
Subject: railML 2.3 infrastructure extension proposal - "change"

Posted by [Torben Brand](#) on Thu, 31 Aug 2017 11:02:20 GMT

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

For the use case "capacity planning" we need to define workflow states that are not fixed in time but in a work flow.

Sometimes changes to elements must be made visible. This can either be made through two files and creating a delta or by declaring the changes in the new suggested attribute @nor:change that can be placed under any element.

Enumeration values:

"none": No changes to the model from its last state

"new": a new element has been added to the model

"modified": values if an existing element have been changed

"deleted": an element has been removed from the model

If no change values is defined the default value is "none".

This feature is usually used in the signalling workflow with the colours:

New = green

Modified = blue

Deleted = red

None (everything else) = black

Placing the new attribute nor:change

Changes are placed on the elements that have a change. For changes that span an entire track the change is placed on the track and all elements on the track inherit the change. For changes that span more than one ocp the change is placed on the infrastructure element.

Subject: Re: railML 2.3 infrastructure extension proposal - "change"

Posted by [christian.rahmig](#) on Fri, 01 Sep 2017 13:06:45 GMT

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

Dear Torben,

the change topic is closely linked with your other post about the states, but let's have a look at the issue in detail:

Am 31.08.2017 um 13:02 schrieb Torben Brand:

- > For the use case "capacity planning" we need to define
- > workflow states that are not fixed in time but in a work
- > flow.
- > Sometimes changes to elements must be made visible. This can
- > either be made through two files and creating a delta or by
- > declaring the changes in the new suggested attribute
- > @nor:change that can be placed under any element.
- > Enumeration values:

- > "none": No changes to the model from its last state "new": a new element
- > has been added to the model
- > "modified": values if an existing element have been changed
- > "deleted": an element has been removed from the model
- >
- > If no change values is defined the default value is "none".

I conclude that there are three different types of changes and that each change can be clearly categorized into "new", "modified" and "deleted". If there are no changes, we don't need to provide a change information.

- > This feature is usually used in the signalling workflow with
- > the colours:
- > New = green
- > Modified = blue
- > Deleted = red
- > None (everything else) = black

The colors may be different in other countries (e.g. in Germany), but the general idea is the same: to see at first sight what has been changed.

- > Placing the new attribute `nor:change`
- > Changes are placed on the elements that have a change. For
- > changes that span an entire track the change is placed on
- > the track and all elements on the track inherit the change.
- > For changes that span more than one ocp the change is placed
- > on the infrastructure element.

I see a main challenge in using an attribute instead of a sub element, because the attribute limits the information down to one value (the element is "new", has been "modified", or "deleted"). A sub element would allow for a more complex approach, e.g. referencing the date when the element has been deleted. Further, the sub element `<change>` could be linked to specific parts of a `<track>` using `@from` and `@to` position attributes. Especially for long tracks, this spatial limitation of changes may help reading systems to be more specific in visualizing changes on the map.

Another advantage of having a `<change>` sub element instead of an attribute: you principally will be able to provide further information that allow a referencing between infrastructure elements from before and after the change. So far, your approach is only able to tell you "this element has been modified", but it is not clear what in detail has been modified since the original state of the element (before the change) is unknown.

Dear railML infrastructure power users, what is your opinion on the topic? Do we need `<change>` elements and `@change` attributes within railML

2.4 or are you happy to see such a major concept being implemented with railML v3? Further, would you like to add more information to the change?

Thank you very much for any feedback!

Best regards
Christian

--

Christian Rahmig - Infrastructure scheme coordinator
railML.org (Registry of Associations: VR 5750)
Phone Coordinator: +49 173 2714509; railML.org: +49 351 47582911
Altplauen 19h; 01187 Dresden; Germany www.railml.org

Subject: Re: railML 2.3 infrastructure extension proposal - "change"
Posted by [Thomas Nygreen](#) on Fri, 04 Dec 2020 14:59:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

For railML 3, the discussion has been continued in a new thread.

Christian's question about implementing this in railML 2 still remains.

Best regards,
Thomas
