Subject: [railML3] Additional Attributes for Revision Management
Posted by Karl-Friedemann Jerosch on Wed, 13 May 2020 14:25:27 GMT

View Forum Message <> Reply to Message

Dear all,

first let me introduce myself: I am Karl Jerosch and I am working in the ETCS trackside engineering department of Siemens Mobility Germany and I am participant of the railML workgroup "ETCS Track Net".

To improve the practical use of railML files as data exchange file,
the work group "ETCS Track Net" suggests to add the following 4 information to be implemented in railML 3.2:

1.) a new attribute <status> providing information of the quality status of a railML file with a closed value list:
draft/verified/released

Note: This attribute could be part of e.g. <metadata>.

2.) a new attribute <fileDocumentId> providing information of the document id of the railML file (as substitution group="any"):
e.g. "ID123467890-LineX-Station1/Station2-XYZrailways"

Note 1: This attribute could be part of e.g. <metadata>.

Note 2: The existing attribute <railML><metadata> @identifier is used for other purpose and should not be used to provide an fileDocumentId.

3.) a new attribute <fileVersion> providing the file version number of a railML file with values:
00.00, ..., 99.99

Note 1: This attribute could be part of e.g. <metadata>.

Note 2: There is an existing attribute <railML>@version, which provides the version of the used railML schema,
but an attribute version is missing, which provides the file version of a railML file.

4.) a new attribute <md5checksum> containing a checksum over all following file contents covering (at least) <common>, <infrastructure>, <interlocking>, <rollingstock> and <timetable> (if exisiting in the railML file),

and if possible also as many elements of <metadata>.

Note 1: This attribute could be part of e.g. <railML>.

Note 2: The calculation of the 128-bit hash-value of <md5checksum> shall follow the common known "message-digest-algorithm 5"
which is also used in various applications, for example to check software downloads from the internet.

Note 3: By the new attribute <md5checksum> it can be detected by software tools importing a railML file,
if the content was modified during the data exchange from the railML exporting tool to the railML importing tool.
This feature is important to ensure high data quality required for SIL 4 applications.

Does the community agree with the suggested extension of the data model in railML 3.2?

Background:
For example, in huge railway signalling projects with different construction stages,
it is necessary to exchange railML files several times describing the same railway topology area,
but with modifications in it according to the construction stages.
To avoid using the wrong file version, and to avoid unintended file modifications,
the railML scheme 3.x shall provide a modelling to store the suggested information above.

best regards

Karl Jerosch
Siemens Mobility GmbH
SMO RI ML PE ENG HW&SW

Subject: Re: [railML3] Additional Attributes for Revision Management
Posted by Henrik Roslund on Thu, 14 May 2020 08:19:52 GMT
View Forum Message <> Reply to Message

Hello,

Very good inputs, Karl.

I am in the opinion too that these four attributes shall be implemented in railML 3.2.
They are essential for versioning and data quality.

Kind regards
Henrik Roslund
Senior Consultant ETCS, MIRSE

Subject: Re: [railML3] Additional Attributes for Revision Management
Posted by   on Tue, 19 May 2020 11:30:17 GMT
View Forum Message <> Reply to Message

Dear Karl and all others,

I would welcome the suggestions for extending the metadata as suggested by you, Karl.

- I think they should be placed into <metadata> at top level, either in existing DublinCore (DC) data fields or by extensions of DC.
- I see no reason why this should not be allowed in railML 2.x versions, at least from railML 2.4/5. I would welcome it for railML 2.x as well.

However, it would not be a standard if we would not define the meaning/contents and usage of the new fields. The aim of the standard including the new fields should be that two software programs can exchange data without knowing each other.

2.) a new attribute <fileDocumentId>
So, concerning the new attribute @fileVersion, some contents rules should be defined (a file with a higher version number replaces a file with a lower version number etc.).

4.) new attribute <md5checksum>
Concerning the new checksum, I would recommend to exclude it from the railML file at all. Including it brings potential problems in defining which elements should be calculated into in detail (including potential spaces, line breaks etc. before/after?). This makes algorithms to calculate it rather difficult to implement.

Please be aware that railML recommends packing railML files (*.railmlx, see [1]). Therefore, we already have a checksum being part of the standard, albeit not an MD5 (but a CRC32). Better than nothing and should already allow recognising "arbitrary" data transfer errors. Which further use would an MD5 bring? Surely not a safety against deliberately manipulating the railML file because that could easily include the MD5.

However, if an MD5 shall be made part of the standard, I would prefer seeing it in one of the ZIP File Format Specification fields. This at least allows them to include all the railML file itself and being easily calculated and checked.

With best regards,
Dirk.

[1] https://wiki2.railml.org/index.php?title=Dev:fileConventions #Compressed_railML.C2.AE_files

Subject: Re: [railML3] Additional Attributes for Revision Management

Dear Mr. Jerosch, dear all,

let me first introduce myself: My name is Michael Gruschwitz and I will strengthen the team at Bahnkonzept in the railway IT area. At the moment I'm working on different projects and hope to be able to contribute and get help in railML matters in the future too.

Am 13.05.2020 um 16:25 schrieb Karl-Friedemann Jerosch:
> To improve the practical use of railML files as data
> exchange file,
> the work group "ETCS Track Net" suggests to add the
> following 4 information to be implemented in railML 3.2:

I think it's a good idea that when writing and reading our infrastructure data, it provides additional information that we previously had to maintain manually. We would very much welcome it if this information were included in railML 2.5 and from railML 3.2 onwards.

> 1.) a new attribute <status> providing information of the
> quality status of a railML file with a closed value list:
> draft/verified/released
>
> Note: This attribute could be part of e.g. <metadata>.

Agreed. But I would suggest to structure the information a bit more finely, because I think there will be more levels.
What about @status with:
 - stub: data which is only partial filled/incomplete
   - internal: data which is complete, but not internal checked and not released to a third party
 - draft: data which is complete, internal checked and therefore released to a third party
 - verified: data which is complete, internal checked and crosschecked by a third party
 - ....

But I assume that there are certainly already some standards or process descriptions for this, so that we do not have to reinvent the wheel.
Can railML check this?

> 2.) a new attribute <fileDocumentId> providing information
> of the document id of the railML file (as substitution
> group="any"):
> e.g. "ID123467890-LineX-Station1/Station2-XYZrailways"

Sounds good, but I think that the "talking ID's" will get no mercy from

the railML coordinators.

> 3.) a new attribute <fileVersion> providing the file version
> number of a railML file with values:
> 00.00, ..., 99.99

Fine with me, but should really only these kind of numbers be allowed?
What about "Alpha", "version 10.10 Yosemite", "version 10.14 Mojave", ...?

> 4.) a new attribute <md5checksum> containing a checksum over
> all following file contents covering (at least) <common>, <infrastructure>,
> <interlocking>, <rollingstock> and <timetable> (if exisiting
> in the railML file),
> and if possible also as many elements of <metadata>.

Even if we don't use it at the moment and therefore don't need it, it
would be a useful extension for the future.

Best regards,
--
Michael Gruschwitz
Bahnkonzept Dresden/Germany

Subject: Re: [railML3] Additional Attributes for Revision Management
Posted by Thomas Nygreen on Mon, 29 Jun 2020 23:59:10 GMT
View Forum Message <> Reply to Message

Dear all,

Apologies for my lack of reply so far!

I have noted the need (https://trac.railml.org/ticket/382), and we have discussed it briefly among
the coordinators. We are double-checking if any of them could/should already be covered by other
metadata fields.

I think it is difficult to implement a hashsum (such as md5) inside the file, as it creates a paradox:
it cannot be calculated before the file is generated, but it also has to be put into the DOM before
the file is generated. See Dirk's post for more pitfalls. I do like Dirk's suggestion for a convention
though. Putting an accompanying file into a zipped package containing the hash sums of the other
files is quite common in other situations.

Best regards,
Thomas