
Subject: XSD design patterns (was: Re: railML 3.x: Data Modelling Patterns)
Posted by [Vasco Paul Kolmorgen](#) on Thu, 10 Jan 2019 10:39:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

I'm moving the topic of XSD design patterns from the thread "railML 3.x: Data Modelling Patterns" (https://www.railML.org/forum/index.php?t=msg&th=573&goto=2057&#msg_2057) into this new thread to make discussion better visible and understandable.

Am 28.12.2018 um 21:39 schrieb Thomas Nygreen:

- > The xsd design pattern was not mentioned in the
- > presentation, but I will include a comment here anyway.
- > Mostly, it currently looks like a Garden of Eden pattern,
- > but the local and global element names do not match. Mostly,
- > it is just a matter of capitalization, but there are also a
- > lot of cases where the difference is more substantial
- > (mostly in IL, but also in IS). And, as XML is
- > case-sensitive even differences in capitalization matter.
- > Also, there is no point in generating global elements for
- > types that are never used in any local elements such as
- > abstract types. The pattern should be to define elements
- > also globally, not to generate elements for all types.
- > Currently there are some global elements that cannot be used
- > in valid xml (because they implement abstract types), some
- > that should not be used (because their names are not similar
- > to any local elements), and no local elements are really
- > defined globally (case-sensitivity + other differences).

A compressed description of the individual modelling variants can be found on an Oracle page:
<https://www.oracle.com/technetwork/java/design-patterns-142138.html>

In today's conference call of the Timetable developers the topic was discussed. After a - still short - discussion of the topic, the following level of discussion is emerging:

- The railML 3 schema should be aligned with one of the patterns, but this should then be consistently adhered to.
- There is a preference for the variants "Garden of Eden" or "Venetian Blind", a preference will be communicated by the individual developers until January 16, 2019.

The consequence of the implementation of the proposals would be that the railML 3.1 RC 2 (expected to be published on January 29, 2019; see also <https://www.railml.org/en/public-relations/news/reader/delayed-3.1-release-and-updated-license.html>) will look significantly different, but the example file will remain

approximately the same.

We are very interested in further opinions and hints in the railML 3 context and railway area, but ask for quick feedback.

Thank you and kind regards,

--

Vasco Paul Kolmorgen - Governance Coordinator

railML.org (Registry of Associations: VR 5750)

Phone railML.org: +49 351 47582911

Altplauen 19h; 01187 Dresden; Germany www.railML.org
