
Subject: [railML 2] IS:005 - ocp/@parentOcpRef Semantic Constraint

Posted by [Milan Völke](#) on Mon, 25 Jul 2022 09:23:08 GMT

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

Hi all,

This post is to inform you about the approval of the semantic constraint IS:005 of the ocp by the timetable developer group as requested by the IS coordinator. It is a constraint that describes that if an ocp refers to a parent ocp via its attribute @parentOcpRef, then all the contents described in this "child" ocp overwrite those of the parent ocp. The timetable group discussed this and as a result approved this semantic.

This is the exact wording of the semantic constraint:

An <ocp> that refers to a parent <ocp> via an @parentOcpRef overwrites the attributes and elements of the parent <ocp> if explicitly defined. If an element is specified on an <ocp> that uses a @parentOcpRef any information provided with that element on a higher layer of the <ocp>-tree is overwritten. There is no merging of element-information from different levels. The same applies for attributes of <ocp>. For further information see example below.

With that we also tried to clarify some of the detail questions that arise when trying to implement this kind of semantics, i.e. what does it mean if a child ocp specified a propOperational/@trafficType while the parent defines the propOperational/@operationalType, are those two informations to be merged to describe the child or not. We agreed that information should not be merged in subelements of ocp between layers and also provided an example in the best practice section: https://wiki2.railml.org/wiki/IS:ocp#Overwriting_of_attributes.2Felements_in_lower_levels_of_an_.3Cocp.3Es_hierarchy

Please let us know if you have concerns or want to suggest how to improve documentation.

Best regards, Milan

Subject: Re: [railML 2] IS:005 - ocp/@parentOcpRef Semantic Constraint

Posted by [christian.rahmig](#) on Mon, 09 Jan 2023 15:10:20 GMT

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

Dear Milan, dear all,

since there has been no further feedback on this topic, I want to confirm the semantic rule discussed by the TT developers group: any information defined for an OCP having a parent OCP overwrites the related information of the parent OCP. At the same time, information provided only for the parent OCP is also valid for the inherited OCP when the information is not overwritten there.

Thank you for adding the best practice documentation for information inheritance at the OCP wiki page. I suggest to include it also in a general wiki page about inheritance.

Best regards
Christian

--

Christian Rahmig - Infrastructure scheme coordinator
railML.org e.V. (Registry of Associations: VR 5750)
Altplauen 19h; 01187 Dresden; Germany

Am 25.07.2022 um 11:23 schrieb Milan Wölke:

- > Hi all,
- >
- > This post is to inform you about the approval of the
- > semantic constraint IS:005 of the ocp by the timetable
- > developer group as requested by the IS coordinator. It is a
- > constraint that describes that if an ocp refers to a parent
- > ocp via its attribute @parentOcpRef, then all the contents
- > described in this "child" ocp overwrite those of the parent
- > ocp. The timetable group discussed this and as a result
- > approved this semantic. This is the exact wording of the semantic
- > constraint:
- >
- > An <ocp> that refers to a parent <ocp> via an @parentOcpRef
- > overwrites the attributes and elements of the parent <ocp>
- > if explicitly defined. If an element is specified on an
- > <ocp> that uses a @parentOcpRef any information provided
- > with that element on a higher layer of the <ocp>-tree is
- > overwritten. There is no merging of element-information from
- > different levels. The same applies for attributes of <ocp>.
- > For further information see example below.
- >
- > With that we also tried to clarify some of the detail
- > questions that arise when trying to implement this kind of
- > semantics, i.e. what does it mean if a child ocp specified a
- > propOperational/@trafficType while the parent defines the
- > propOperational/@operationalType, are those two informations
- > to be merged to describe the child or not. We agreed that
- > information should not be merged in subelements of ocp
- > between layers and also provided an example in the best
- > practice section:
- > https://wiki2.railml.org/wiki/IS:ocp#Overwriting_of_attributes.2Felements_in_lower_levels_of_an_.3Cocp.3Es_hierarchy
- >
- >
- > Please let us know if you have concerns or want to suggest

- > how to improve documentation.
 - >
 - > Best regards, Milan
-