

Dear all,

the element <track> covers two independent concepts:

- navigation per definition (A Track is defined by a railway section between two switches/crossings ...)
- usage pattern - main, secondary, siding and main usage direction.

Assume the application importing the RailML file should find a train path. Now it can use two different approaches:

- NetElements with NetRelations
- Track and TrackNodes (introduced with track/trackBegin and track/trackEnd)

This requires, that in each file NetElements and Tracks cover the entire network identically - otherwise the applications would create different paths depending on the used approach.

To prevent this situation, I would propose to remove navigability from the Track-definition. The first step would be renaming of Track, as it imposes navigability from RailML-2, into UsagePattern. The application creating paths for trains would have to use NetElements and consult functionalInfrastructure for UsageType, UsageDirection, Gauge etc. for each NetElement it traverses.

Example:

```
<usagePattern type="main" mainDir="up" owner="DB">
  <netElementRef ref="ne_1"/>
  ...
  <netElementRef ref="ne_100"/>
</usagePattern>
<usagePattern type="siding" mainDir="both" owner="SNCF">
  <netElementRef ref="ne_101"/>
  ...
  <netElementRef ref="ne_200"/>
</usagePattern>
```

Regards
Stefan Wegele
