Subject: [railML3] Proposal for removing "any" elements
Posted by christian.rahmig on Tue, 22 Mar 2022 08:13:26 GMT
View Forum Message <> Reply to Message

Dear railML IS community,

At our last conference in Gothenburg, Sweden, our Common Coordinator raised the issue of changing the way extensions for railML can be created in railML3. Thomas presented a new approach there, which is, however, mutually exclusive with the current approach. He has created a forum post on this, which unfortunately has not received much attention so far:

 https://www.railml.org/forum/index.php?t=msg&th=638& start=0&

The railML.org coordinators think it is a good idea to use this new technique, as it allows, among other things, to validate documents with custom extensions of the respective railML interfaces. However, since such a change will mean that existing extensions will no longer work with a new railML version, we need your feedback to check whether the advantages we see in the new way are convincing to you as well. Please let us know what you think under the thread above.

Background:

Previously, the common practice was to provide extension points in the official railML schema. This allowed an XML that contained non-railML tags at such a point to still be considered valid railML. The disadvantage of this approach is that it is not possible to specify through the extension schema itself where these non-railML tags are allowed and where they are not. An extension that is intended to record opening hours for a station building could thus also be used to record opening hours for a circulation or for a train number. From a technical point of view, this does not make sense. With the extension points it is not possible to restrict this. With the newly proposed approach (see forum post) this would no longer be a problem. In addition, code generation tools could also be used to implement code for importing and exporting railML more efficiently.

We therefore propose to replace the extension points with railML 3.2 with the new inheritance method. Please let us know at the link above whether there are any professional/technical arguments against this from your point of view.

Best regards
Christian

Subject: Re: [railML3] Proposal for removing "any" elements
Posted by christian.rahmig on Tue, 14 Jun 2022 08:38:41 GMT
View Forum Message <> Reply to Message

Dear all,

as you may have noticed, railML 3.2 comes along with a new approach for modelling schema extensions based on xsi:type as introduced with the previous forum post. In order to better understand the usage of xsi:type the wiki page [1] has been set up to describe the extension

concept with a practical example.

[1] https://wiki3.railml.org/wiki/Dev:Using_xsi:type

Best regards
Christian

Subject: Re: [railML3] Proposal for removing "any" elements
Posted by Jörgen Strandberg   on Mon, 04 Jul 2022 12:16:13 GMT
View Forum Message <> Reply to Message

Hi,
The example of how to define and use the extension concept to represent data seems reasonable.

As a tool vendor, I still have a doubt about how to support the extension concept when parsing. This specifically when xsi:type defines an unknown subclass of a standard railML type.

The railML parser we are developing is generated by and based on the Eclipse Modeling Framework (EMF). It will not, out of the box, support unknown subclasses.
EMF provides a pattern for handling unknown elements/attributes, which could be a part of the solution. But I would rather not have to resort to it because of the additional implementation effort.

Is there already a standard mechanism or concept of XML parsers that can be used to ignore any unknown xsi:type values, but still read all attribute values of the standard railML type?

Subject: Re: [railML3] Proposal for removing "any" elements
Posted by Milan Wölke   on Mon, 04 Jul 2022 14:20:57 GMT
View Forum Message <> Reply to Message

Hi Jörgen,

sorry, but I havent had much experience with the EMF so far. So what you are saying is that the EMF will completely skip unknown subclasses of known ones if encountered during parsing? Do you evaluate a xsi:schemaLocation included in the document? Im asking because other generated parsers in the case you are describing fall back to the known base class.

Best regards, Milan

Subject: Re: [railML3] Proposal for removing "any" elements
Posted by christian.rahmig on Mon, 13 Feb 2023 08:39:20 GMT
View Forum Message <> Reply to Message

Dear Jorgen,

---

since there was no further feedback on the questions of Milan I assume that you have solved the problem with handling xsi:type extensions in EMF? If so, could you share your experiences with us so that we can learn something from this?

Thank you very much and best regards
Christian