Subject: Grouping of <ocp> elements in an <ocpGroup>
Posted by Christian Rahmig on Wed, 27 Jun 2012 15:07:48 GMT

Hello everyone,

in recent discussions the need for a grouped operation control point was expressed. This <ocpGroup> shall have all the attributes of an <ocp> element. Regarding the hierarchy, the attributes of the <ocpGroup> element shall be overwritten by the correspondent attributes of the <ocp> if existent.

Several <ocpGroup> elements are grouped together in a <ocpGroups> container, which is an element of the infrastructure base element. The proposed ocp grouping is documented in detail in trac ticket #153 [1].

Please feel free to comment on this first concept, which may be implemented with railML 2.2. Do you require attributes for an <ocpGroup> additional to the ones from the <ocp> element?

Any comments appreciated...

[1] https://trac.assembla.com/railML/ticket/153

Regards

--
Christian Rahmig
railML.infrastructure coordinator

Subject: Re: Grouping of <ocp> elements in an <ocpGroup>
Posted by Christian Rahmig on Sat, 01 Sep 2012 11:48:12 GMT

Hello everyone,

> in recent discussions the need for a grouped operation control point was
> expressed. This <ocpGroup> shall have all the attributes of an <ocp>
> element. Regarding the hierarchy, the attributes of the <ocpGroup>
> element shall be overwritten by the correspondent attributes of the
> <ocp> if existent.
>
> Several <ocpGroup> elements are grouped together in a <ocpGroups>
> container, which is an element of the infrastructure base element. The
> proposed ocp grouping is documented in detail in trac ticket #153 [1].
>
> Please feel free to comment on this first concept, which may be

> implemented with railML 2.2. Do you require attributes for an <ocpGroup>
> additional to the ones from the <ocp> element?
>
> Any comments appreciated...

As you can see from the forum posting in [1] the concept of grouping
<ocp> elements in a <ocpGroup> has been modified. Using the new optional
parameter "parentOcpRef" an ocp can refer to its parent ocp and thus
implicitely be part of a group. This solution has been implemented in
railML 2.2.

Regards

[1] http://www.railml.org/forum/ro/index.php?group=1&thread= 43&id=165

--
Christian Rahmig
railML.infrastructure coordinator