railML® 2.x Infrastructure
Topologic modelling of switches and crossings

Christian Rahmig
Outline

- What is railML® Infrastructure?
- Topology modelling:
  - The principle of Topology Modelling with railML
  - Modelling of a simple switch
  - Modelling of a simple crossing
  - Modelling of a simple switch crossing
  - Modelling of a double switch crossing
Introduction
What is railML® Infrastructure?

- Track Topology
- Track Geometry
- Track Topography and railway service-relevant data

... Graph with Nodes and Edges
Introduction
What is railML® Infrastructure?

- Track Topology
- Track Geometry
- Track Topography and railway service-relevant data

...straight lines, curves, increasing/decreasing slope
Introduction
What is railML® Infrastructure?

- Track Topology
- Track Geometry
- Track Topography and railway service-relevant data

... signals, platforms, tunnels, electrification etc.

The principle of Topologic Modelling with railML

Example: simple switch
The principle of Topologic Modelling with railML

Option 1: simple switch at the end of a track
The principle of Topologic Modelling with railML

- Option 1: simple switch at the end of a track
The principle of Topologic Modelling with railML

- Option 2: simple switch in the middle of a track
The principle of Topologic Modelling with railML

- Option 2: simple switch in the middle of a track
Institute of Transportation Systems
railML.org meeting in Paris > 19. September 2013 > Slide 11

Track
- id [GenericID]
- name [GenericName]
- description [ElementDescription]
- type [TrackType] = [mainTrack, secondaryTrack, connectingTrack, sidingTrack, stationTrack]
- mainDir [DirValidity] = [none, up, down, both, unknown]
- infraAttrGroupRef [GenericRef]

TrackTopology

Connections
- id [GenericID]
- pos [LengthM]
- absPos [LengthM]
- absPosOffset [LengthM]
- trackContinueCourse [Course] = [left, right, straight]
- trackContinueRadius [RadiusM]
- normalPosition [Course] = [left, right, straight]
- model [string]
- length [LengthM]
- type [SwitchType] = [ordinarySwitch, insideCurvedSwitch, outsideCurvedSwitch, threeWaySwitch]

Switch

Crossing
- id [GenericID]
- pos [required: LengthM]
- absPos [LengthM]
- absPosOffset [LengthM]
- trackContinueCourse [Course] = [left, right, straight]
- trackContinueRadius [RadiusM]
- normalPosition [Course] = [left, right, straight]
- model [string]
- length [LengthM]
- type [CrossingType] = [simpleCrossing, simpleSwitchCrossing, doubleSwitchCrossing]

BufferStop

Connection

GeoCoord
- coord [CoordinateList]
- extraHeight [double]
Modelling of a simple switch
Modelling of a simple switch

Orientation and course:

- Incoming Right
- Outgoing Left
- Incoming Left
- Outgoing Right
Modelling of a simple crossing

<track id="34" name="track_34">
<track id="39" name="track_39">
<track id="55" name="track_55">
<track id="70" name="track_70">
<track id="340" name="track_340">
<trackTopology>
  <trackBegin id="34001" pos="0">
    <connection id="34099" ref="152" />
  </trackBegin>
  <trackEnd id="34002" pos="31.5">
    <connection id="34098" ref="39099" />
  </trackEnd>
  <connections>
    <crossing type="simpleCrossing" pos="31.5" id="34097">
      <connection orientation="incoming" id="34096" ref="55098" />
      <connection orientation="outgoing" id="34095" ref="70090" />
    </crossing>
  </connections>
</trackTopology>
</track>
<track id="390" name="track_390">
<track id="550" name="track_550">
<track id="700" name="track_700">
<track id="3470" name="track_3470">
<track id="3955" name="track_3955"
Modelling of a simple switch crossing

- Option 1: use switches and crossings
  - 1 simple crossing
  - 2 ordinary switches
  - 3-5 extra tracks
Modelling of a simple switch crossing

Option 2: the qualified element simpleSwitchCrossing
Modelling of a double switch crossing

- Option 1a: use switches and crossings
  - 4 ordinary switches
  - 4 extra tracks
Modelling of a double switch crossing

- Option 1b: use switches and crossings
  - 4 ordinary switches
  - 6 extra tracks
Modelling of a double switch crossing

- Option 1c: use switches and crossings
  - 4 ordinary switches
  - 6 extra tracks
  - 1 simple crossing
Modelling of a double switch crossing

- Option 1: use switches and crossings
  - 1 simple crossing
  - 4 ordinary switches
  - 6 extra tracks
Modelling of a double switch crossing

Option 2: the qualified element doubleSwitchCrossing
Modelling of a double switch crossing

Option 2: the qualified element doubleSwitchCrossing
Thank you for your attention!

Christian Rahmig
coord@infrastructure.railML.org
railML® Infrastructure v3 concept
Towards a new infrastructure model

Christian Rahmig
Different maps for different applications.
A New Data Model

Problem

railML: Topology
IDMVU: Netzmodell

Node-Edge Model

Is it sufficient for a complete modelling of the railway infrastructure?
A New Data Model

Problem

What is the smallest unit?

(x, y, z)
A New Data Model
The basis

- Smallest unit is the point in 4D (lat, lon, alt, t)
- All other elements consist of a number of elements of this smallest unit
Topology model

Concept Infrabel:

- **Topology**
  - Edges
    - ID
    - From Node
    - To Node
  - Nodes
    - ID
  - Connectivity
    - From Edge
    - To Edge
    - Allowed

- **Node Definition**
  - Nodes
    - ID
  - Port List
    - ID
    - Distance from center point

- **Edge Definition**
  - Edge
    - ID
    - From Port
    - To Port
    - Length

- **Geometry**
  - Representation
    - SRS
    - Units
  - Nodes
    - ID
    - Geometry
  - Edges
    - ID
    - Geometry
Topology model
The basis

Possible approach within railML 3.0:
- node
- edge
Topology model
The basis

- Possible approach within railML 3.0:
  - node
  - edge
  - intersection groups connections
  - connection defines topological way from one edge to another edge
Topology model
The basis

Source:

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<railML>
  <infrastructure version="3.0">
    <topology>
      <nodes>
        <node id="1" name="node_1" />
        <node id="2" name="node_2" />
        <node id="3" name="node_3" />
        <node id="4" name="node_4" />
      </nodes>
      <edges>
        <edge id="5" name="edge_5" beginNodeRef="2" endNodeRef="1" />
        <edge id="6" name="edge_6" beginNodeRef="1" endNodeRef="3" />
        <edge id="7" name="edge_7" beginNodeRef="4" endNodeRef="1" />
      </edges>
      <intersection id="8" name="intersection_8" nodeRef="1">
        <connections>
          <connection id="9" name="conn_5_6" fromEdgeRef="5" toEdgeRef="6" />
          <connection id="10" name="conn_6_5" fromEdgeRef="6" toEdgeRef="5" />
          <connection id="11" name="conn_6_7" fromEdgeRef="6" toEdgeRef="7" />
          <connection id="12" name="conn_7_6" fromEdgeRef="7" toEdgeRef="6" />
        </connections>
      </intersection>
      <intersection>
        <intersection id="8" name="intersection_8" nodeRef="1">
          <connections>
            <connection id="9" name="conn_5_6" fromEdgeRef="5" toEdgeRef="6" />
            <connection id="10" name="conn_6_5" fromEdgeRef="6" toEdgeRef="5" />
            <connection id="11" name="conn_6_7" fromEdgeRef="6" toEdgeRef="7" />
            <connection id="12" name="conn_7_6" fromEdgeRef="7" toEdgeRef="6" />
          </connections>
        </intersection>
      </intersection>
    </topology>
    <trackNetwork>
      <infrastructure>
    </railML>
```
Topology model
Positioning in the rail network

- Pure topology: there are no coordinates and no geometry → also no length
- New element: **Trail**
  - A trail references an edge and gives it a length
  - Length = distance between the connected nodes
  - But: there is *more than one distance*…
Topology model
Positioning in the rail network

“Virtual” distances:
Topology model
Positioning in the rail network

- Pure topology: there are no coordinates and no geometry → also no length
- New element: **Trail**
  - A trail references an edge and gives it a length
  - Length = distance between the connected nodes
  - But: there is more than one distance…: `fromNodeLength`, `betweenNodeLength`, `endNodeLength`

```xml
<trailNetwork>
  <trails>
    <trail id="13" name="trail_5" edgeRef="5" fromNodeLength="0" betweenNodeLength="369" endNodeLength="0" />
    <trail id="14" name="trail_6" edgeRef="6" fromNodeLength="0" betweenNodeLength="468" endNodeLength="0" />
    <trail id="15" name="trail_7" edgeRef="7" fromNodeLength="0" betweenNodeLength="123" endNodeLength="0" />
  </trails>
</trailNetwork>
```
Topology model
Positioning in the rail network

- Pure topology: there are no coordinates and no geometry → also no length
- New element: **Trail**
  - A trail references an edge and gives it a length
  - Length = distance between the connected nodes
  - But: there is more than one distance…: fromNodeLength, betweenNodeLength, endNodeLength
  - **The trail is the basic element for referencing the railway network’s geometry and elements.**
Topology model
Modelling various levels of details

References
Geometry model

- Railway Geometry in 3D:
  - **Curvature** / Radius [1/m]
  - **Slope** [Promille]
  - **Superelevation** [mm]

- Aim: it should be possible to determine the 3D geometry in every point along the track.

- Alignment approach: **geometry ways** (arcs, straight lines, transition bends)

- Measurement approach: **geometry points** (3D geometry in that point; geometry ways can be determined by “reverse engineering”)
Railway Elements Model

- Concept Infrabel:

- The model proposed by railML does not differ much...
Railway Elements Model

Example platform:

Example platform:

- geoCoordRefs
railML® Infrastructure 3
Extensions

- The core:
railML® Infrastructure 3

Extensions

Example **SwitchLayer** for Map-matching purposes:
Example **SwitchLayer** for Map-matching purposes:
- intersectionRef
- leftBranchConnectionRef
- rightBranchConnectionRef
- leftBranchRadius
- rightBranchRadius
- distanceToClearancePost
Thank you for your attention!

Christian Rahmig
cord@infrastructure.railML.org