred to:

- Trackside Train Detection Area in SPT2TS-7 - RCA Digital Map – MAP Object Catalogue [RCA.Doc.69]
- TVP Section in SPT2TS-9 - LinX4Rail CDM ->PSM (Platform Specific Data Model)
- TVP Section in https://eulynx.eu/index.php/dataprep

[Open]

TvpSection

Tree View
Diagram

SPT2TS-12698 - TvpSection

{

  "structs": [

  {

```
    "name": "TvpSection",
    "info": "Defines the train vacancy proving section",
    "belongsToSubPackage": "FunctionalArea",
    "attrs": [
      {"intId": 1, "name": "id", "dataType": "string", "key": "global", "info": "Identity of the object; used for referencing ; struct
ure refers to EULYNX EU.SAS.77"},
      {"intId": 2, "name": "name", "dataType": "string","info": "User-friendly name, only if different from id", "multiplicity":
"0..1"},
      {"intId": 3, "name": "linkedArea", "composition": "LinkedArea", "info": "composes of linked area"},
      {"intId": 4, "name": "trainDetectors", "reference": "TrainDetector", "multiplicity": "0..*", "info": "refers to train detectors"}
    ]
  }]
}
```

SPT2TS-125470 - TrainDetector

```
{
  "structs": [
  {
    "name": "TrainDetector",
    "info": "Defines the train detector system for the train vacancy proving section",
    "belongsToSubPackage": "FunctionalArea",
    "attrs": [
      {"intId": 1, "name": "id", "dataType": "string", "key": "global", "info": "Identity of the object; used for referencing"},
      {"intId": 2, "name": "spotLocation", "composition": "TrackEdgePoint"},
      {"intId": 3, "name": "trainDetectorType", "enumType": "TrainDetectorType", "info": "Defines the train detection system
applicable for the TVPS"},
      {"intId": 4, "name": "countUpDir", "enumType": "Direction", "info": "Direction in which axle counter counts up"}
    ]
  }],
"enums": [
{
    "name": "TrainDetectorType",
    "info":"List defining different train detection systems",
    "enumLiterals": [
      {"intId": 0, "name": "axleCounter", "info": "defines an axle counter"},
      {"intId": 1, "name": "trackCircuitBorder" , "info": "defines a track circuit border"}
    ]
}
]
}
```
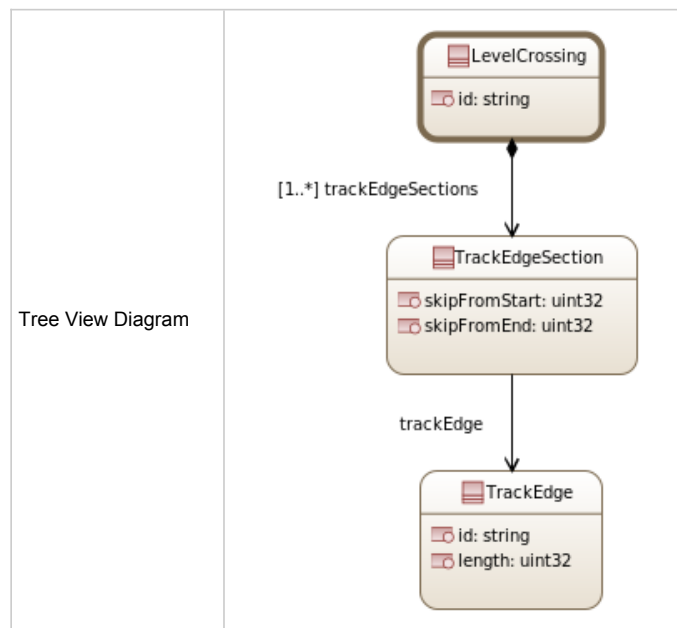
2.18 Level Crossing

SPT2TS-64101 - Defines the technical object level crossing along the track

The following equivalent classes from other models can be referred to:

- Level Crossing in ▣ SPT2TS-7 - RCA Digital Map – MAP Object Catalogue [RCA.Doc.69]
- Level Crossing in ▣ SPT2TS-9 - LinX4Rail CDM ->PSM (Platform Specific Data Model)

[➜ Open]

LevelCrossing

Tree View Diagram	

SPT2TS-12699 - LevelCrossing

```
{
  "structs": [
{
    "name": "LevelCrossing",
    "info": "Defines the track asset level crossing",
    "belongsToSubPackage": "FunctionalArea",
    "attrs": [
      {"intId": 1, "name": "id", "dataType": "string", "key": "global", "info": "Identity of the object; used for referencing"},
      {"intId": 2, "name": "trackArea", "composition": "TrackArea", "info": "composes of track area"}
    ]
  }
]
}
```
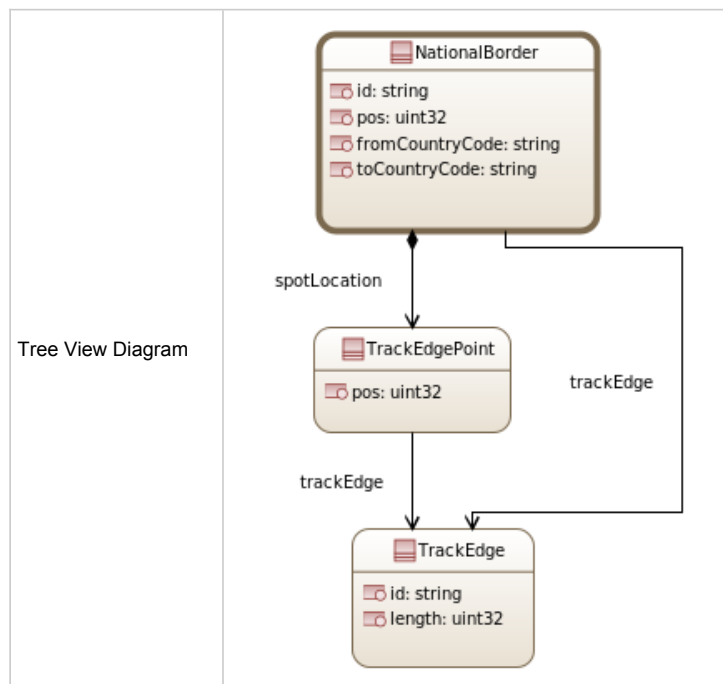
2.19 National border

SPT2TS-64105 - Defines an object National Border between countries.

The following equivalent classes from other models can be referred to:

- Track Node in ⊞ SPT2TS-7 - RCA Digital Map – MAP Object Catalogue [RCA.Doc.69]
- Border in ⊞ SPT2TS-9 - LinX4Rail CDM ->PSM (Platform Specific Data Model)

[↗ Open]

NationalBorder

Tree View Diagram

SPT2TS-48983 - NationalBorder

```
{
  "structs": [
  {
    "name": "NationalBorder",
    "info": "A border point is used to separate the railway network due to different reasons. Typical examples are country borders, the change of owning infrastructure manager or the border of a station",
    "belongsToSubPackage": "FunctionalArea",
    "attrs": [
      {"intId": 1, "name": "id", "dataType": "string", "key": "global", "info": "Identity of the object; used for referencing"},
      {"intId": 2, "name": "spotLocation", "composition": "TrackEdgePoint"},
      {"intId": 3, "name": "fromCountryCode", "dataType": "string", "info": "starting part of the TrackEdge"},
      {"intId": 4, "name": "toCountryCode", "dataType": "string", "info": "ending part of the TrackEdge"}
    ]
  }
]}
```
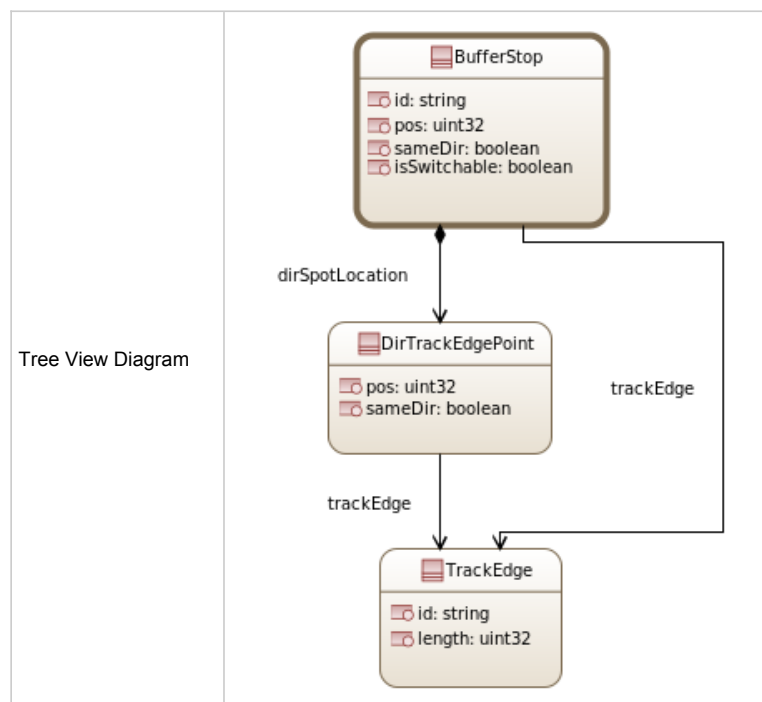
2.20 Buffer Stop

SPT2TS-100571 - Defines an object buffer stop on the track

The following equivalent classes from other models can be referred to:

- Track Node in ⊞ SPT2TS-7 - RCA Digital Map – MAP Object Catalogue [RCA.Doc.69]
- Buffer Stop in ⊞ SPT2TS-9 - LinX4Rail CDM ->PSM (Platform Specific Data Model)

[🔗 Open]

BufferStop


SPT2TS-100570 - BufferStop

```
{
 "structs": [
 {
  "name": "BufferStop",
  "info": "Defines the object Buffer stop on the track",
  "belongsToSubPackage": "FunctionalArea",
  "attrs": [
     {"intId": 1, "name": "id", "dataType": "string", "key": "global", "info": "Identity of the object; used for referencing"},
     {"intId": 2, "name": "dirSpotLocation", "composition": "DirTrackEdgePoint"},
     {"intId": 3, "name":"isSwitchable", "dataType":"boolean", "info":"true - if switchable ; false - if not switchable"}
  ]
 }
]}
```

2.21 Container for functional objects

**FunctionalArea**
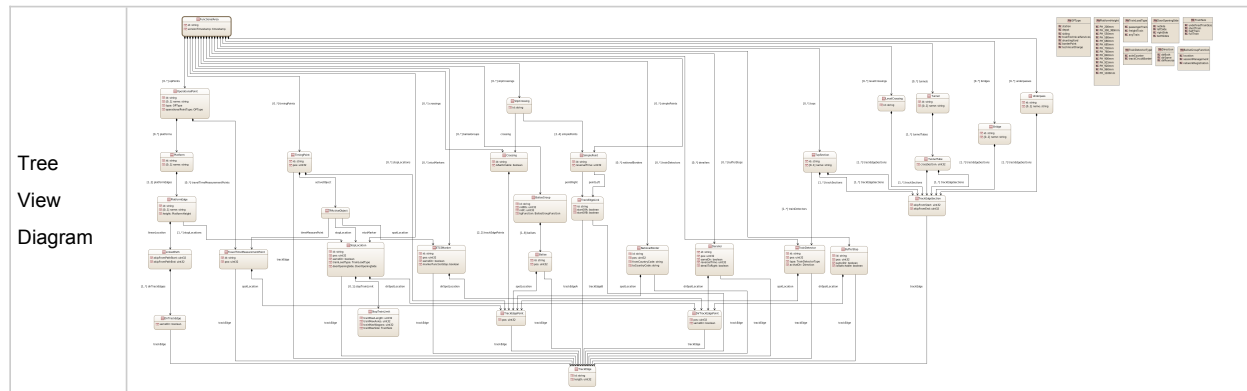
SPT2TS-63853 - FunctionalArea

```
{
  "structs": [
    {
      "name": "FunctionalArea",
      "attrs": [
        {"intId": 1, "name": "id", "dataType": "string", "key": "global", "sameKeyAs": "TopoArea", "info": "Identity of the object; used for referencing"},
        {"intId": 2, "name": "versionTimestamp", "dataType": "timestamp", "info": "version information which is valid since timestamp"},
        {"intId": 3, "name": "opPoints", "composition": "OperationalPoint", "multiplicity": "*", "sortedByKey": true},
        {"intId": 4, "name": "simplePoints", "composition": "SimplePoint", "multiplicity": "*", "sortedByKey": true},
        {"intId": 5, "name": "slipCrossings", "composition": "SlipCrossing", "multiplicity": "*", "sortedByKey": true},
        {"intId": 6, "name": "derailers", "composition": "Derailer", "multiplicity": "*", "sortedByKey": true},
        {"intId": 7, "name": "crossings", "composition": "Crossing", "multiplicity": "*", "sortedByKey": true},
        {"intId": 8, "name": "trainDetectors", "composition": "TrainDetector", "multiplicity": "*", "sortedByKey": true},
        {"intId": 9, "name": "baliseGroups", "composition": "BaliseGroup", "multiplicity": "*", "sortedByKey": true},
        {"intId": 10, "name": "etcsMarkers", "composition": "ETCSMarker", "multiplicity": "*", "sortedByKey": true},
        {"intId": 11, "name": "stopLocations", "composition": "StopLocation", "multiplicity": "*", "sortedByKey": true},
        {"intId": 12, "name": "tunnels", "composition": "Tunnel", "multiplicity": "*", "sortedByKey": true},
        {"intId": 13, "name": "bridges", "composition": "Bridge", "multiplicity": "*", "sortedByKey": true},
        {"intId": 14, "name": "underpasses", "composition": "Underpass", "multiplicity": "*", "sortedByKey": true},
        {"intId": 15, "name": "tvps", "composition": "TvpSection", "multiplicity": "*", "sortedByKey": true},
        {"intId": 16, "name": "levelCrossings", "composition": "LevelCrossing", "multiplicity": "*", "sortedByKey": true},
        {"intId": 17, "name": "nationalBorders", "composition": "NationalBorder", "multiplicity": "*", "sortedByKey": true},
        {"intId": 18, "name": "bufferStops", "composition": "BufferStop", "multiplicity": "*", "sortedByKey": true},
        {"intId": 19, "name": "timingPoints", "composition": "TimingPoint", "multiplicity": "*", "sortedByKey": true}
      ]
    }
  ]
}
```

2.22 Container for Infrastructure

Infrastructure

Tree

View

Diagram
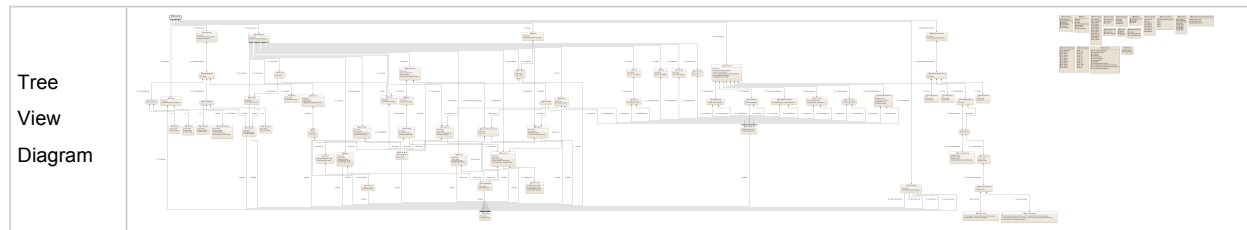
SPT2TS-64106 - Infrastructure

```
{
  "structs": [
    {
      "name": "Infrastructure",
      "attrs": [
        {"intId": 1, "name": "topoAreas", "composition": "TopoArea", "multiplicity": "*", "sortedByKey": true},
        {"intId": 2, "name": "geometryAreas", "composition": "GeometryArea", "multiplicity": "*", "sortedByKey": true},
        {"intId": 3, "name": "sampledGeometryAreas", "composition": "SampledGeometryArea", "multiplicity": "*",
"sortedByKey": true},
        {"intId": 4, "name": "functionalAreas", "composition": "FunctionalArea", "multiplicity": "*", "sortedByKey": true},
        {"intId": 5, "name": "propertiesAreas", "composition": "PropertiesArea", "multiplicity": "*", "sortedByKey": true},
        {"intId": 6, "name": "trackUsageAreas", "composition": "TrackUsageArea", "multiplicity": "*", "sortedByKey": true}
      ]
    }
  ]
}
```

SPT2TS-125119 - Here are the scripts generating output. **[🔍Content to be approved]**