Subject: Required 'id' attributes (was: More detailed 'speed change' definitions)
Posted by Susanne Wunsch railML on Wed, 07 Nov 2012 13:25:11 GMT
View Forum Message <> Reply to Message

Hello Dirk and others,

The following aspect was quite overseen, sorry for that.

I changed the follow-up to railML.misc because of the more general
topic.

Dirk Bräuer <dirk.braeuer@irfp.de> writes:
> <speedChange id="sc3"...>
>
> Can we have the ‚id' of a speed change as an optional attribute? Since
> we have many, many speed changes and seldom the need to refer to them
> I do (optionally) want to save some space and effort.

* The "id" attribute provides good facilities to fetch single elements
(with its attributes and sub-elements) from an XML file irrespective
of an existing reference in the XML file.

* There are other often-used XML standards that use the same mechanism:

e.g. GML (Geographic Markup Language)

```
<attribute name="id" type="ID">
  <annotation>
    <documentation>The attribute gml:id supports provision of a handle
    for the XML element representing a GML Object. Its use is
    mandatory for all GML objects. It is of XML type ID, so is
    constrained to be unique in the XML document within which it
    occurs.</documentation>
  </annotation>
</attribute>

<complexType name="AbstractGMLType" abstract="true">
  <sequence>
    <group ref="gml:StandardObjectProperties"/>
  </sequence>
  <attribute ref="gml:id" use="required"/>
</complexType>
```

This attribute is required for all GML objects irrespective any
reference.

A huge amount of XML Schemas in the Geographic field of application

base on the above mentioned XML Schema.

* It is already a very smart implementation for identification.

  The following example shows up a typical alternative from the banking sector:

  <act:id type="guid">aa671349a0befafcfff1773e42eb8a49</act:id>

* There are algorithms for different programming languages for the ID-generation in general.

  e.g. generate-id() in XPath (staying at the XML-world) ;-)

* The _required_ attribute may only be changed into "optional" for the next major release.

I hope it clarifies the issue a bit.

Kind regards...
Susanne

Crosspost & Followup-To: railML.misc

--
Susanne Wunsch
Schema Coordinator: railML.common

Subject: Re: Required 'id' attributes (was: More detailed 'speed change' definitions)
Posted by on Thu, 08 Nov 2012 19:52:12 GMT
View Forum Message <> Reply to Message

Dear Susanne,

thank you for the reply.

Some of the aspects on 'id's in RailML

> * There are other often-used XML standards that use the same mechanism:

Yes, of course, all strict-relational data models do so.
(Strict-relational data models need to have Id's.)

In RailML, we also have some hierarchy (so RailML is not strict-relational) and, more important, we do not know about the data model of the reading/writing software. So I would say it is bad to force it to strict-relational data models.

Additionally, we do not have 'id's at each <elements>, so the selection of elements with id (out of all elements) is a rather arbitrary one. If your arguments would be accepted: Wouldn't we need to provide 'id's at all <elements>, would we?

So to come out of this current "arbitrariness" I thought about the rule "no 'id' without a 'ref' to it". So: If we need a reference, we have to provide both 'id' and 'ref'.

That would lead us to a rather logic structure: Hierarchy (<elements>) combined with non-hierarchic references with 'id's.

The question on the meaning of 'id's in RailML (as a primary key for relational data models) is a more general one - not specially on speed changes, so I will end this arguing here and open a new topic at "misc".

Best regards,
Dirk.

Subject: Re: Required 'id' attributes
Posted by Christian Rahmig on Fri, 09 Nov 2012 20:17:08 GMT
View Forum Message <> Reply to Message

Hello everyone,

> Additionally, we do not have 'id's at each <elements>, so the selection
> of elements with id (out of all elements) is a rather arbitrary one. If
> your arguments would be accepted: Wouldn't we need to provide 'id's at
> all <elements>, would we?

That's a good remark and actually I prefer defining IDs for all elements within a railML file instead of marking existing IDs as optional parameters. However, as Susanne already mentioned, this is a topic for railML 3.0.

Regards

--
Christian Rahmig
railML.infrastructure coordinator