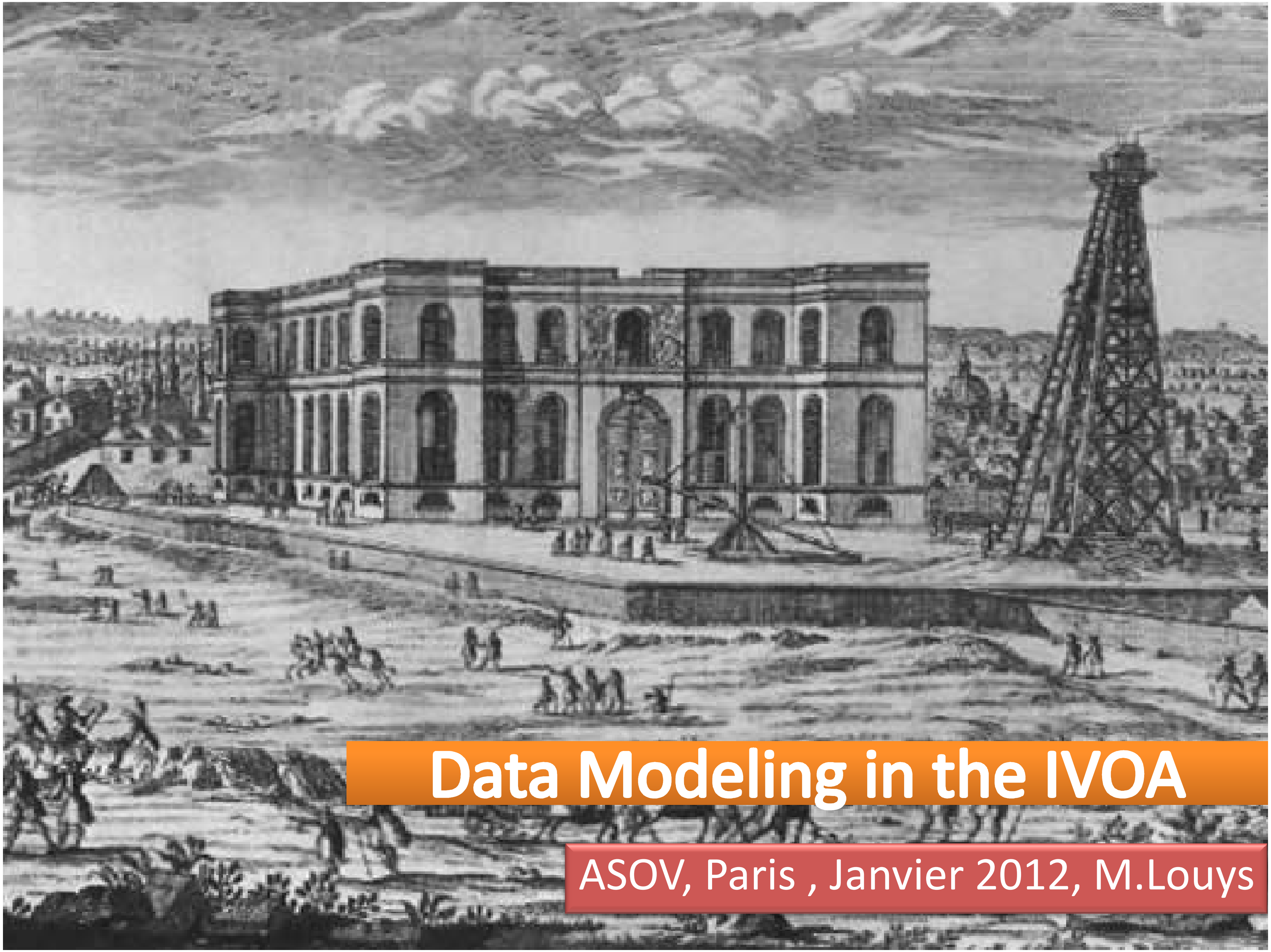


Data Modeling in the IVOA

ASOV, Paris , Janvier 2012, M.Louys

Goal of data modeling

- Describe metadata for astronomical data sets observed or simulated
- Stabilise a common language
  - that federates concepts in the astronomical community (+ or – implicit)
  - Propose reusable concepts for various wavelength/frequency domains
- Describe data products for scientific use-cases

History (2003 -2010)

- Sketching out the landscape
  - an overview of necessary metadata
  - Split simulation and observed data
  - First focus on data product (management), physical properties, and coordinates representation
    - Characterisation DM
    - Space Time Coordinates
    - VOResource
- Describe simple data products and dedicated metadata
  - Spectrum DM (distributed via SSA protocol)
  - Simple spectral line (distributed via SLAP protocol)
- Generalise the Observation Concept
  - Data discovery focused: Observation Core Components DM
  - Propose a common service for all archive data

History (2010 -)

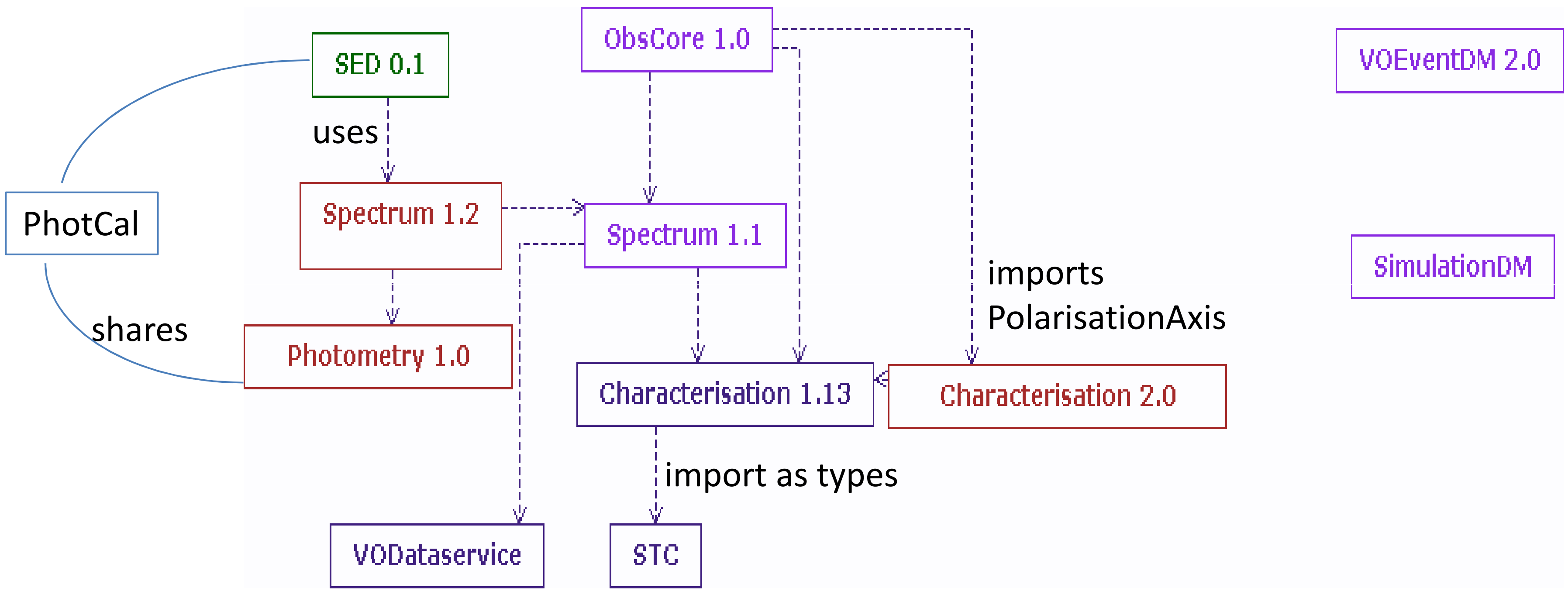
- Complete the coverage of data properties used for data analysis use-cases
  - Metadata for Photometry
    - Photometric calibration
    - Photometry DM
  - Spectral Energy Distribution (SED DM)
  - TimeSeries

- Articulate all data models in a comprehensive and consistent "patchwork"
  - Unique and re-usable definition of concepts
    - No overlap , no redundancy but re-usability
  - Any data product may be described "at minima"

Data Modeling approach

- Focus
  - Identify main concepts dealing with astronomical objects and signals properties
- Re-use core definitions
  - STC, VOResource, Characterisation
- Allow for evolution and extension
  - Classes can be re-used, enriched, bound together if needed (fine grain descriptions)
    - Characterisation Data Model v2.0 : calibration, variation maps, polarimetry, etc.

Data models dependencies



Data Modeling IVOA rules

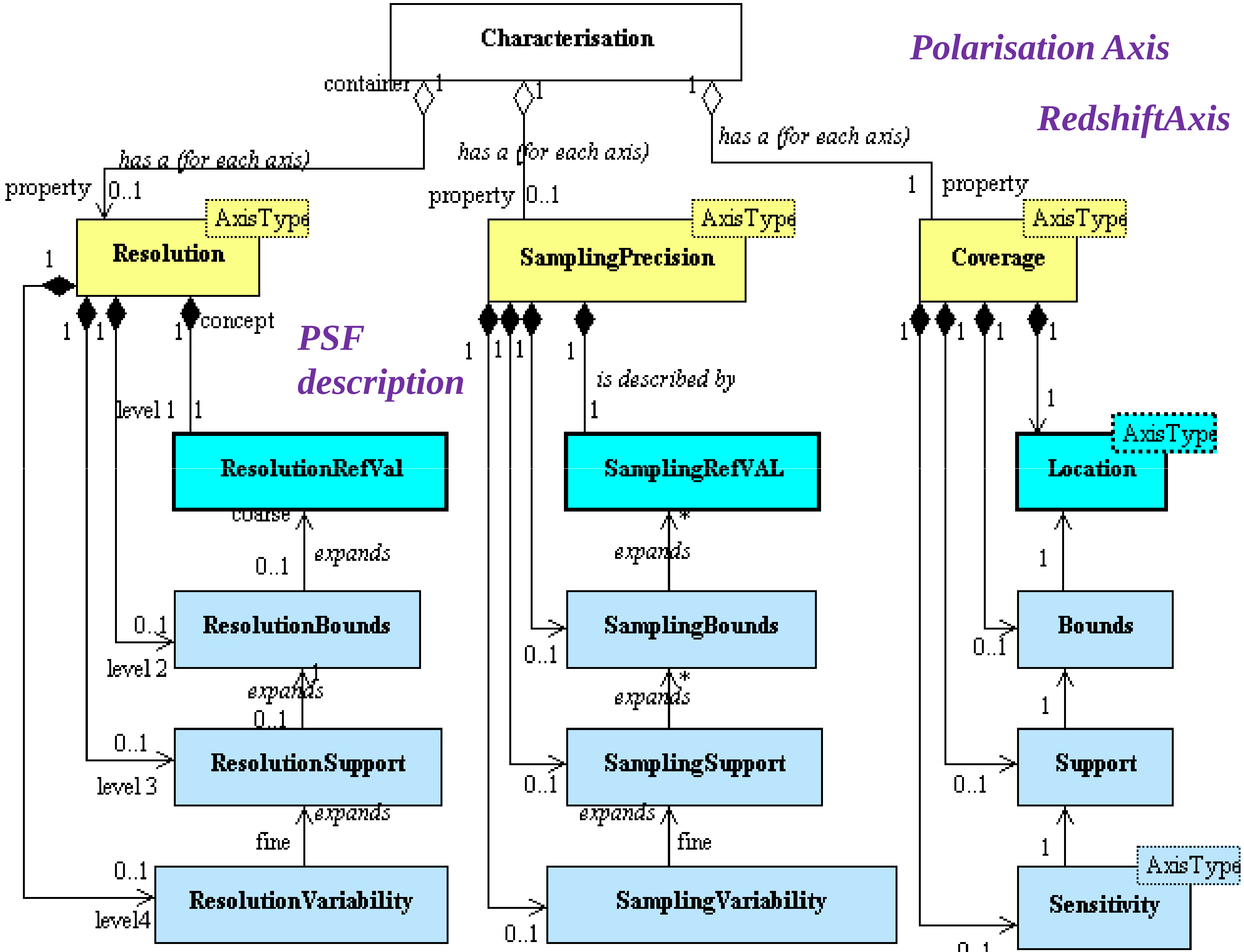
- Publish a Recommended IVOA document
  - Identify and describe main concepts dealing with astronomical datasets properties
  - Sketch out the logical bindings in a UML class diagram
- List and describe independent reference implementations
  - At least two, with link access and documentation
- Formalise how to serialise the data model
  - XML schema describing all possible data products
  - Utypes list derived from the model
  - Serialisation examples
    - XML instance documents
    - VOTable, FITS, (utype,value) list

CHARACTERISATION v2.0 DM

Segment vs Global Characterisation

Polarisation Axis

RedshiftAxis



PSF description

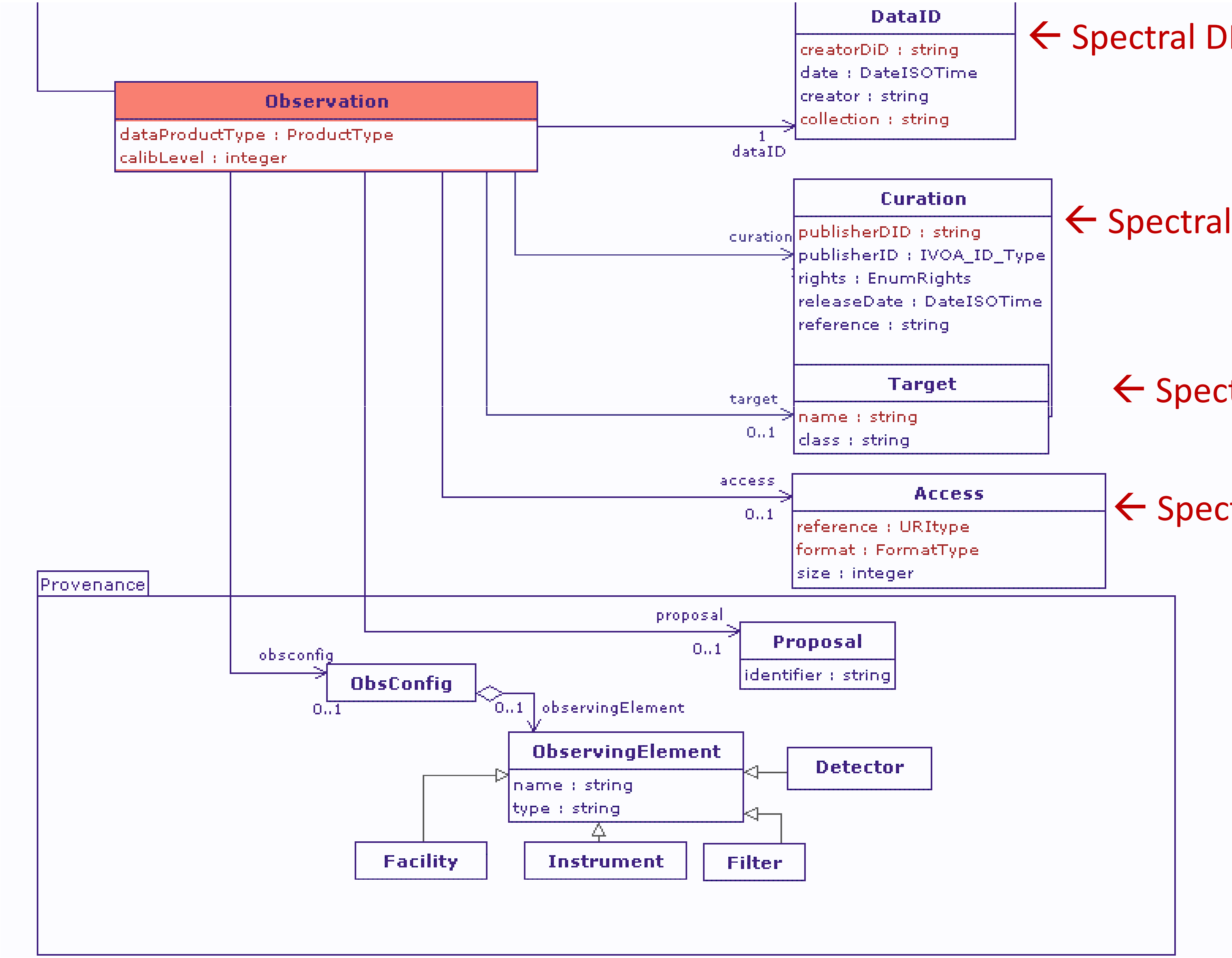
Functions

Maps

Access parameters

Obscore v1.0

Re-use



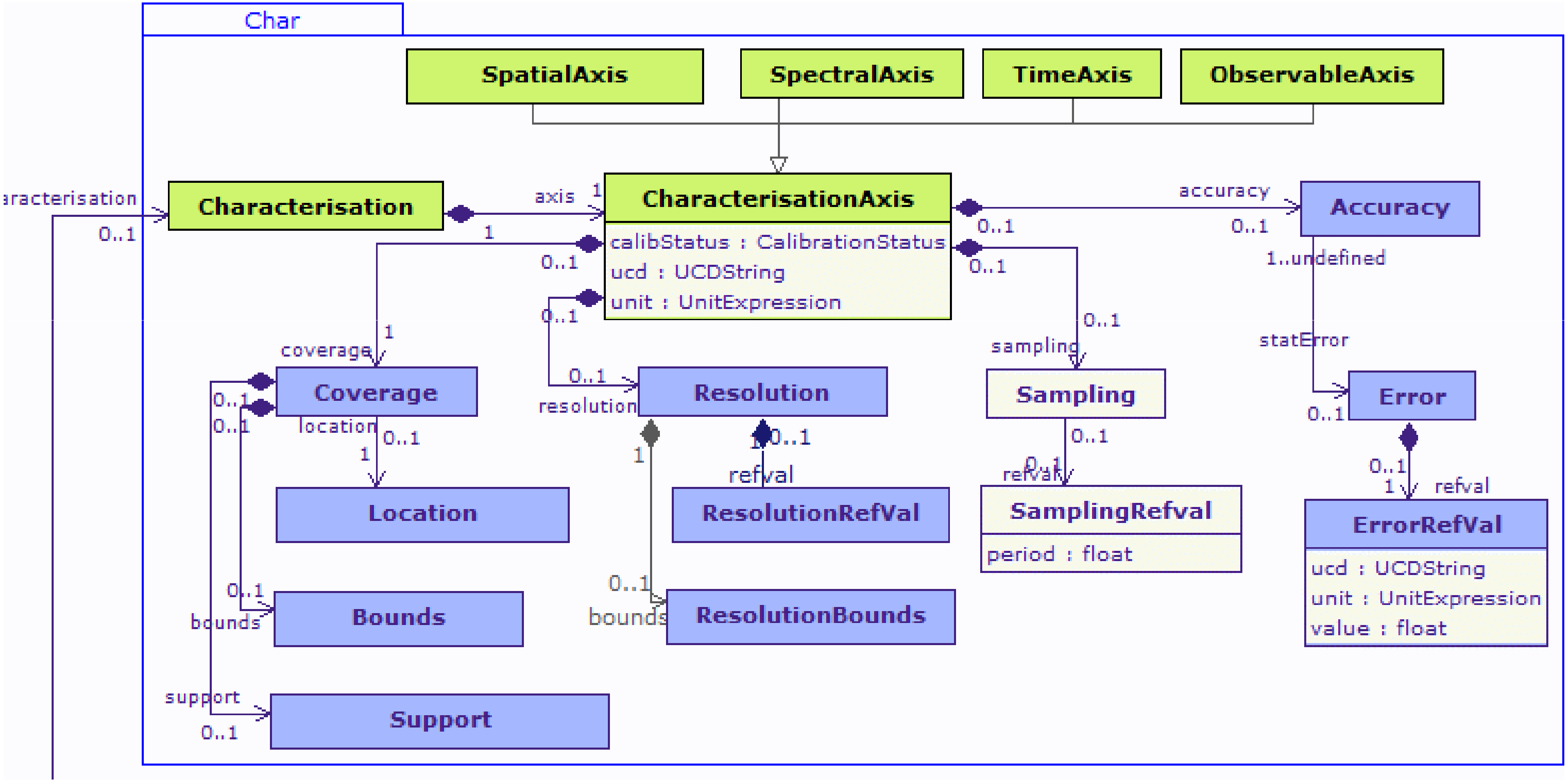
← Spectral DM

← Spectral DM

← Spectral DM

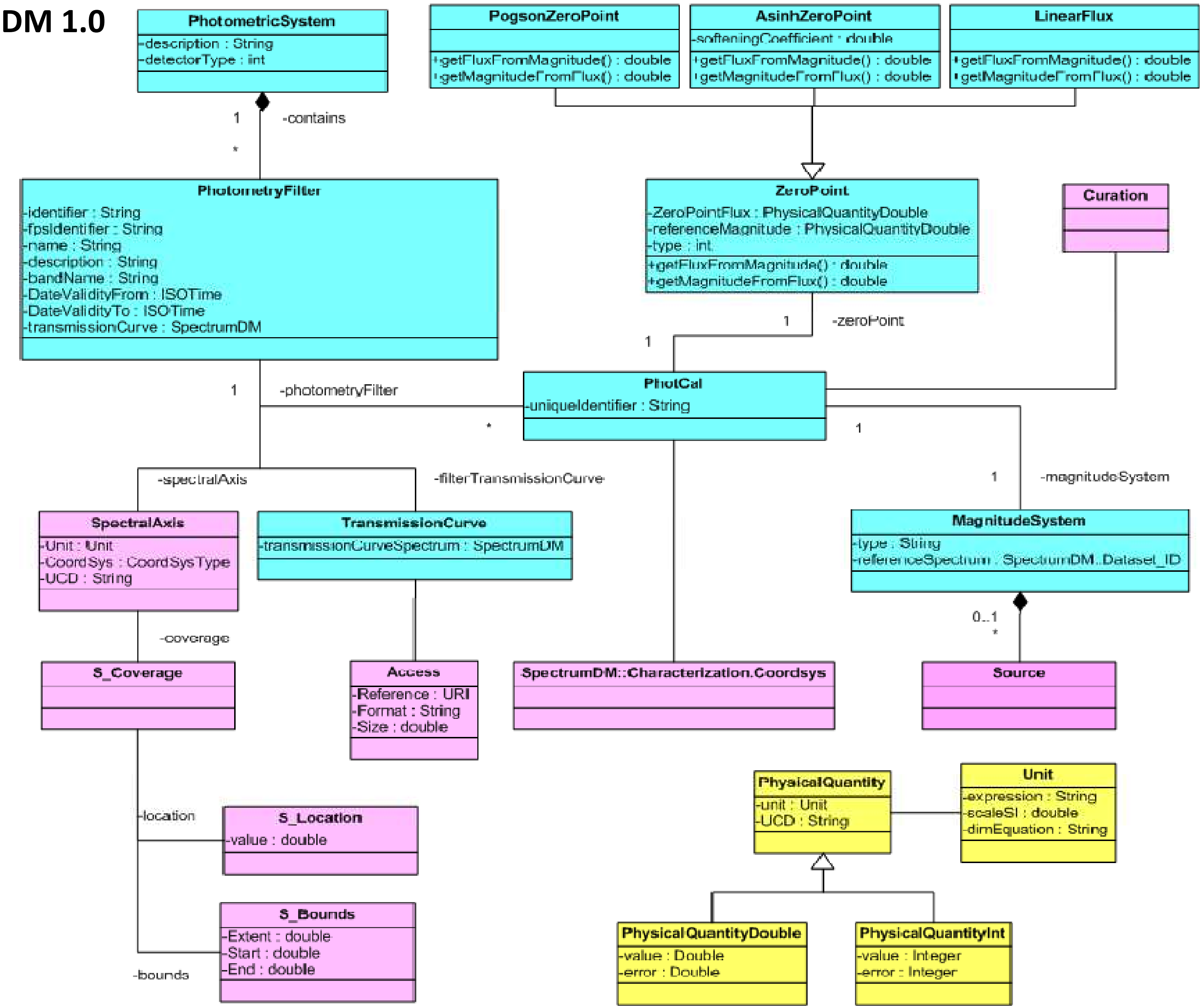
← Spectral DM

Re-uses of Characterisation DM



Colored boxes reused in ObsCore v1.0

PHOT DM 1.0



PhotometricSystem
-description : String
-detectorType : int
1
-contains
*
PogsonZeroPoint
+getFluxFromMagnitude() : double
+getMagnitudeFromFlux() : double
AsinhZeroPoint
-softeningCoefficient : double
+getFluxFromMagnitude() : double
+getMagnitudeFromFlux() : double
LinearFlux
+getFluxFromMagnitude() : double
+getMagnitudeFromFlux() : double
PhotometryFilter
-identifier : String
-fpsIdentifier : String
-name : String
-description : String
-bandName : String
-DateValidityFrom : ISOTime
-DateValidityTo : ISOTime
-transmissionCurve : SpectrumDM
ZeroPoint
-ZeroPointFlux : PhysicalQuantityDouble
-referenceMagnitude : PhysicalQuantityDouble
-type : int
+getFluxFromMagnitude() : double
+getMagnitudeFromFlux() : double
Curation
1
-zeroPoint
1
1
-photometryFilter
PhotCal
-uniqueIdentifier : String
*
1
-spectralAxis
-filterTransmissionCurve
1
-magnitudeSystem
SpectralAxis
-Unit : Unit
-CoordSys : CoordSysType
-UCD : String
TransmissionCurve
-transmissionCurveSpectrum : SpectrumDM
MagnitudeSystem
-type : String
-referenceSpectrum : SpectrumDM..Dataset_ID
-coverage
0..1
*
S_Coverage
Access
-Reference : URI
-Format : String
-Size : double
SpectrumDM::Characterization.Coordsys
Source
PhysicalQuantity
-unit : Unit
-UCD : String
Unit
-expression : String
-scaleSI : double
-dimEquation : String
-location
S_Location
-value : double
S_Bounds
-Extent : double
-Start : double
-End : double
-bounds
PhysicalQuantityDouble
-value : Double
-error : Double
PhysicalQuantityInt
-value : Integer
-error : Integer

Utypes

- Role : To identify a quantity in the context of a data model.
  - **phot:PhotometryFilter.TransmissionCurve**
  - **phot:PhotCal.ZeroPoint.Flux.value**
- Used in IVOA services
  - VOTable serialization
  - TAP services to describe items used in queries via TAP_SCHEMA
  - Useful for list of (utype, value) pairs for simple applications
- WD draft currently revised
  - Use-cases clearly identified and sorted out
  - Example of Utypes usage in various cases
  - Definition of a semi-automatic generation framework to derive Utypes, XML Schema, documentation, from UML class diagrams (Theory IG effort)
  - Proposition of a publication mechanism for Utype together with the DM specification

Data models in action

Purpose	DM name	Protocols	Serialisation
Data discovery			
	ObsCore1.0	TAP TAPschema	VOTable
TapHandle	ObsCore1.0	TAP/QL	VOTable
Application			
SED extended Object extraction	Characterisation v1.13		XML
VOSpec	Spectrum DM	SSA	VOTable
Aladin	Characterisationv1.13	SIA	VOTable
Spectral fitting tool	Characterisation v2.0	TAP	VOTable
PDR, Starformat, Deuvo, Millenium	SimDM	SimDAL	XML

Conclusion

- Le cadre de travail est stable mais non figé
- Nombreux concepts généraux et re-utilisables
- Evolutions et compléments possibles

  Etendre les modèles à des données plus spécifiques

  – Interférométrie optique?
  – Exprimez vos besoins …