Subject: Operation and Control System
Posted by nfries on Thu, 27 Nov 2003 16:49:00 GMT

Hello,

concerning the discussion about the modelling of the Operation and Control Systems, I think it necessary to discuss some general aspects.
To be honest, I am not really pleased about the solution with a seperate element for each technical component of the OCS because it will end in an unmanageable collection of different elements having all in common the function to describe a possibility of transferring information between infrastructure and rolling stock.
This collection will have to be updated continously due to the needs of different user groups and might cover in the end different technical solutions for the same function (e.g. different types of balises) depending on the technical realisations implemented in certain national railway networks.
The RailML schemes have the purpose to be valid independant of national or company specific solutions. For that reason I thought about modelling the OCS in a more functional way i.e. replace the lineside components by the information to be transferred. Here we arrive at the problem concerning the future applications using RailML. Simulation programs normally do not ask for technical components but are interested in the transferrable information. On the other hand applications for infrastructure planning demand the opposite.
I am not really sure wether it is possible to satisfy these different needs without having two alternative elements (i.e. redundancy)to be chosen by the user. Maybe we can find a solution right in the middle between the technical and the informational way of representing the OCS.
So far for now - with best regards,

Nikolaus

Subject: Re: Operation and Control System
Posted by volker.knollmann on Fri, 28 Nov 2003 11:53:50 GMT

Hello newsgroup!

Nikolaus Fries wrote:

> To be honest, I am not really pleased about the solution with a seperate
> element for each technical component of the OCS because it will end in an
> unmanageable collection of different elements having all in common the
> function to describe a possibility of transferring information between
> infrastructure and rolling stock.

Basicly, I agree that defining all possible OCS-components will lead to large number of elements. And all elements will have to be designed carefully to provide all neccessary information without being oversized or redundant. And this consumes time. No doubt about that.

> For that reason I thought about modelling the
> OCS in a more functional way i.e. replace the lineside components by the
> information to be transferred.

Uhhhh.... I'm afraid that this is a too abstract approach to this complex topic. First of all, the information that is exchanged between track and train is not static. It changes dynamically depending on the current state of your "railway-system". Easiest example: you cannot "hard code" the aspect of all signals into the XML-File. And this example applies to most of the other components as well.
And second, not only the information itself is relevant (WHAT is transmitted) but also HOW it is transmitted. Different trains and tracks use different systems. To check whether train and track are "compatible" you really need to store the kind of component that sends / receives the information.
Therefore, an abstract element like <ATPData> is not sufficient in my opinion.

> Here we arrive at the problem concerning
> the future applications using RailML. Simulation programs normally do not
> ask for technical components but are interested in the transferrable
> information. On the other hand applications for infrastructure planning
> demand the opposite.

Here you hit central question! I would like to make that question even more general: should RailML be used to give a physical or a logical / structural representation of the track? I know, I'm boring you since I posted a similar question some days ago. I'll try to explain my thought more precisely:

A physical representation contains the maximum amount of information. If you go to extremes, RailML could be a formal way to describe the original plans that were used for building the tracks including every trackside (ATP-)element.

If you choose a logical representation, you loose information. This information cannot be recalculated or rebuild from the remaining data. But in many cases, sooner or later you will discover or develop applications that demand some or all of the data that has been erased from your database. And then you are in trouble.

But from a representation, which is closer to the "real" physics, you can

derive the reduced logical information (the kind and amount of this information-subset may vary with your application) at any time. Of course I know that the extraction of the demanded data can be complex and expensive (expensive in both meanings of "computation power" and "man power"), but seen under a long-term point of view maybe it's worth it.

(Of course we cannot create an element for every single screw; we have to find a reasonable "granularity")

So this is why I'm always tending towards a more physical description, since this seems to be the more general approach in my opinion. But I'm new to the RailML-business... perhaps you have thought about this topic before and I simply don't know about other arguments.

> Maybe we can find a solution right in the middle
> between the technical and the informational way of representing the OCS.

Yes, I think that this is definitely an important question with a high priority - see my arguments above.

> So far for now - with best regards,

Yes, right! That's it for today and for this week. I will take a look at Matthias' new scheme on Monday and send a comment.

Wishing you a pleasant weekend,
Volker