Subject: Re: [railml3] Proposing new @epsgCode for <geometricPositioningSystem>
Posted by Mathias Vanden Auweele on Sun, 22 Feb 2026 10:52:11 GMT
View Forum Message <> Reply to Message

Hello Marharyta,

I'm not sure I think it's a good idea to go from @crsDefinition to @epsgCode because of the loss of functionality:
- no possibility to reference non-epsg CRS (probably not a very big problem but I was planning on using this in the future for a more schematic representation of the network)
- no longer dereferenceable, I really like the use of URI's instead of ints and strings... Makes it much easier to really understand what you are reading
- axis order... just the EPSG will need the data consumer to assume axis ordering.. OGC:CRS84 (lon/lat) vs EPSG:4326 (lat/lon). Although there is a lot of confusion about this and some implementations are not doing it correctly
- from memory, there is also something about the vertical axis

I did some research about this a while ago and my conclusion is to prefer URI's. It also seems strange to have GML on the one side that uses srsName and then a simple epsgcode on the other side.

I don't think I would agree with this statement: "This proposal is based on the idea that the EPSG code was intended by the previous crsDefinition". I know several people that have developed RTM and I don't think that they would have chosen crsDefinition if they meant to use EPSG code. So for me, that's really intentional.

Although I would very much agree on a more strick set of possible values for the crsDefinition. I would force the use of URI's and disallow the use of strings such as EPSG:4326