

---

Subject: Re: ocp's/stations and their properties

Posted by [Christian Rahmig](#) on Mon, 02 Jul 2012 05:05:25 GMT

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

---

Hello Dirk and anyone interested,

- > For my understanding of the concept of <ocpGroups>, please check whether
- > the following statements are right:
- >
- > - An <ocpGroup> cannot be referred to - all other RailML elements refer
- > to <ocps> but never to <ocpGroups>.

Well, for my understanding it should be possible to refer to an <ocpGroup> as well, because it has the same attributes like an <ocp> element.

- > - An <ocp> can belong to more than one <ocpGroup> (which means: can be
- > referred by more than one <ocpGroup>).

This is basically possible, but is that a realistic case?

- > - An <ocpGroup> can have all the attributes of an <ocp>. There is no
- > attribute of an <ocp> which wouldn't also be available as an attribute
- > of an <ocpGroup>.

Yes.

- > - An attribute defined at an <ocp> can be overwritten by a corresponding
- > <ocpGroup> (which means: by an <ocpGroup> which refers to that <ocp>).

I think the concept of overloading attribute values needs to be turned upside-down: The attributes defined in the <ocpGroup> should be overwritten by corresponding attributes from a referenced <ocp>.

- > - An attribute defined at an <ocpGroup> is valid for all its <ocps>.

Considering the above concept, this will be only correct if there is not a corresponding attribute in the <ocp> element itself.

- > [...]
- > I would prefer the other way 'round: An <ocp> refers to it's <ocpGroup>
- > (if there is one). If necessary, an <ocpGroup> can also name it's <ocps>
- > but this would mean redundancy (crossing links).

That is an interesting idea. If this bottom-up-referencing is better applicable for the <ocp> modeling, we should adapt our first idea.

- > Another question is: With <ocps> and <ocpGroups> having the same

- > attributes: Why don't we allow an <ocp> to refer to another <ocp>? An
- > <ocp> can act as an 'direct' ocp or as an <ocpGroup> (or both).

The idea of <ocpGroup> was supposed to follow the pattern of <track> and <trackGroup>. However, your question brings it to the point: Do we need an explicit <ocpGroup> or may we reference from one <ocp> to the next/parent <ocp>? What do others think about this question regarding their usage of <ocp>?

- > So, my suggestion would be:
- > - to define an optional attribute 'parentOcpRef' (=tGenericRef) of an
- > <ocp>.

Thank you, Dirk, for the approach.  
Best regards

---  
Christian Rahmig  
railML.infrastructure coordinator

---