Subject: Re: XSD design patterns (was: Re: railML 3.x: Data Modelling Patterns)
Posted by Thomas Nygreen JBD on Thu, 10 Jan 2019 18:04:33 GMT
View Forum Message <> Reply to Message

I find it useful to be able to create files that do not start at the railML root, and will still validate against the XSD and be read by a schema-aware parser. This is a benefit of using a pattern with global elements (such as Garden of Eden). Without global element definitions, one has to include all ancestor elements (which in turn can force inclusion of required subelements of these ancestors). However, even with global element definitions, such partial files require the use of UUIDs if referencing elements that are not included in the file.

A couple of use cases for such partial files:
Splitting data into separate files (e.g. separate IS, IL, TT, RS files). With global elements these files can start at their domain element, with a separate file starting at <railML> referencing them all by UUID. Without global elements, each file must start at <railML>, but nothing else changes.Sending an update for one specific object as a snippet. With global elements + UUIDs it is possible to create a valid file containing just the desired object(s). Without global elements this must be handled via methods outside of railML (e.g. diff files in version control systems).

Apart from the ability to start at any place in the tree, I do not really se an additional benefit in our case from global element definitions compared to just having global type definitions (Venetian Blind). Element definitions can easily be reused through reusing types.