

Support of multiple measurement units in VSS

2023-11-21

Problem

- VSS currently defines one “default” unit of measurement for signals
 - Mostly, but not always, VSS favours metric
- However, the default does not always suit everyone
 - Example: Some countries regulate on speed in mph, others in km/h
- Result is that the default measurement unit needs changed by an implementor to suit the environment/location of where a system is intended to be used
 - This is allowed, per VSS
 - However, this then requires tailoring of applications for those systems so that they have prior knowledge that a signal value returned is different to that specified in VSS

Goal

- Provide greater visibility in VSS as to what measurement unit is used by a provided signal value
 - Thus, rather than choose “one side of the coin” in deciding a default value, enable multiple by providing an indication
- Note: We should **not** choose multiple default values per single signal
 - This would introduce other issues and ultimately break VSS

Solution

- Provide multiple branches to take into account measurement units used in the real world i.e. those that have a significant market need
 - Example: Vehicle.Speed
 - Could be expanded to the following:
 - Vehicle.Speed.Mph – Provide vehicle speed in miles per hour
 - Vehicle.Speed.Kmh – Provide vehicle speed in kilometres per hour
 - Significant market need:
 - Most European countries, Canada, Australia, and other countries require vehicle speed be measured in km/h while USA, UK, most of the Caribbean require mph.
 - No significant market need for any other measurement unit for road vehicles e.g. knots, m/s, lightyears, etc. 😊

Benefits

- VSS can consider all market needs, rather than having to be selective
- Applications can be developed from outset to accommodate VSS-specified measurement units
 - As opposed to having to be tailored to “know” what the returned measurement unit is on a per system basis
 - Implementors may of course support only Vehicle.Speed.Mph and not Vehicle.Speed.Kmh (or vice versa) but at least applications can detect this and still “know” how to handle the response
 - For Vehicle.Speed, preconfigured knowledge of the current system is needed, which requires per-system manual tailoring
- Ease developer burden for per-system tailoring of applications

Summary

- Support of multiple measurement units via multiple signal branches solves the issue of a chosen default measurement unit being unsuitable for significant markets
- The Vehicle.Speed signal is one particular use case, others may exist
 - E.g. tire pressures (PSI, bar, kPa), vehicle height (feet and inches, metres),...
- Eases developer burden regarding per-system tailoring
- Need to be conservative on number of units supported
 - VSS should specify quantities only where there is significant market need
 - Implementors can always add additional ones via overlays
 - We could set a requirement/guideline for ourselves e.g. maximum of 3