



railML.org

Interlocking Schema

39th railML Conference – April 21st 2021

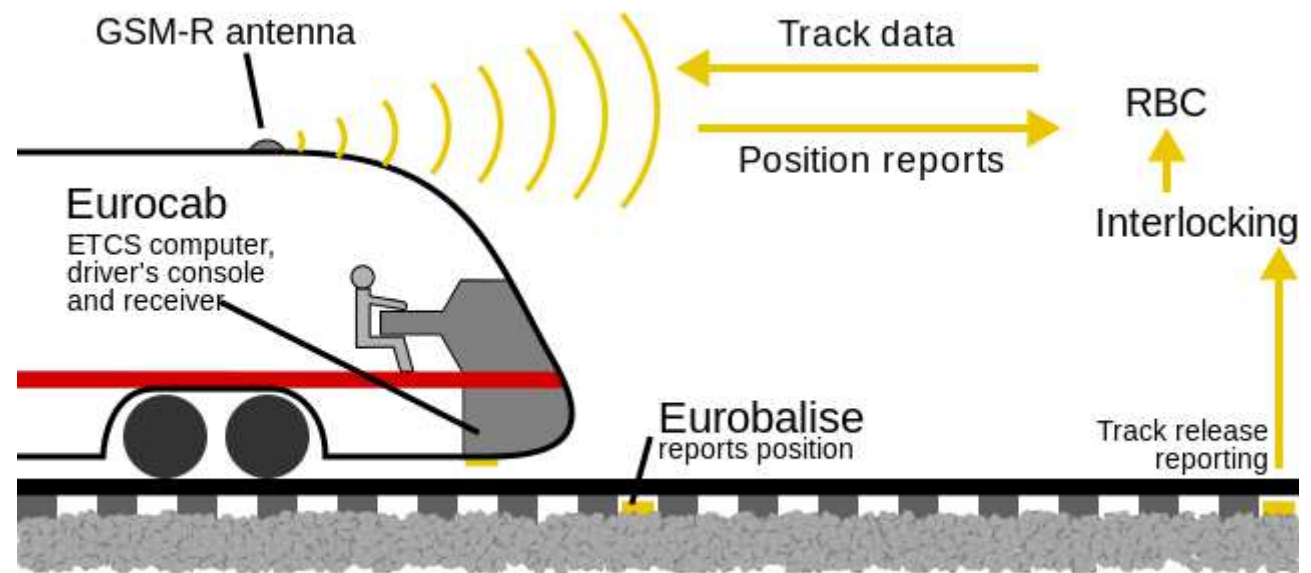
Agenda

1. Development in Interlocking Schema
2. Forum Posts
3. Schema Documentation
4. Next steps



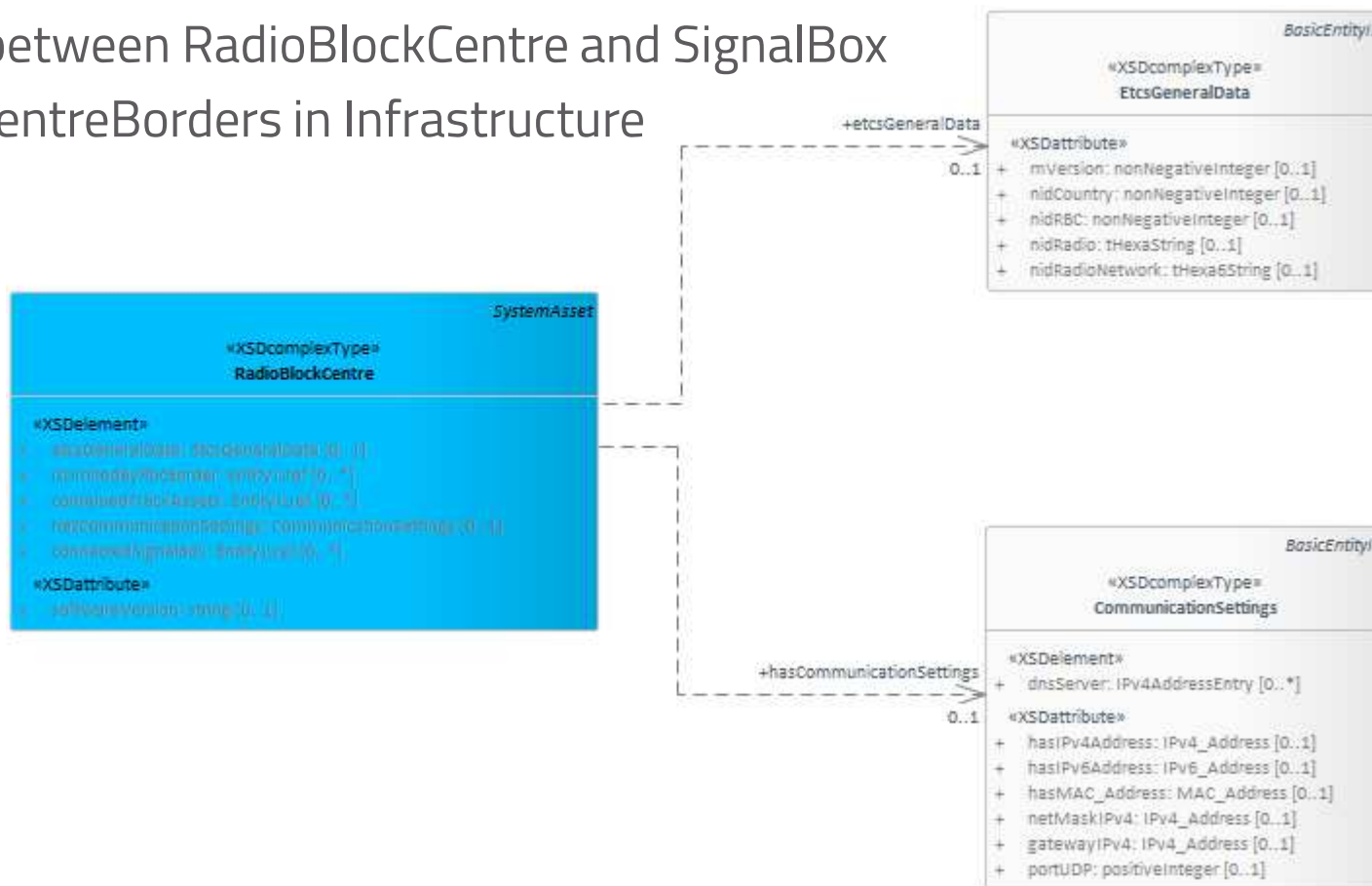
1. Development in Interlocking Schema

- No interlocking specific working group
- Current development driven from ETCS working group under lead of infrastructure
- Some inputs by forum posts
- Feedback only from two users
- Enhancement of use case IMED



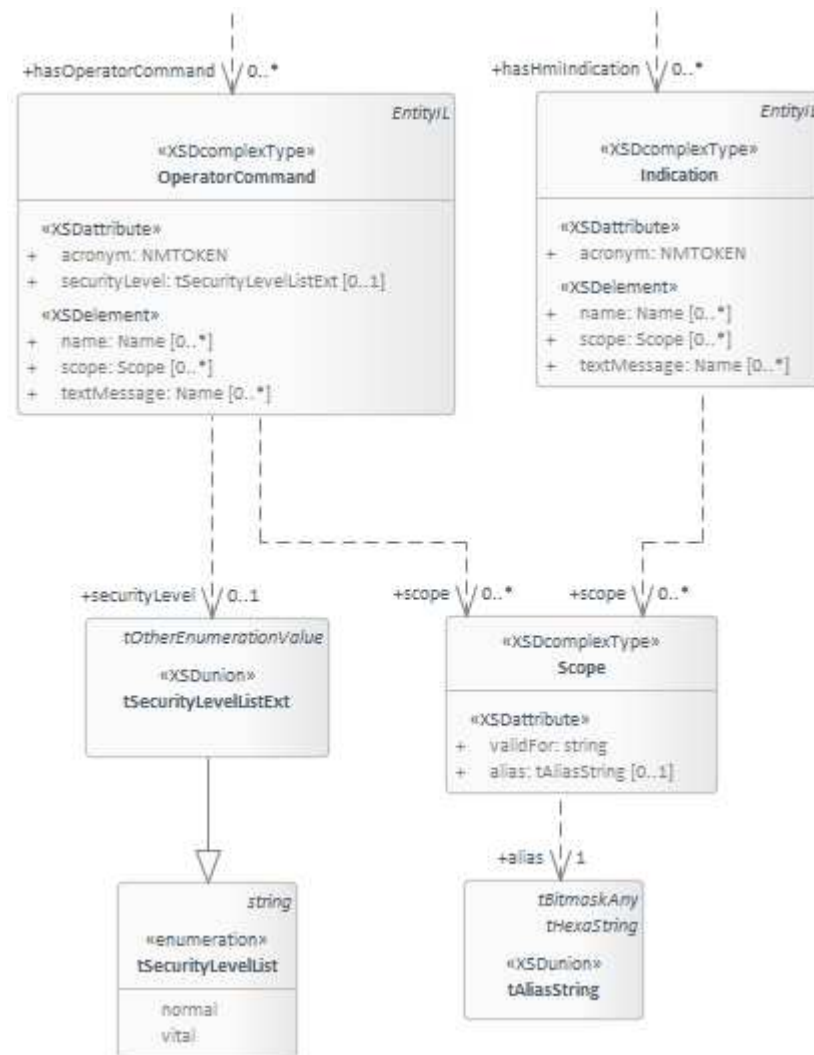
1. Development in Interlocking Schema

- RadioBlockCentre in Interlocking – ticket #386 in parallel to SignalBox
- References between RadioBlockCentre and SignalBox
- RadioBlockCentreBorders in Infrastructure



1. Development in Interlocking Schema

- Operator commands and indications (HMI) – ticket #451
- definition of commands and indications as "genericTypes" for a specificInfrastructureManager
- references from any element/indicator to defined commands and indications



1. Development in Interlocking Schema

- Aggregation of elements – ticket #465
using same principle as in Infrastructure
reference to same class as parent

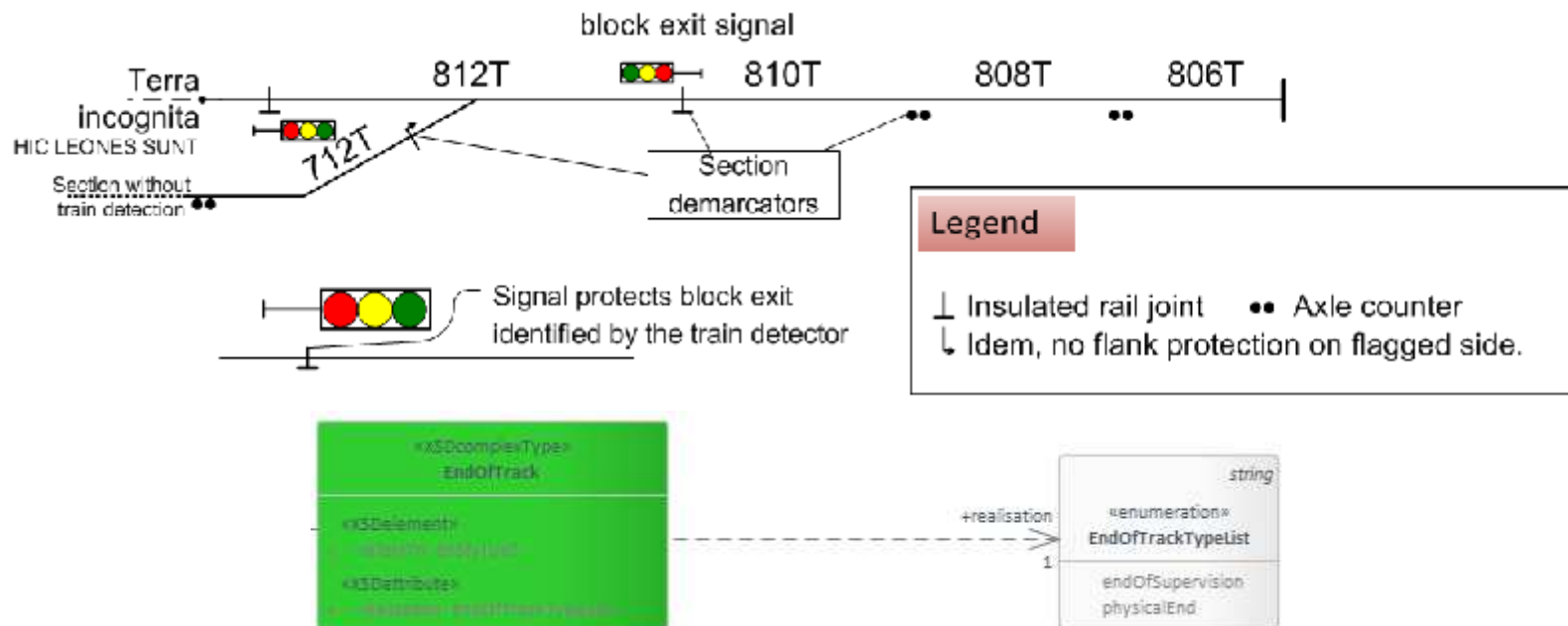
```
<xs:attribute name="belongsToParent" use="optional" type="rail3:tRef" />
```

- Harmonising common data – ticket #468
<hasCommunicationSettings> for SignalBox, RadioBlockCentre, Controller



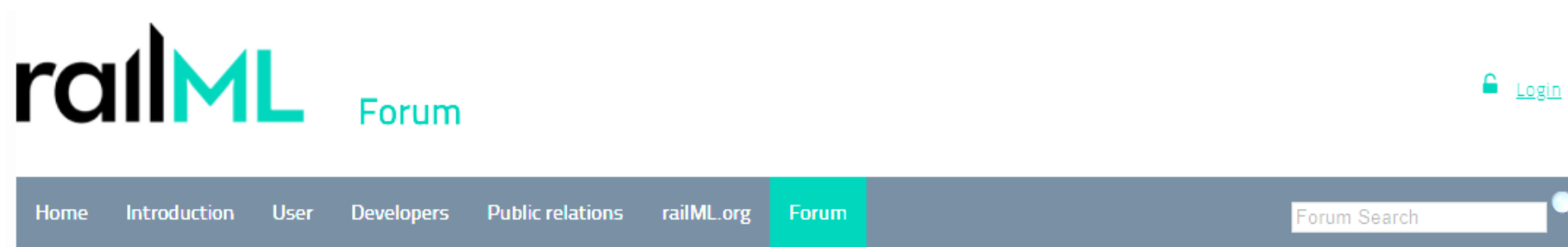
1. Development in Interlocking Schema

- EndOfTrack
kind of border element for Interlocking purpose
handling of physical end and limit of supervision



2. Forum Posts

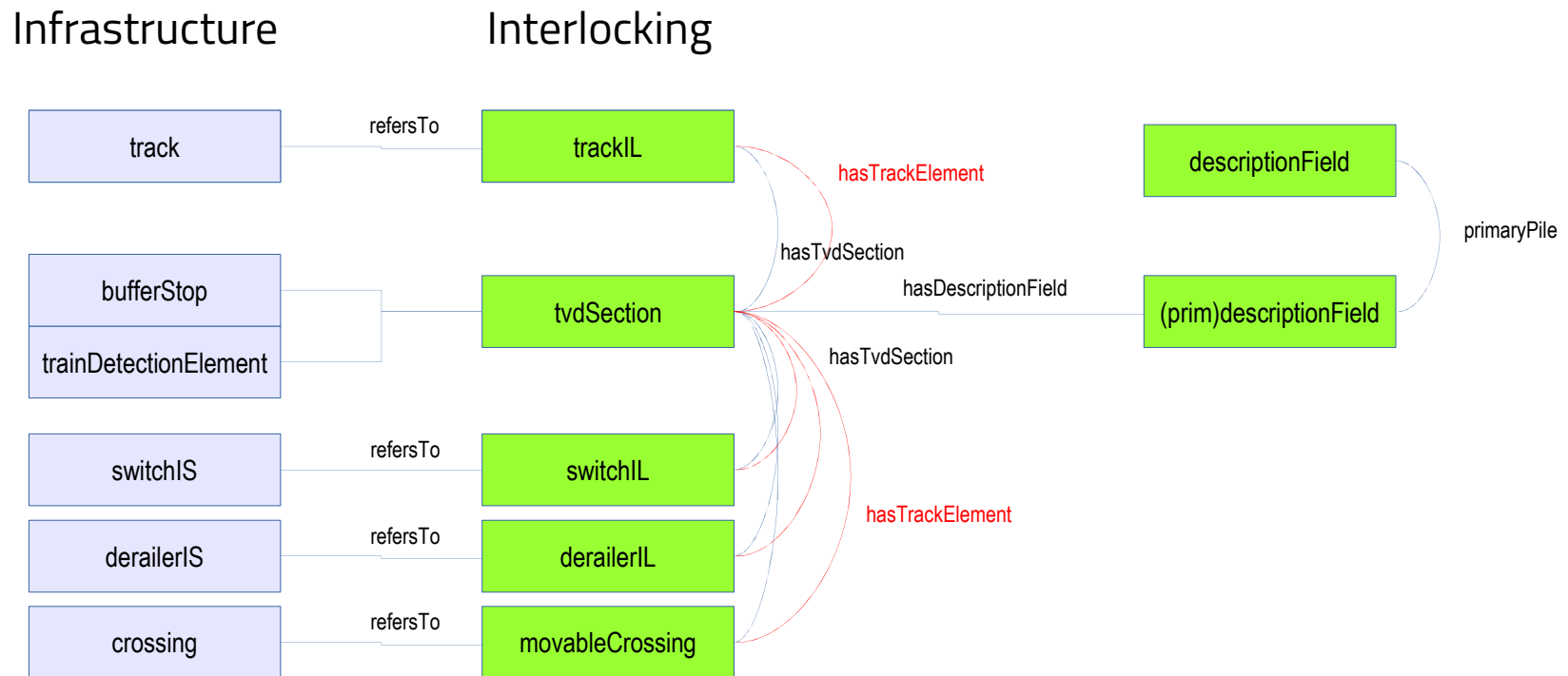
1. Referencing between IS and IL
2. Flank protection
3. What is the rationale for multiple <assetsForIL>s?
4. Signal aspect changes and simulation
5. Restricted Areas: limitedBy vs. elements inside
6. special infrastructure in IL - bascule bridge, tunnel gates
7. Train Number Description Field



2. Forum Posts

1. Referencing between IS and IL

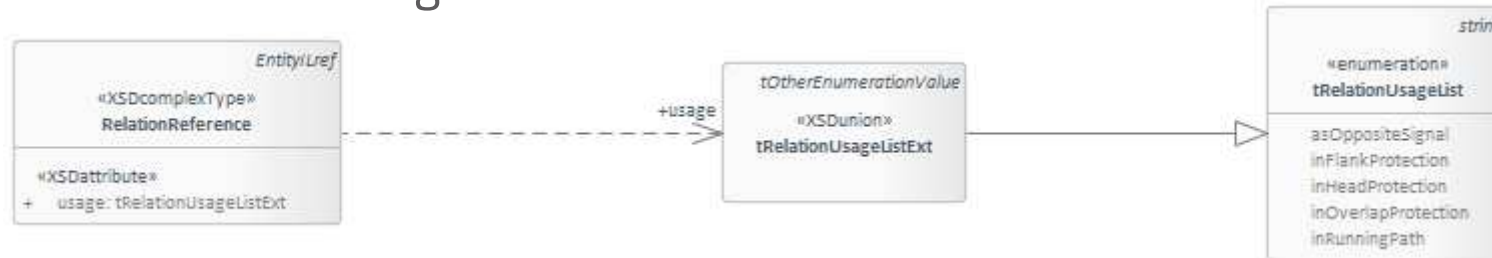
- Overview of possible references between IS and IL elements



2. Forum Posts

2. Flank protection

- can be defined using RouteRelations



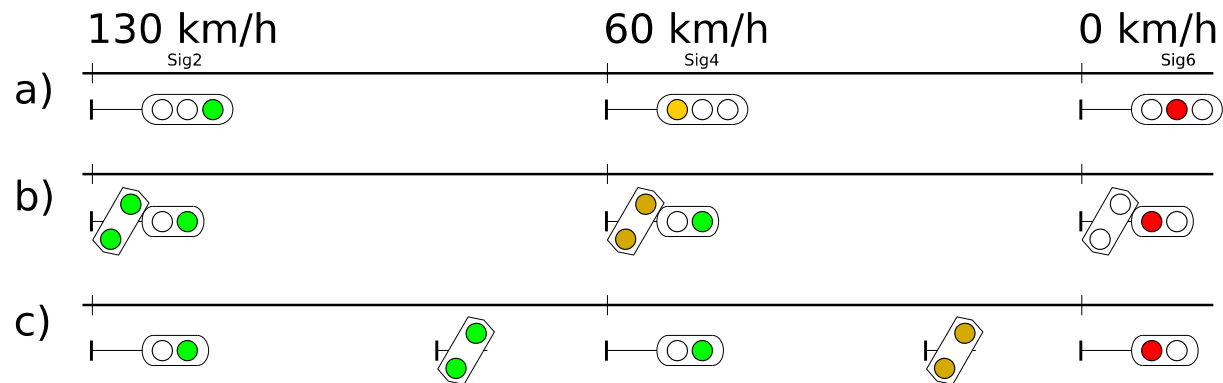
3. What is the rationale for multiple `<assetsForInterlocking>`s?

- having more than one `<assetForInterlocking>`
 - ➔ different phases of evolution or different InfrastructureManagers

2. Forum Posts

4. Signal aspect changes and simulation

- Dependency of signal aspects in <implementsSignalPlan> of <signalBox>

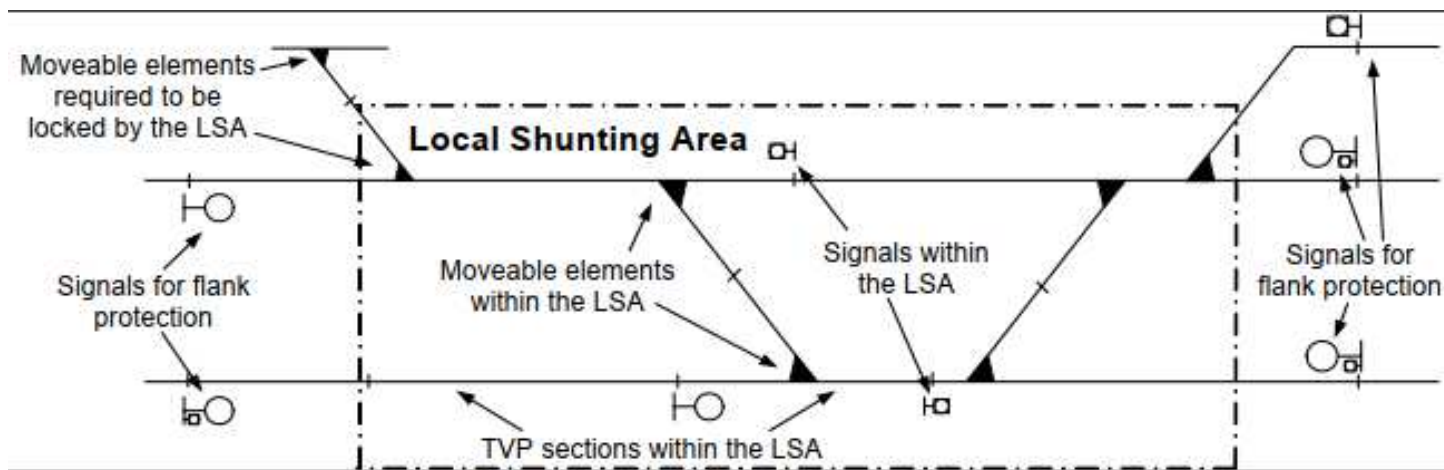


- Interlocking functions, i.e. handling of routes or signals depending on the situation, are not yet part of the schema

2. Forum Posts

5. Restricted Areas: limitedBy vs. elements inside

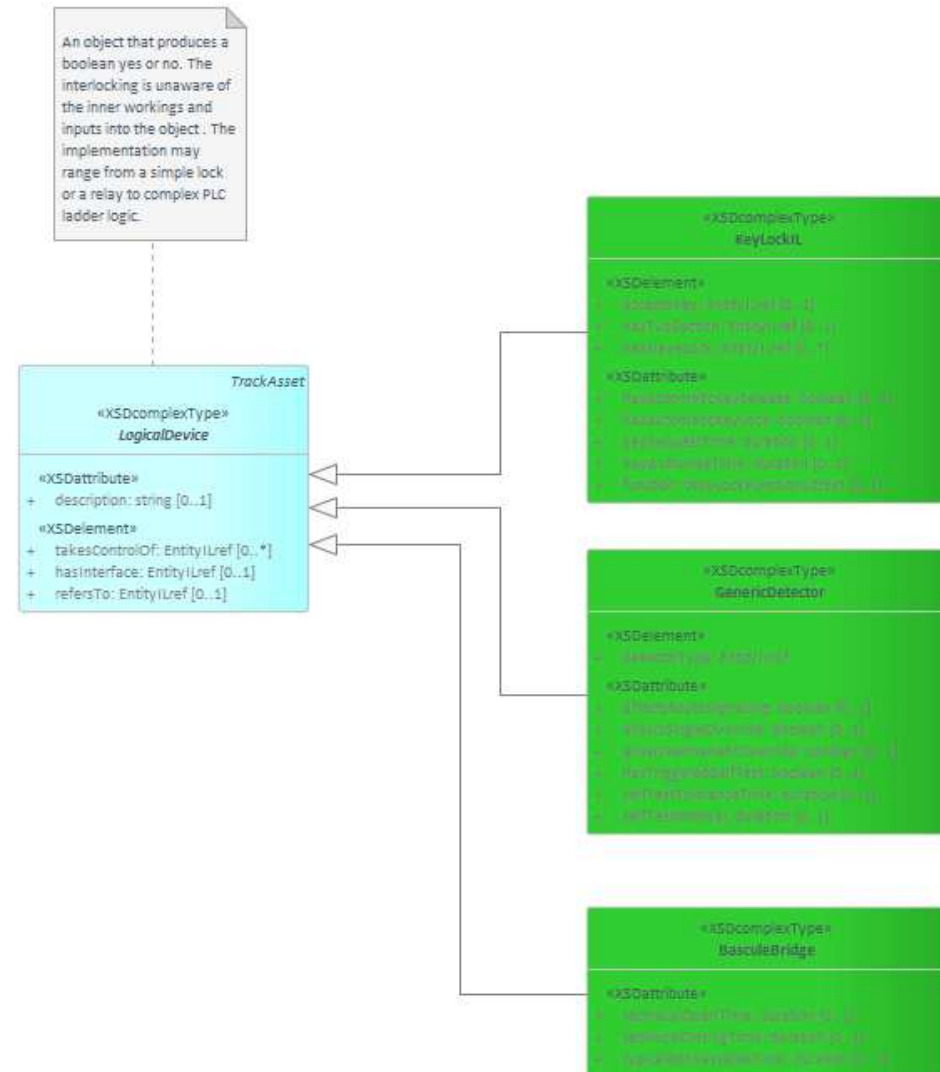
- Both were originally included but `<trackAssetInArea>` removed by community vote
- Now requested to have it again
→ issue of border element is without info on which side of the area



2. Forum Posts

6. special infrastructure in IL - bascule bridge, tunnel gates

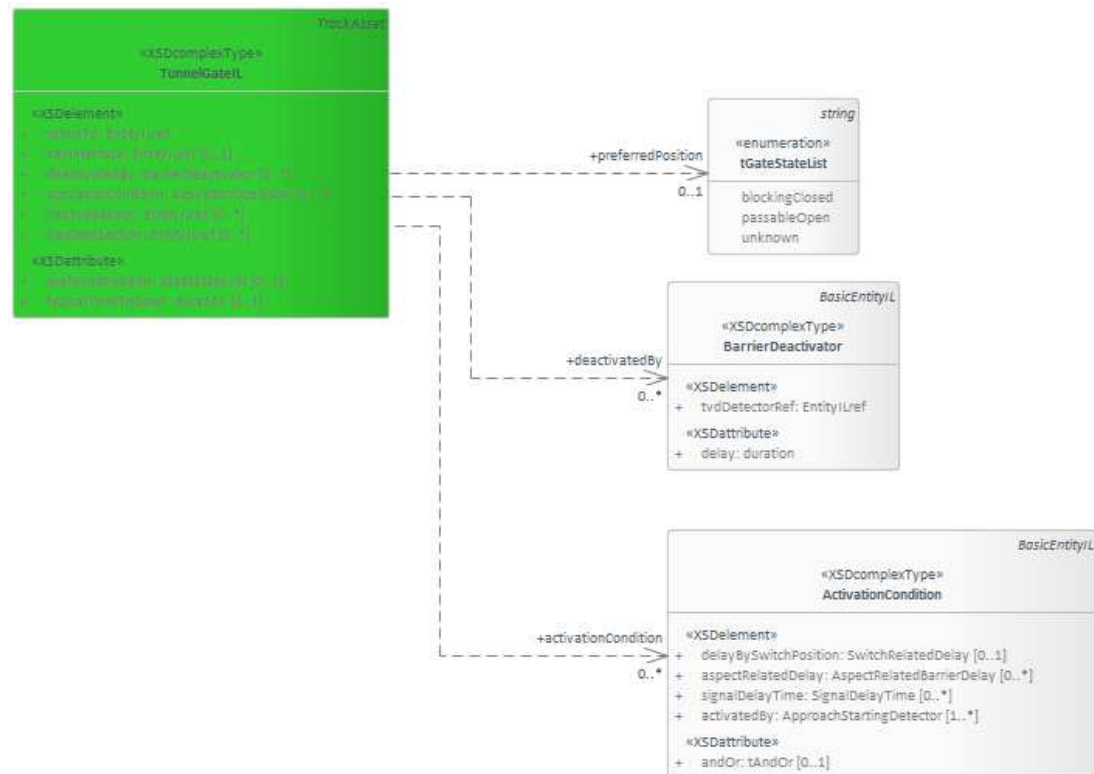
- Similar information needed as keylock or level crossing – depending whether which positions are controlled by interlocking
- BasculeBridge as LogicalDevice only locked position of interest
- Interlocking does not control the bridge drive
- No change in Infrastructure needed – Overcrossing



2. Forum Posts

6. special infrastructure in IL - bascule bridge, tunnel gates – ticket #450

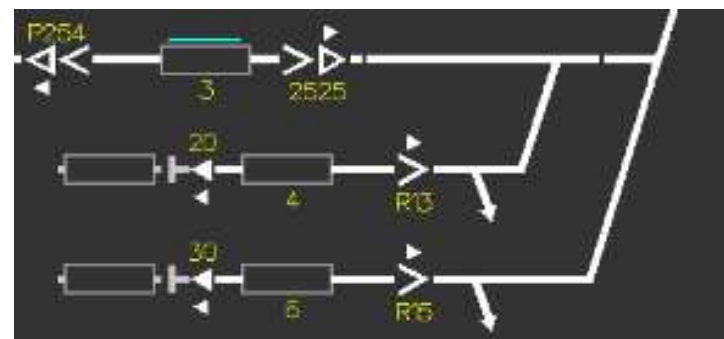
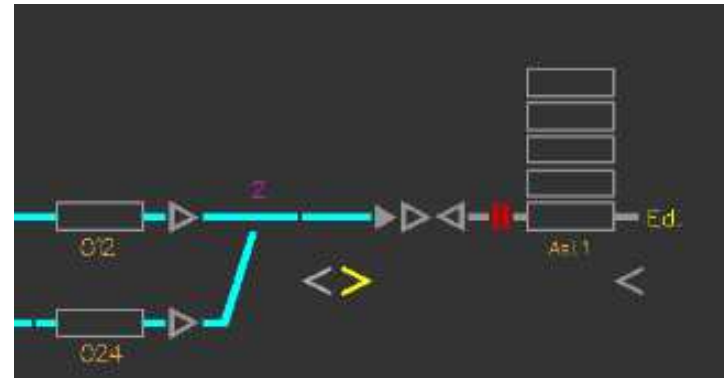
- TunnelGateIL derived from LevelCrossing
- Both positions controlled, with approach information
- Counterpart TunnelGateIS needed as positions might be not only at tunnel portals



2. Forum Posts

7. Train Number Description Field

- Reference from TvdSection



3. Schema Documentation

https://wiki3.railml.org/index.php?title=Main_Page

Semantics

Best Practice / Examples

The logic for control of track assets and train traffic is housed in a physical compartment traditionally called signalbox. In order to avoid ambiguity one has to bear in mind the clear discrimination. The entire sub-scheme is called as "interlocking" but the term is not used here without addition of "scheme". Whenever the pure term interlocking is used, it will refer to a particular interlocking logic located inside a signalbox.

The `<signalBox>` combines the information of the related network in the interlocking. Therefore the description of is mainly a list of references to the related items needed for the interlocking logic.

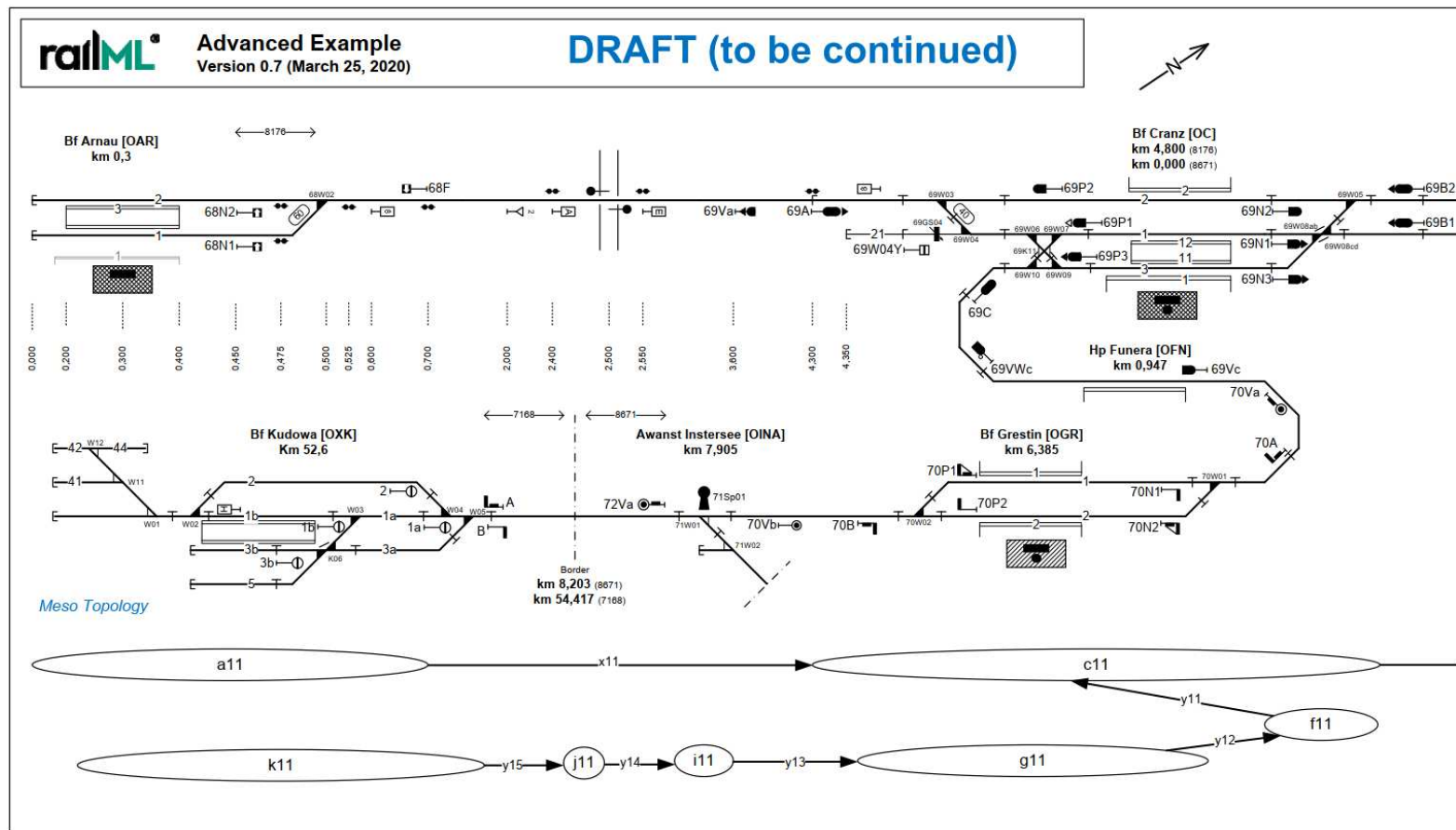
- `<controlsSystemAsset>` – This is the reference to any system asset which is controlled by this interlocking. In addition it gives also the extent of control the interlocking has of the asset.
- `<controlsTrackAsset>` – This is the reference to any track asset which is controlled by this interlocking. In addition it gives also the extent of control the interlocking has of the asset.
- `<controlsRoute>` – This is the reference to any route the interlocking controls.
- `<controlsCombinedRoute>` – This is the reference to any route combination the interlocking controls. For details refer to chapter 1.38.
- `<implementsSignalplan>` – This contains the list of signal aspect relations the interlocking has to consider when steering any signals. For details refer to chapter 1.34.
- `<implementsElementGroup>` – This contains the list of element groups that are operated together with a common command. Additionally the type of group is given as reference to a definition in `<hasElementGroupType>`. The referred elements in this list must be all controlled by this interlocking to the same level of extent.
- `<hasPermissionZone>` – This is the reference to any permission zone within the area controlled by this interlocking.
- `<hasConflictingRoutes>` – This is the reference to any route pairing that cannot be used simultaneously. For details refer to chapter 1.30.
- `<hasConfiguration>` – This gives some general information about the particular interlocking. For details refer to chapter 1.35.

The example shows an interlocking with a simplified list of features. It gets status information from the power supply `ups01`. It fully controls the TVD section `A01T` and the switch `pt_sw01`. It commands the virtual signal `mb_sig01` without getting any status information back. This `@extentOfControl` is explained here. This interlocking controls the routes `rt_sig02_sig04` and `rt_sig01_sig04` and has a related signalplan for the routes. In addition it knows the group `estop01` to set the listed signals to stop and has some configuration information.

```
<signalBox id="ilx01">
  <designator register="_SimpleRegister" entry="ILX-ARN"/>
  <controlsSystemAsset extentOfControl="notificationOnly">
    <connectedSystemAsset ref="ups01"/>
  </controlsSystemAsset>
  <controlsTrackAsset extentOfControl="fullControl">
    <connectedTrackAsset ref="A01T"/>
  </controlsTrackAsset>
</signalBox>
```

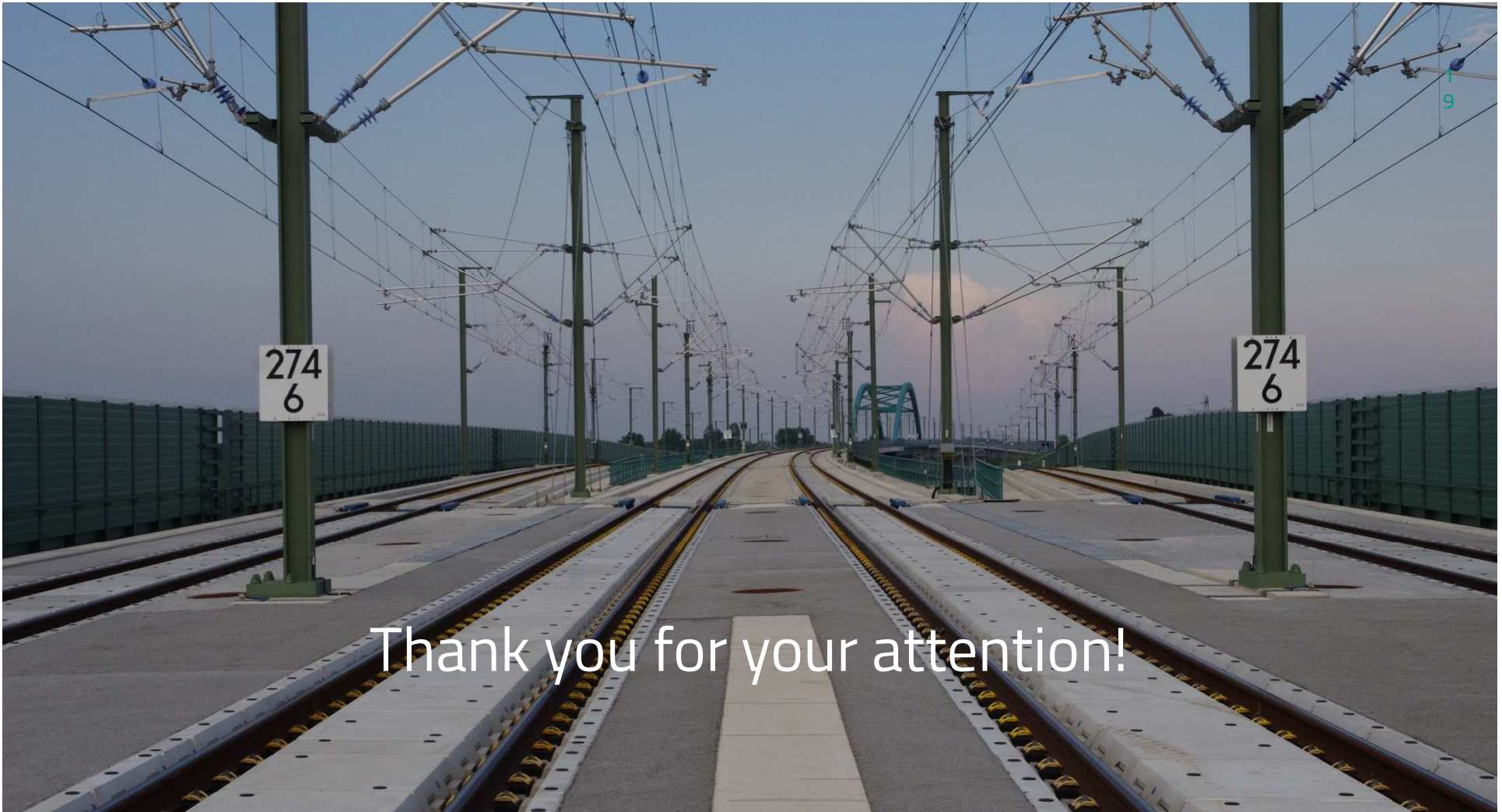

3. Schema Documentation – Advanced Example

- Extension based on Simple Example v11 with Topology
- just first draft as schematic plan + draft railML file



4. Next steps/tasks

- Target release of final railML 3.2
- Collection of feedback and best practise
- Enhancement of documentation: wiki & tutorial
- Completion of IL part of advanced example
- Further development improving extent of existing use case
- No additional IL use case
- Cooperation with IS for ETCS/ITMS use case
- Review of open Trac tickets for railML3.x
- Discussions on forum



Thank you for your attention!



www.railml.org
railML.org

Jörg von Lingen
Interlocking coordinator



coord@interlocking.railml.org



+49-351-8775940