
Subject: Re: Modified V94_12

Posted by [nries](#) on Wed, 19 Nov 2003 22:16:47 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello,

>> It is for sure a possible solution, although the whole element might get
>> rather clumsy if you try to cover all technical solutions of communication
>> between engine and infrastructure.

> Yes, I definitely agree to your point. But in a first step, we don't need
> to include all possible means of communication in this group. I just
> demonstrated with some examples, how <ocsElements> should be used. Perhaps
> "LEU" was way too exotic... I should have chosen "<axleCounters>" for
> example...
> The number and kind of "sub-elements" for the container <ocsElements> can
> be defined depending on the needs of the first applications. But I think
> it's a good idea to have at least the most important systems (track
> circuits, axle counters, LZB, ETCS) at hand.

Here you are risking an endless discussion about the "most important"
systems, because this is the german point of view...

Still you are right that for a version 1.0 we will have to have a limited
choice of subelements in the <ocsElements>.

Altogether I agree to integrate all these kinds of elements into the
<ocsElement>.

>> The <signal> element will need further
>> development in order to be practically applicable.

> YES. The same applies to various other elements (e.g. switches). I'm
> having a list with notes and comments regarding railML on my desk and this
> list is growing continuously. So you will have to suffer from some more of
> my postings in this newsgroup in the future...

You will probably have taken notice of my propositions concerning the
signal element. What do you think of them? Unfortunately Matthias has not
integrated them into his scheme so that now we are still working with its
first version.

>> Still there might be some occasions where the possibility to connect two
>> <track> elements will be useful, e.g. if you want to combine two sections
>> of infrastructure already existing in RailML (analogy to OpenTrack), there
>> will be the need to refer to the appropriate point in the neighbour model.

> You think of distributing one line over several documents (like
> OpenTrack)? Okay, in this case we need a function to combine two <track>s,

> that's right.

>> In that concern one single attribute "nextTrackId" will not be sufficient
>> because the point of reference on the given track is not specified.

> Yes, I've seen that problem, too. You really need a double-linked list
> (with all its drawbacks).

>> My version provides a <connection> element containing the attributes
>> "trackID", "pos" and "dir" (sense of mileage).

> According to Annex B of your thesis, you completely skipped <tracks> and
> <track>. And you renamed <trackData> to <linaData> and assigned all
> information directly to the <line>. Apart from the fact that was a useful
> simplification (in my opinion), I found no <connection>-Element in your
> infrastructure-scheme :-(

In fact with my thesis I was still based on an older version. As far as I know the elements <tracks> and <track> have been introduced by Ulrich Linder whose scheme formed the base for Matthias (please correct me if I'm wrong).

My <connection> element ist derived from <kilometer> if that might help you to find it. To fit it into your version it might be best made a subelement of <track> as an alternative to your two new attributes.

Best regards from Paris,

Nikolaus Fries
