Subject: Differences between <screenCoordinates> and <Infrastructure visualization> tags
Posted by Fabiana Diotallevi on Mon, 24 Jun 2019 10:34:48 GMT
View Forum Message <> Reply to Message

Dear all,
we are adapting our tool RaIL-AiD to import/export data in the new railML 3.1 format.

I have a question regarding the two different tags:

<infrastructureVisualizations>
<screenPositioningSystems>

My aim is to draw with RaIL-AiD a railway track-plan, then export it in railML 3.1 and then re-import it and visualize it.
Suppose that I have drawn the station, and that for each of my objects (i.e. signals, switches, balises, etc) I have defined a specific measure of the linear mileage in meters. I need to export, together with the linear coordinates of the objects, also the "screen" coordinates, because I would like to reload the railML file and see again on the screen the exact same drawing I had before the export.

My question is: where do the screen coordinates shall be saved in the railML file? In the <screenPositioningSystems> or in the <infrastructureVisualizations> tag?

In particular, I have a problem with the netElements, that are not simple "point-objects" defined by a couple of (x,y) coordinates, but they can be polygonal chains.
So far the only way I have found is to export them as a multi-point <linearElementProjection> in the <infrastructureVisualizations> tag.
The same method (i.e. export objects coordinates using the <infrastructureVisualizations> tag) could be applied to map all other objects (spot-like, line-like, area-like).

In this approach, what is the use of the <screenPositioningSystems> tag? Can I neglect it?

I hope to have explained myself,
thanks in advance,

Fabiana

Subject: Re: Differences between <screenCoordinates> and <Infrastructure visualization> tags
Posted by christian.rahmig on Fri, 05 Jul 2019 09:23:11 GMT
View Forum Message <> Reply to Message

Dear Fabiana,

<infrastructureVisualizations> contains all the projections of elements

for graphical visualization of infrastructure, e.g. points, polygones, etc. while <screenPositioningSystems> is just the container for storing the coordinate system's parameter, e.g. the maximum screen resolution.

So, you shall save the screen coordinates with the elements under <infrastructureVisualizations>.

The connection between both elements is given with the reference attribute <visualization>@positioningSystemRef. Therefore, you need to have both elements in your export file, but the coordinates themselves (x,y) are given in the sub elements of <visualization>.

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

Am 24.06.2019 um 12:34 schrieb Fabiana Diotallevi:
> Dear all,
> we are adapting our tool RaIL-AiD to import/export data in
> the new railML 3.1 format.
>
> I have a question regarding the two different tags:
> <infrastructureVisualizations>
> <screenPositioningSystems>
>
> My aim is to draw with RaIL-AiD a railway track-plan, then
> export it in railML 3.1 and then re-import it and visualize
> it. Suppose that I have drawn the station, and that for each of
> my objects (i.e. signals, switches, balises, etc) I have
> defined a specific measure of the linear mileage in meters.
> I need to export, together with the linear coordinates of
> the objects, also the "screen" coordinates, because I would
> like to reload the railML file and see again on the screen
> the exact same drawing I had before the export.
>
> My question is: where do the screen coordinates shall be
> saved in the railML file? In the <screenPositioningSystems>
> or in the <infrastructureVisualizations> tag?
>
> In particular, I have a problem with the netElements, that
> are not simple "point-objects" defined by a couple of (x,y)
> coordinates, but they can be polygonal chains.

> So far the only way I have found is to export them as a
> multi-point <linearElementProjection> in the
> <infrastructureVisualizations> tag.
> The same method (i.e. export objects coordinates using the
> <infrastructureVisualizations> tag) could be applied to map
> all other objects (spot-like, line-like, area-like).
>
> In this approach, what is the use of the
> <screenPositioningSystems> tag? Can I neglect it?
>
> I hope to have explained myself,
> thanks in advance,
>
> Fabiana
>