Subject: Re: More standard attributes for objects
Posted by Thomas Nygreen JBD on Wed, 06 Feb 2019 16:23:32 GMT
View Forum Message <> Reply to Message

Dear Claus,

Are these suggestions for railML2.x or railML3.x?

> Suggestion 1: id - unique within all contexts
>
> We suggest that the existing railML attribute 'id' is defined as it is
> at present, allowing the same id to be found in different railML models.
> However, its general usage shall be encouraged to be as follows: The id
> should be unique within all foreseeable contexts and generated once at
> the time the object was created.

railML2.x uses the XML Schema ID datatype, and IDREF for references to other objects. As a consequence it is not possible to reference an object that is not contained in the same file. The datatype also dictates that the id must be unique within the file and must start with a letter or an underscore. Apart from this, the writing system can decide freely on how to generate IDs.

railML3.x allows both XML Schema IDs/IDREFs and UUIDs for all id and reference attributes. This provides the possibility to identify and reference objects uniquely across files, contexts and time. How and when the UUIDs are generated is up to the owners of the data. railML3.x also allows separately listing multiple external identifiers and/or references for an object.

In my opinion the possibilities provided in railML3.x are sufficient. I think it is a good idea to encourage the use of UUIDs, but I do not think it can be made a requirement.

> We suggest that the following should always hold true for a well-formed
> railML file:
>
> a) A globally unique ID (guid) shall be computed and assigned to each
> object's id attribute at the time the object is created.[/quote]
>
> b) The id will never change.

This is the normal usage of UUIDs, but not for file-internal XML ids.
This is normally useful in detailed planning and asset management

> Suggestion 2: tag - unique within one administration's context
> (...)
> Suggestion 3: code - unique within the context of a given station and
> object type
> (...)
> Suggestion 4: name - an ornamented and context-dependent identification
> of an object

> (...)

To me, these suggestions seem to be a perfect match against the following already existing railML2.x attributes (with description from the wiki):

@code (for the suggested @tag): "Machine-interpretable string (e.g. an abbreviation) used for identification of the object across exchange partners, usecase specific uniqueness constraints may apply." In railML3.x one can use the <designator> subelement.

@name (for the suggested @code): Established, human-readable short string, giving the object a name. Usually not intended for machine interpretation. In railML3.x this is a repeatable <name> subelement, so that e.g. names in different languages can be provided.

@description (for the suggested @name): Human-readable, more detailed description as addition to the name. It should give additional explanations or hints to the contents of this data set. Usually not intended for machine interpretation. In railML3.x this is an attribute of the <name> subelement.