Versioning and Features

Context: One Data Model (OneDM) evolution
(See also previous talk)

— Model versioning
— Language versioning

What is a feature?

- Capability indication
  - unidirectional announcement
  - no consequence if ignored by peer
- Interpretation agreement
  - shared setting, must be heeded by peer
  - result of negotiation/common context
  - need to prevent false interoperability

What is a "version"?

Feature that has been

— blessed a bit,
— named by straightjacketing into a linear space

Can be roll-up of both:
capability and interpretation features

Imagine: feature string "SDF2.0"

Versions vs. Revisions

Revisions of a file need not have semantic implications
(e.g., updating copyright messages)

N.B.:
"fixing typos" often does have a semantic implication
(cf. "writable")

Model features/versions

Availability of a feature can be intended as a capability indication
Distinguish model capability from implementation capability

Also: roll-ups

Language features/versions

Unidirectional interpretation requirement

Can be added to header as `"require": [...]`
Needs stability of overall (data level) syntax

(Being based on JSON reduces issue of surface syntax evolution)

Might have `"optional"` as well, for features that can be ignored (e.g., protocol bindings)

Numbers and Code Point spaces

Numbers/Codepoint spaces could be used

— By a single ecosystem: qualified numbers (e.g., IPSO objects, even YANG-CBOR SIDs)

— Across ecosystems: global numbers

Note that global strings are not much different from global numbers: Both need to be agreed

Define the ecosystem numbers where?

Putting numbers into the common models?

— Great to have all info in one place

— Adding an ecosystem requires changing the model; merging problem

Putting numbers into separate files?

— Needs hooks in model to tie these numbers to

In other news

https://datatracker.ietf.org/doc/draft-bormann-core-senml-versions/