
Subject: Re: Modified V94_12

Posted by [volker.knollmann](#) on Wed, 19 Nov 2003 08:41:19 GMT

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

Nikolaus Fries wrote:

> What does LEU mean?

LEU is a "Lineside Electronic Unit" and a part of the ETCS System. On one side, a LEU is connected to the lamp circuits of a conventional signal to determine the signal aspect. On the other hand, it is connected to variable balises to transmit arbitrary ETCS-telegrams depending on the signal aspect. With a LEU, existing tracks can be upgraded to ETCS Level 1 without too expensive changes to the infrastructure (signals, cables, interlocking etc. stay the same).

> Why did you leave <trainProtectionChanges> as an own element?

I interpreted that element as a kind of "instruction", which kind of ATP-system is to be used on lines with multiple systems. So from my point of view, <trainProtectionChanges> does not describe / define / enumerate elements of "command- and control-systems" (thanks for your suggestion!) and is therefore not a part of <ocsElements>.

>> [... introduction of <ocsElements> ...]

> 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.

> 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 continously. So you will have to suffer from some more of my postings in this newsgroup in the future...

>> Therefore, I see the need to compose one line of several
>> tracks only for the use of a macroscopical model, not for the use of a
>> microscopical model.

> Still there might be some occasions where the possibility to connect two
> <track> elements will be useful, e.g. if you want to combine to 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 :-(

Best regards from Braunschweig,
Volker Knollmann
