Subject: Re: simple Infrastructure scheme usage as a base for Timetable reference
Posted by Dr. Volker Knollmann on Mon, 22 Sep 2008 16:36:19 GMT
View Forum Message <> Reply to Message

Hello Dirk,

thank you for your message and please excuse my late reply. Regarding your first question about defining special positions as reference points for time table purposes ("Fahrzeitmesspunkt") I think I've good news. You requested a link between a track and an ocp and I can confirm that exactly such a link already existis. The associated element is called "crossSection" an dis located in the infrastructure part.

A crossSection is bound to a certain location on the track (identified by lineID, trackID and position on track) and references an ocp via its IP. As far as I know from RailML's history, this element has been introduced with exactly the same thought in mind that you descibed in your original posting.

Regarding your other two questions (attribute for the number of tracks / empty <track> elements), I'm afraid to have no such easy answers for you. An attribute for the number of regular tracks per line is so far not implemented. Unfortunately, you cannot derive it from the number of <track>-elements belonging to that line, because cross-overs, passing loops, station tracks etc. lead to numerous <track>-elements all inside the same <line>-section. If there really is a common need to add an appropriate attribute for the number of tracks, we can insert it in a future RailML release. For the time being I suggest to used the <anyAtribute>-extension which allows you to add arbitrary attributes to almost all RailML elements.

Your approach to use "empty <track>-elements" in order to indicate the number of operational tracks does definitely not conform with the RailML-architecture. The <track> is a container for the "physical" tracks and all other (physical) elements along that track. With its sub-element <trackTopology> (which owns <trackBegin>, <trackEnd> and <mileageChanges> among others), the coordinate system for subsequent position information is established. So these elements contain crucial data for the interpretation and evaluation of the infrastructure's topology. For this reason I doubt that making <trackBegin> and <trackEnd> optional elements - as you suggested in your posting - is a good move. This would affect the very heart of the infrastructure schema and would lead to much more effords for parsers and import filters. We should have a certain set of core data which every RailML-capable application can rely on. An in my opinion, a consistend track description including length etc. is part of that core data.

I hope that these explanations answer your questions. If not, please don't hesitate to get in touch with me!

Best regards,
Volker

--

Dr. Volker Knollmann
Siemens AG
Industry Sector
I MO RA SPP SM MT 1
Ackerstr. 22
38126 Braunschweig
Tel.: +49 (531) 226-2592
mailto:volker.knollmann@siemens.com