ObsCore & extensions from a model perspective

IVOA InterOp 2025a

Intro

Current ObsCore Status

The Data Model Process - my take, there are many others to chose from

Some History/Experiences

ObsCore Status

https://github.com/ivoa-std/ObsCore

REC-ObsCore-1.1 ported to ivoatex

- table formatting needs work
- bibliography needs to be checked
- internal cross-references need to be fixed

Almost ready to embark on WD-ObsCore-1.2

Data Model Process

Use Cases for ObsCore:

- data discovery
- data discovery in TAP
- data discovery in S(imple) protocols

The original TAP and VOTable underpinnings mean this is *not really data model* - it is a model of a query result

Concepts for ObsCore-1.0 did come from DMs (Characterisation, CAOM, …)

Data Model Process

How sparse?

- ObsCore: less sparse
- CAOM: more use cases, more fields, more sparse

How flexible?

- content: quantities, reference frames, etc
- structure: extensions, redundancy

How structured?

- ObsCore: flat, denormalised (because query result)

Experience

The ObsCore Cube extension (~2015)

- use cases from the CSP priority to provide standards for cube access
- a range of metadata items were proposed
- further analysis revealed some core features
- update: s_xel1, s_xel2, em_xel, t_xel, pol_xel (dimensions) added to ObsCore
- degenerate cases handled with no magic (e.g. em_xel = 1 for a 2d image)

The extension that was actually core…

Experience

CAOM version 1 was a *core + extensions* model

- much less sparse
- archive-specific metadata: flexible structure
- less motivation to find common ground, use common terminology
- cross-domain discovery more limited
- users face barriers to use extension metadata
- software complexity++

CAOM version 2: no extensions, evolve to support new use cases, accept sparseness

ObsCore is a model of a query result

… extensions are the result of more complex queries

How sparse?

How flexible?

How structured?