Subject: Re: [railml3] Proposing new @epsgCode for <geometricPositioningSystem>
Posted by Thomas Nygreen on Thu, 09 Apr 2026 15:47:13 GMT
View Forum Message <> Reply to Message

Dear Mathias,

Of the concerns you raise in your first reply, I think the main one is the first: if we decide to replace crsDefinition with epsgCode it is exactly to restrict references to EPSG CRSs in a completely unambiguous manner, which of course removes the option to reference non-EPSG CRSs.

Regarding your other concerns:
- An int is not directly dereferenceable, but consider it equivalent to require a URI starting with "http://www.opengis.net/def/crs/EPSG/0/". Most valid URIs are also not dereferenceable, so unless we impose further restrictions than "any URI", this concern will remain.
- The data consumer does *not* have to assume the axis ordering of an EPSG CRS. It is given in the CRS definition that is as uniquely identified with an int as with an opengis URI. OGC:CRS84 is not the same CRS as EPSG:4326, because the axes are switched, and their definitions are therefore not the same.

I agree that this would introduce one more difference between the railML and GML modelling, and we will need a best practice for srsName in gmlLocation no matter what we do with crsDefinition (because we cannot change the syntax of GML). From OGC, the GML recommendation has for quite many years been to use a dereferenceable URI/IRI, but for "well-known references it is not required" [GML 3.2 specs], allowing for things like the short strings mentioned above. As far as I know, the definition of "well-known" is left to the exchanging parties. Before recommending URIs, the recommendation was to use URNs. OGC no longer operates the URN resolver, but they can easily be translated to resolvable URIs.

I agree that the flexibility to use non-EPSG CRSs was probably intended. And this is not the only issue where we see that originally intended flexibility when exporting creates too much ambiguity or too many cases that importers have to prepare for.

You write that you would like a more strict value space for crsDefinition. You also write that you would allow any URI. In XML Schema Definition anyURI also allows relative URI references, allowing basically any string, including things like "WGS84", "4326" or "EPSG:4326" (which could in theory resolve in the unlikely case that a resource with that name is available in the same directory as the railML file), or a URN (which requires a dedicated resolver).

Do you have any suggestions on how you would restrict the value space of crsDefinition, syntactically or by best practice?

Best regards,
Thomas