VIRTUAL ASTRONOMICAL OBSERVATORY

Image Data Model – Heidelberg

Doug Tody, NRAO, USVAO

Image Data Model

- Two Talks
  - DT
    - Use cases, DM overview, key issues, image cubes
  - FB
    - More on DM, Mapping / WCS / STC

Image Data Model

- Use Cases
  - DAL Protocols
    - SIAV2, TAP/ObsTAP, potentially others
    - Archive Metadata to support these
  - 2-D Astronomical Images
    - Relatively straightforward, but important!
  - Image Cube Data (N-d)
    - Large cubes have some new issues
    - Astronomical N-d Image model (FITS) only partially supports these
  - Hypercube Data
    - Not really addressed by ImageDM, but related

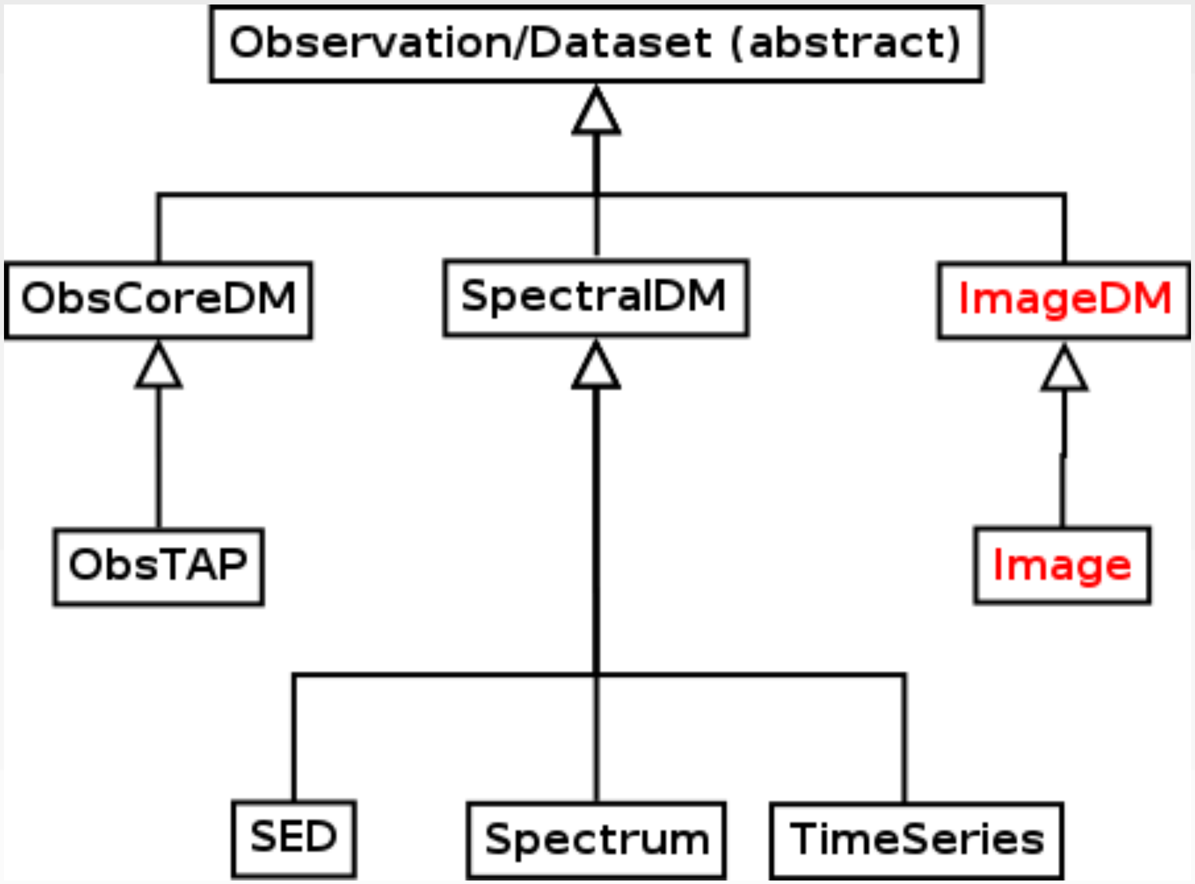
Image Data Model

- Astronomical Image Concept
  - Introduced by FITS late 1970's (Wells&Greisen, 1981)
    - N-d numerical data array with associated metadata (N = 1-4, maybe more)
    - M-d world coordinate system (WCS), maps pixel/voxel to physical coordinates
    - STC is related, but not quite the same thing (FB talk)
  - Astronomy (outside VO) regards this as THE Image data model

Image Data Model

- Why not just use FITS?
  - We do actually; adopted as a core VO technology
    - Used for binary data; to format image datasets (and binary tables)
  - Incorporates some important astronomical data models
    - Astronomical N-d image with associated WCS
  - But has some key issues
    - Data models not described as abstractions, separate from serialization
      - FITS 80-char card image, 8 char keyword serialization is obsolete
      - Newer serializations are more sophisticated
        - - HDF5, CASA image tables, JPEG2000, etc. (also VOTable for queries)
    - VO metadata, data models are much richer, must be supported by ImageDM
    - FITS serialization breaks down for very large cubes

ImageDM – Relation to Other Models



ImageDM Classes

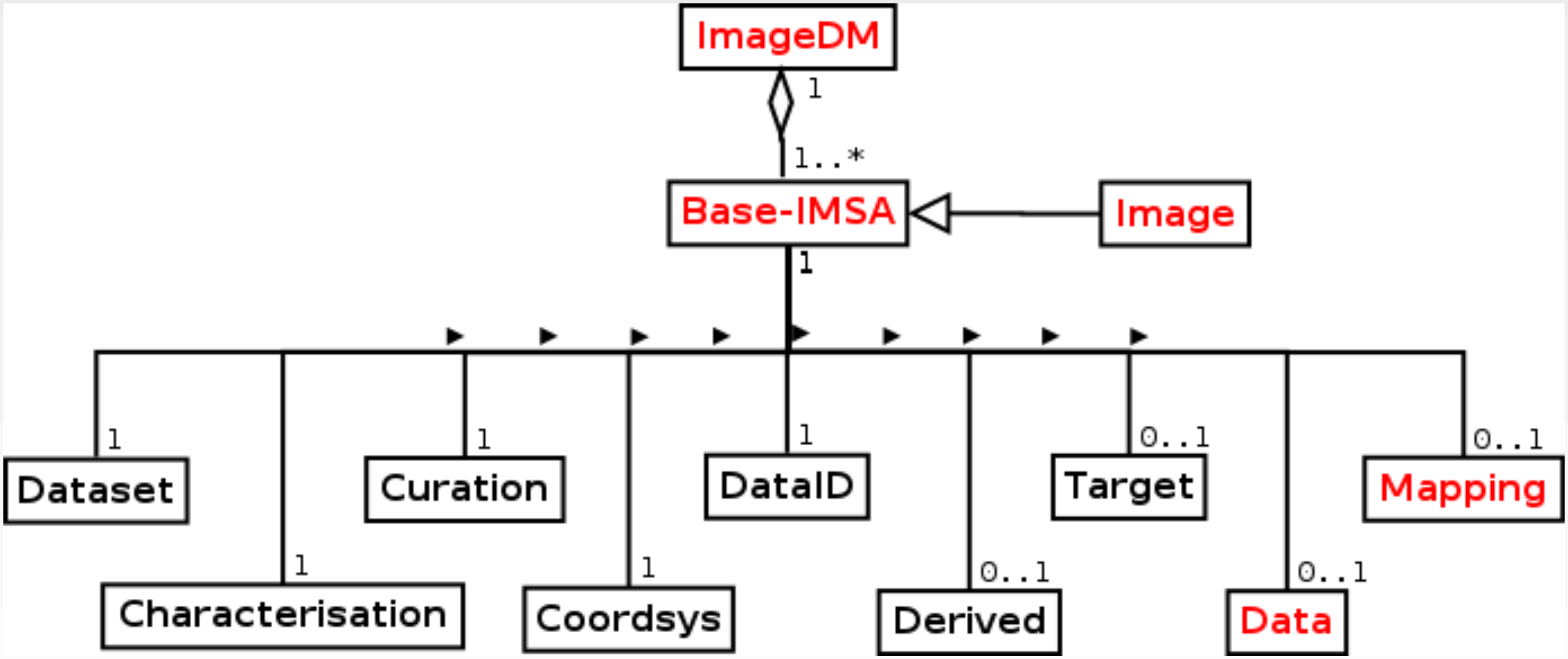


Image Data Model

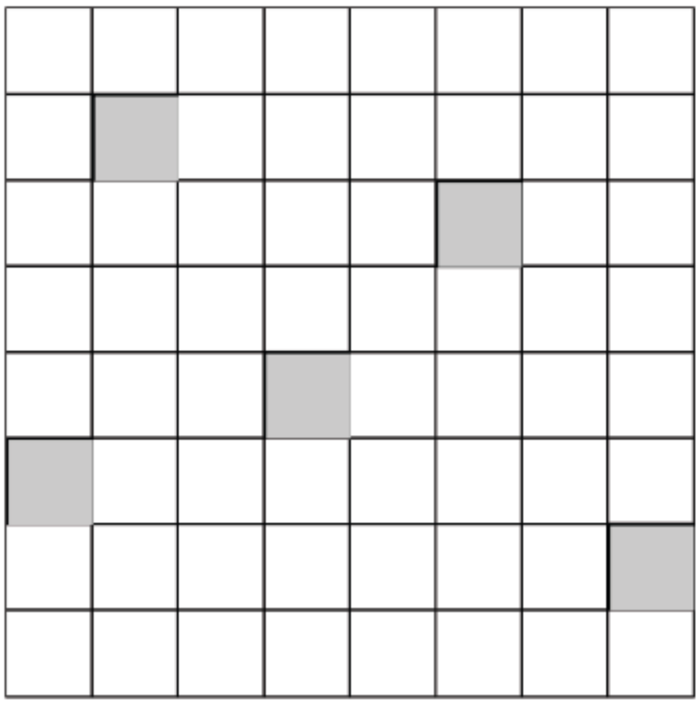
- Large Cubes
  - ALMA, JVLA, LOFAR, SKA pathfinders, etc.
  - Single cube can be Gigabytes to Terabytes
- Issues
  - Requires a more complex storage model
    - Simple FITS cube no longer adequate
      - file too large, access inefficient for non-spatial axes
    - Cube stored as multiple files, as a database,
      - possibly encoded (e.g. wavelet transform, JPEG2000)
    - Parallel storage/computation required for access
- Implication for ImageDM
  - Data model abstraction must hide how data is stored/represented
  - Can still view as a single large cube

Image Data Model

- Sparse Data
  - 2-D images may be sparse
    - We have generally lived with this as inefficiency was tolerable
  - Large cubes may be sparse
    - An axis has gaps where there is no data
      - Spatial plane not fully sampled, widely space spectral regions
    - No longer tolerable as cubes can be very large

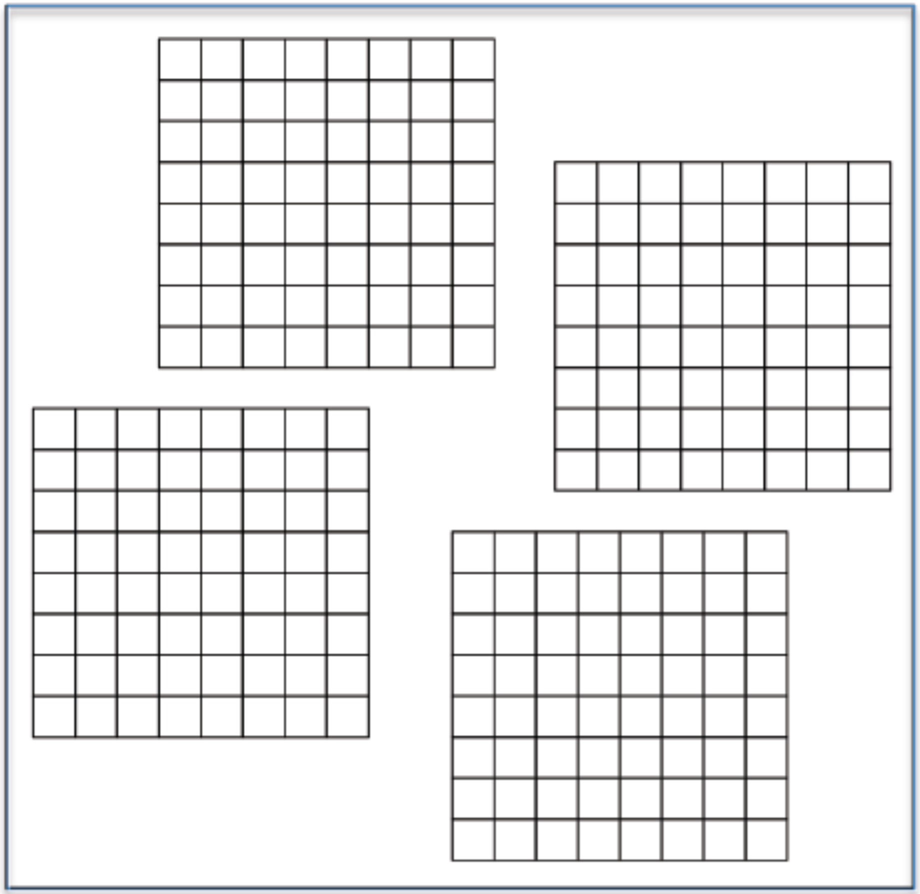
- Implication for ImageDM
  - Data model must support this directly
  - Techniques: multiple cubes, segmented cube, WCS-based index

Sparse Image – Selected Pixels/Voxels



Example of a **sparse image** (image or image cube which is sparse on the two coupled spatial axes). Data was obtained only for the points shown as gray in the figure. Rather than store the entire array, only data for the five sampled regions is stored. The coordinates of each sampled region are stored in a table included in the WCS for the image/ cube. In this example the sparse cube would be represented in 5/64 of the space that would be required to store the fully sampled cube.

Sparse Image - Segmented



Example of a **sparse image** (image or image cube which is sparse on the two coupled spatial axes), that is composed of several sub-arrays. The outer box defines the area of the super-array, or overall Image dataset. The four sub-arrays are individual

smaller images for which data was obtained. This example illustrates the use of multiple sub-arrays to cover a larger spatial region, however the same technique may be used for other axes such as the spectral, time, and polarization axes of a general cube.

Image Data Model

- Hypercube Data
  - N-d image is basis for ImageDM
  - General hypercube data is cube data, but not array-valued
    - e.g., event and visibility data
  - Approach
    - View as an N-d image by default
    - Reference to more fundamental data stored externally

- Event Data
  - Can be represented in ImageDM as sparse data, using WCS
  - WCS axes x, y, time, energy; observable optional (e.g. PHA)

Image Data Model

- AccessData Model
  - Closely tied to Image data model
  - Define logical model in ImageDM; usage in DAL protocol

- Logical Model
  - Filter -> WCS projection -> Pixel cutout -> Function
  - All steps are optional; entire image returned by default

Image Data Model

- Serializations
  - Data model described independent of serialization (Utypes)
  - Many serializations are possible
  - Each requires that a mapping be defined

- Some Important Formats
  - FITS
    - Main format for returning data, e.g. subsets, cutouts
  - VOTable
    - For VO discovery queries
  - HDF5 - container
  - CASA Image Table - blocked storage; CASA internal image format
  - JPEG2000 - wavelet encoding, multi-resolution, big data capability

Dataset	General dataset metadata
DataID	Dataset identification (creation)
Provenance	Instrumentalm or software Provenance
Curation	Publisher metadata
Target	Observed target, if any
CoordSys	Coordinate system frames
Char	Dataset characterization
Mapping	Dataset Axes Mapping or WCS
Characterization Metadata	
Char/FluxAxis	Observable, normally a flux measurement
Char/SpectralAxis	Spectral measurement axis, e.g., wavelength
Char/TimeAxis	Temporal measurement axis
Char/SpatialAxis	Spatial measurement axis
Char/Polarization	Polarization Axis
Char/*.Coverage	Coverage in any axis
Char/*.Resolution	Resolution on any axis
Char/*.SamplingPrecision	Sampling or Precision on any axis
Char/*.Accuracy	Accuracy and error in any axis
Mapping metadata	
Image matrix mapping	
WCS Mapping	

UTYPE	Description	Req	Default
Image Matrix Transform			
Mapping.NAxes	Number of image axes		
Mapping.NAxis[]	Length of each axis in pixels		
Mapping.CoordRefPixel[]	Reference pixel		
Mapping.CoordRefValue[]	WCS value at reference pixel		
Mapping.CDMatrix[]	Coord definition matrix		
Mapping.PCMatrix[]	Coord definition matrix		
Mapping.CDelt[]	World coord delta per pixel		
Mapping.AxisMap[]	Image-to-WCS axis mapping		
Mapping.WCSAxes	Number of WCS axes		
World Coord Transform			
Mapping.SpatialAxis.CoordType	Coordinate type as in FITS		
Mapping.SpatialAxis.Projection	Celestial projection		
Mapping.SpatialAxis.CoordFrame	Spatial coordinate frame		
Mapping.SpatialAxis.CoordEquinox	Coordinate equinox (if used)		
Mapping.SpatialAxis.CoordUnit	Unit for coordinate value		
Mapping.SpatialAxis.CoordName	Axis name (optional)		
Mapping.SpectralAxis.CoordType	Coordinate type as in FITS		
Mapping.SpectralAxis.Algorithm	Algorithm type as in FITS		
Mapping.SpectralAxis.RestFreq	Rest frequency of spectral line		
Mapping.SpectralAxis.RestWave	Rest wavelength of spectral line		
Mapping.SpectralAxis.CoordUnit	Unit for spectral coordinate value		
Mapping.SpectralAxis.CoordName	Axis name (optional)		
Mapping.SpectralAxis.CoordValue[]	Spectral value/band at pixel index		
Mapping.TimeAxis.CoordType	Time scale (UTC, TT, TAI, ...)		
Mapping.TimeAxis.CoordUnit	Time unit		
Mapping.TimeAxis.CoordName	Time axis name (optional)		
Mapping.TimeAxis.CoordValue[]	Time value at pixel index		
Mapping.TimeAxis.RefPosition	TOPOCENT, BARYCENT, ...		
Mapping.PolAxis.CoordType	Polarization system (Stokes etc.)		
Mapping.PolAxis.CoordName	Polarization axis name (optional)		
Mapping.PolAxis.CoordValue[]	Polarization type at pixel index		

Visibility Data

- Characterization
  - Pointing, FOV
  - UV distance plot
    - min/max UV distances, number of antennas, duration of exposure
  - Dirty beam plot
    - FWHM axes, max sidelobe expressed as % of peak
  - Freq sub-bands observed
    - support for velocity units (convention, ref frame, rest freq)
  - Resolution
    - size of synthesized beam (major, minor axes and angle)
  - Flux density, Jy/beam
  - Sensitivity, rms noise
  - Properties of possible generated images/spectra as ranges