Subject: ocp's/stations and their properties
Posted by on Tue, 10 Apr 2012 17:48:01 GMT
View Forum Message <> Reply to Message

Dear Christian and all "along-readers",

I would like to draw your attention to one more "well-known" item I'd like to see solved in RailML 2.2.

Currently, it is not possible that one station has different properties (RailML: <ocp>.<propService>, <propOperational>, <propEquipment> a. s. o..) at different tracks or lines. This is because the properties attributes are only given at the <ocp> but not at the tracks.

From our experience, there are often stations where two or more lines meet and which have different properties depending on the line. First I thought these are seldom special cases but on the contrary - after a closer look you'll find plenty of them.

I think one can easily imagine a station where two lines meet and which has platforms at one line but has no platforms at the other line. So, the attribute "<propService>.passenger" should be 'true' at the tracks of the first line and 'false' at the tracks of the other line.

Also, you'll find often the arrangement where a junction for one line is a head of a station for the other line. (A line branches off at the head of a station but does not pass through the station.) A typical German example is Unterlemnitz (coords 50.469913° 11.624211°, http://www.openstreetmap.org/index.html?mlat=50.469913&m lon=11.624211&zoom=16) which is a junction for line #6683 (the line to the north-east) and a station for line #6709 (the line to the west). There are many of these examples, e. g. Bissenhofen, Niederhone/Eschwege West, Augsburg-Hochzoll, Dresden-Pieschen. Leipzig-Neuwiederitsch, Arnsdorf Nord/West, and many more.

We also will find them in other countries but it soon becomes clear that it depends on the definition of a 'station'. In countries which do not have the railway-operational term 'station' (like US or Canada) there are no such problems. (The term 'station' still refers to the traffic function - a place for accessing the railway - but not the operational function where there are just signal box areas or junctions and nothing more.)

So, for RailML we have two possibilities:
  a) either simply to allow track-depending properties of a station because of the plenty examples where this is necessary,
  b) or to avoid the term 'station' in an operational sense with all its properties.

a) I think for RailML 2.2 only the first is practicable. This can be done by repeating the properties of a station at the <track>.<trackTopology>.<crossSections> element below the "ocpRef" attribute. This would mean: If there are such attributes at <crossSections>, they overwrite the same attributes of the <ocp> element of "ocpRef". If there are no such properties at <crossSections> all stays as it is and the properties of "ocpRef" are valid.

b) We could possibly follow the (b) solution from RailML 3.0. This would mean to delete all the properties like <propService>, <propOperational>, <propEquipment> of an <ocp>. We should then introduce track elements (in <trackTopology> or <trackElements>) like "beginOcpArea" and "endOcpArea" both with an ocpRef attribute. All which lies in an ocp area belongs to the ocp. It is then up to a reading programme to decide whether it sees the ocp as a station or as a junction or as a halt or whatever. (For example, if there are at least two signals per direction between the station boundaries, it would be a station after the German definition of a station. If there is one signal per direction and at least one point, it is a junction, one signal and no point is a block post a. s. o.)

This would make it very difficult for a reading programme even to do such simple things as finding all passenger stations. Up to now (RailML 2.2) one can find the passenger stations simply by scanning all ocp’s for <propService>.passenger=true. If we would delete these "station properties" from RailML 3.0 on, this would become to "scan all tracks for station areas which have at least one platform". So, I think despite all the arbitrariness which lies in the term 'station' and its properties, we should keep them even after RailML 3.0

Conclusion: I herewith make a request for the above described first solution (a) for RailML 2.2.

Best regards,
Dirk.