DOI 10.5162/ettc2020/2.4

A standard proposal for a sensor schema. SensorDefML.

José Antonio González Pastrana, Sergio Sanchez-Montes, Roberto Paredes,
Airbus Defence and Space, Getafe-Spain.
Jose.o.gonzalez@airbus.com, Sergio.m.montes.external@airbus.com,
Roberto.paredes.external@airbus.com

Abstract:
As in other areas of Instrumentation, xml files are used for defining the characteristics of different elements, in this case we are going to focus on instrumented sensors which are not already covered by existing specific standards such as XidML or MDL files. Our intention is a new proposal that could drive the industry to create a new standard that can help with some instrumentation tasks related with these elements, specially software configuration and global configuration management. This would be alike a sensors datasheet but in a xml file.

Key words: MDL [1] , XML [2], Open Standards, XidML [3], XdefML[4].

Introduction

Revising the existing schemas which could represent a sensor we can find one stablished schema called SensorML, but it is used in the same way as XidML or MDL, because it understands a sensor as a system to distribute data instead of representing its internal features and configuration options, but what we want to propose it is to represent the datasheet of a basic sensor like accelerometers, gauges, etc.

Similarly than standards for transducers, known as TEDS (Transducer Electronic Data Sheet), which are focussed just in hardware settings, we need a standard for covering software settings and configuration.

The main objective is to involve sensor manufacturers so maybe they could provide in the future this SensorDefML files the same way they supply their datasheets.

What fields should show SensorDefML?

SensorDefML is a metamodel that allow us to define the characteristics of a sensor. So if we have a magnitude that we want to measure, this it will be our input. We also can have additional inputs to take into account. Generally, the expected output should be a voltage or a current, but could be any other magnitude. This demonstrates that the schema has to be flexible enough to cope all the variety we find in the market. Meanwhile a minimum of characteristics have to be taken into account.:

- Identification.
  - Manufacturer.
  - Model Identification.
  - Part Number.
  - Ordering notes.
  - Local supplier address.
  - Operating principle.
  - Application field.

- Mechanical Characteristics:
  - Weight.
  - Dimensions.
  - Mounting instructions.
  - Drawing files.

- Input/Output (Channel for)
  - Magnitude (Temperature, Pressure, Voltage, Current, etc.), Power
  - Units.
  - Range / Span.
  - Impedance.
  - Bandwidth.

  - Grounding.
  - Noise.

- Metrologic characteristics:
  - Uncertainty.
  - Accuracy.(Class).
  - Dead zone / band.
  - Sensitivity.
  - Cross channel sensitivity.
  - Repeatability.
  - Hysteresis.
  - Bias.
  - Scale type.
  - Damping
  - Calibration formula.
  - Formula parameters.
  - Errors (sources of error).

- Operating conditions
  - Temperature range.
  - Storage temperature.
  - Pressure. Static/Dynamic.
  - Humidity range.
  - Warm-up time.
  - Vibration range.
  - Maximum acceleration.
  - Impact acceleration.
  - MACH number limit.
  - Type of standard.
  - Accomplished Standards.

All of these characteristics are examples of what has to be taken into account to help us to create a first version of the schema. Obviously a lot of different characteristics can appear later on as more sensor classes start adopting the SensorDefML. For that reason SensorDefML has to be as much flexible and scalable as possible, not only for supporting new and different sensors but also new features or settings without changing the defined internal file structure.

Working with SensorDefML SDK

In order to have an agile way of working and also prevent recurrent errors it is helpful to have a SDK[5] making easier some commonly used methods such as create, read and write SensorDefML files. This SDK should have functions to extract all necessary information in order to fill-in data in the memory data models and/or in databases that the customer could need.

Tools based in SensorDefML

Two tools are planned to use this schema.

One of them is an Inventory Control System (ICS) which manages all the hardware related elements belonging to a system. The other one is a Systems Designer (SD), a tool for designing the complete architecture of a system, taking into account all the different manufacturers and full integrated with the ICS where the hardware items are available to be placed in an architecture.

The ICS needs to persist new sensors, so the best way to do this task is by importing a SensorDefML file which contains all the necessary information to fulfill the tool's requirements, such as quality politics and others.

The SD can take advantage of SensorDefML files to configure the complete measuring chain, check specified ranges and ensure that every sensor fits in the system. The calibration of the sensor can be added afterwards, but most of configuration will be valid, so a great deal of work can be done from the beginning. Also, as the SD is connected to the ICS database you have available not only the sensor models that you can use, but you can have the real serial number you will use from the beginning.

Conclusions

SensorDefML files intends to be a key technology for the ICS, SD and metrology in ADS [6]. This kind of schema could become a standard in the future. As a standard, their definition should be agreed between all stake holders in this field. So all of them are invited to join this initiative.

References

[1] MDL -http://www.irig106.org/docs/106-17/Chapter23.pdf

[2] XML - https://www.w3.org/XML

[3] XidML - http://www.xidml.org

[4] XdefML - http://www.xidml.org

[5] SDK – Software Development Kit

[6] ADS – Airbus Defense and Space