The National Virtual Observatory Book
ASP Conference Series, Vol. 382, © 2008
M. J. Graham, M. J. Fitzpatrick, and T. A. McGlynn, eds.

Chapter 38: Data Models

Jonathan C. McDowell

Introduction

Modern astronomical datasets are complex in nature, and go well beyond simple images. Both the observational (or simulated) data points and the metadata that describe them are diverse both in structure and in the way they are represented as data files in archives. In particular, astronomers who specialize in different wavebands have adopted different ways of thinking about what is essentially the same data; sometimes these differences are arbitrary and trivial, and sometimes they are driven by real differences in the physics of the object or the instrumentation used in the detection. This complexity and diversity is a barrier to Virtual Observatory interoperability. We would like to allow VO-aware software to process standard data and metadata from any branch of astronomy, and this requires us to define these standards and map them to the representations familiar to the different kinds of astronomer. This is accomplished by the data-modeling process.

The Virtual Observatory data models standardize the structure and information content of metadata (header information) so that the VO and its users can interpret a dataset and its context, no matter what waveband or instrument is being represented. In this chapter we will discuss some of the more general aspects of the VO data model, and examine in detail how these concepts have been utilized in the Spectrum Data Model.

1. Data Models and the VO

Data model in this context means a standardized description of the information content of a particular kind of data. If you are a data provider who wants to publish some data to the VO in a compliant manner, you will have to create a file describing that data, typically in XML. You may also need, in some cases, to reformat the data itself, whether into VOTable or in a specified FITS binary format. You will need to know the specifics of the format (or `serialization'), but before that you need to know what information, i.e. what pieces of data and metadata, are needed. The data model can be thought of as a checklist to see if you have all the information you need in your archive; this checklist is independent of the actual format. The data model also can be used in a comparative way; by comparing it with a local data model for archive or domain specific data, one can describe formally the differences between the local model and the standard.

One of the main data modeling efforts to date has been to create a standard representation of a 1-dimensional astronomical spectrum. At its core, a spectrum is an array of fluxes versus a spectral coordinate such as wavelength or frequency. But to

be useful, a spectrum also needs a lot of contextual information: When was the spectrum taken? Where was the telescope pointing? What kinds of errors are provided? What is the effective spectral resolution? The spectral data model tells you what extra information is required, and what is optional but still standardized. A serialization of the model tells you how to write this information to the file. The advantage of the data model is that it separates these two ideas, so that if you later adopt a different file format you don't have to change the underlying model.

The data model is more than just a list of items - it also imposes a structure. In that sense, you can think of it as defining the internal data structures or software classes that should be generated if you read the data into a program. In many cases the data model will be delivered with a subroutine library that implements it. However, we don't consider the software functions (methods) in the library to be part of the definition of the data model: we concentrate on the structures only.

1.1 Fields and utypes

Each data model is defined by an IVOA document; each piece of information required by the model is associated with a 'field'. The names of the fields are called 'utypes' and are hierarchical in nature, with subfields separated by dots. Utypes are thus somewhat like UCDs, but have a different purpose: they specify a quantity's role within a given data model. Thus in a spectrum, the concept 'wavelength', which always has the UCD 'em.wl', may appear in several different roles, including 'Spectrum.Data.SpectralAxis.Value' (“I am the X axis of this spectrum object”). On the other hand, in a different spectrum instance the same 'Spectrum.Data.SpectralAxis.Value‘ utype may have a quite different UCD, for instance 'em.freq' (frequency). The field and utype definitions in an IVOA data model specify whether they are mandatory, recommended or optional. We try and keep the list of mandatory fields small, recognizing that archives often don't have a lot of metadata available to them. The recommended list corresponds to a higher level of compliance: data providers should try their best to supply the recommended metadata if at all possible, as users are likely to want this information. The optional list of fields provides a standard place for data providers to specify things if they happen to be available. Finally, the overall structure allows for extra archive-specific fields to be added for the use of specialized code, with the understanding that general VO software may skip them.

2. Characterization and Provenance

The VO Data Models make a distinction between the 'characterization' of the data and its 'provenance'. We assume that most VO datasets are processed, in the sense that they can be used for further analysis without special knowledge of the instrument or simulation code used to create the data. If you trust the archival data reduction of an image, then you don't need to know what flux conversion was applied to the pixel values or what the respective contributions to the spatial resolution are from the instrument and the seeing - but you do need to know the final flux units and the effective spatial resolution of the final data. The characterization gives you that kind of information, while the provenance gives you the information you would need if you don't trust the reduction and want to redo part of it. Provenance would include in-

strumental setup, observing conditions, and the software chain applied to the data; it answers the question 'Where did this data come from and what did you do to it?', while characterization answers the question 'What is this data now?'. We decided that standardizing characterization was more urgent; that model has now been completed and is going through the approval process, while work on a standard provenance model has been deferred.

2.1. Component Models: The structure of an observation

The Characterization model is a key component of the more general Observation metadata model that is still under development. The components of the Spectrum model will be reused for this general Observation model. We group the metadata in several key concepts: *CoordSys*, *DataID*, *Curation*, *Characterization* and *Target*.

The *Characterization* model is discussed separately below, and the *CoordSys* model will ultimately be covered by the Space-Time Coordinates paradigm, which has a chapter to itself (Chapter 37).

Two simple parts of the data provenance have been standardized, distinguished as the data identification metadata (*DataID*) and curation metadata (*Curation*). The former carries information specific to the original creation of the dataset, such as the instrument, filter, and sequence identifications assigned by the originating observatory. The latter describes this particular version of the dataset and the organizations or individuals responsible for it. It is intended to allow users to distinguish between different versions of a dataset held by different repositories, to let users find out who they can ask for more details about the data, and to tell them who they should give credit to if the data is used.

The *Target* metadata gives data providers a place to put contextual information about the field being observed. The target may be an astronomical object, or an empty field in the sky, or even a lab calibration source. A particular archive may find it convenient to label an observation of a star with a V magnitude or a cataloged redshift even if these values can't be derived from the data itself, and Target provides a structure to store such information.

2.2. The Characterization Data Model

The VO Characterization model describes the context and basic properties of a dataset, and is useful both for data discovery and further analysis. A Characterization consists of descriptions of axes (*Characterization Axes*), each with similar descriptions. The standard axes are the *Spatial* (usually 2-dimensional celestial), *Spectral* and *Time* axes, but others may be added as needed. For example, a derived dataset of spectral line properties might have 'ionization state' as an axis; you can have anything as an axis if there is a VO UCD that describes it. You can also add an axis for the dependent (measurement) variable, which describes the UCD, units, accuracy, etc. of the flux. A key idea is that the standard definition of a characterization description applies to any axis; it is the same for space, time, and spectral coordinates, and so it can be generalized to new axes easily. Note that we treat the spatial coordinate as a single `axis' even though it is two-dimensional; you cannot split up the RA and Dec

axes because the sky position is a single concept (and the appropriate 2D metric has off-diagonal elements)

Within the characterization axes, we describe *coverage*, *resolution* and *accuracy*. The 'frame' concept is shorthand for specifying exactly what the axis is and includes the UCD (defining what the axis represents physically), the units, and a reference to the coordinate system. It also specifies whether the axis is calibrated or not: for a science use case where the wavelengths of spectral lines are needed, it may be perfectly fine that the flux axis of a spectrum is uncalibrated if the wavelength axis is calibrated; conversely, measuring the flux of a bright known source in an image may not require the spatial axis to be absolutely calibrated. In the adopted IVOA model, the frame metadata are associated directly with the relevant characterization axis, with utypes like 'SpatialAxis.unit', while the coverage, resolution and accuracy metadata are separated out, with utypes like 'SpatialAxis.Resolution.Unit'.

The 'coverage' concept describes from where in the axis the data were taken. For example, the data in an R-band optical image are taken from a particular region of the spatial coordinates (the field of view) in a particular spectral range (the R band) and over a particular time interval (the exposure start time to stop time). Note that the number of characterization axes is always larger than the number of axes in the actual data.

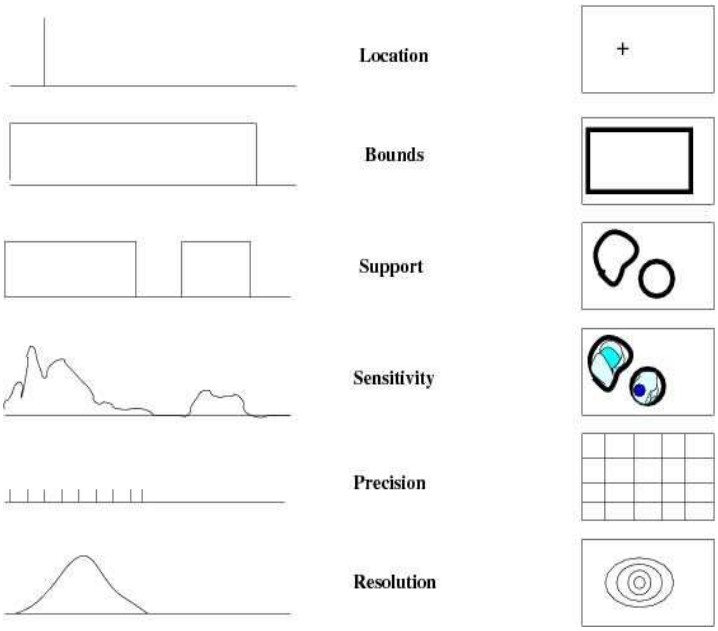

Figure 1. Levels of coverage. The location, bounds, support and sensitivity represent the region observed at different levels of fidelity. The sampling precision and the resolution give discrete and continuous limitations on the data sampling. The left hand and right hand columns illustrate the different cases for 1- and 2-dimensional coordinates respectively.

The 'resolution' concept is fairly straightforward and describes the effective resolution (spatial, temporal, spectral, etc.) of the data. A similar 'sampling' concept describes the discrete precision of the data axes.

The 'accuracy' concept contains all kinds of errors and quality flags. It includes provisions for describing two-sided statistical errors and systematic errors.

For each of coverage, resolution and accuracy, we allow different levels of fidelity (see Figure 1). The coarsest level, called *Location*, gives a single position on the axis as a representative indication of where the data is, e.g. the approximate pointing direction, the wavelength and the observation date. The next level, called *Bounds*, gives a range within which the data are expected to lay, i.e. the field of view, bandpass range and start/stop time. A finer level, *Support*, indicates where on the axis the sensitivity was non-zero. For the spatial axis, this is a polygon or other shape outlining the effective field of view, while for the time axis it may be an array of start and stop times.

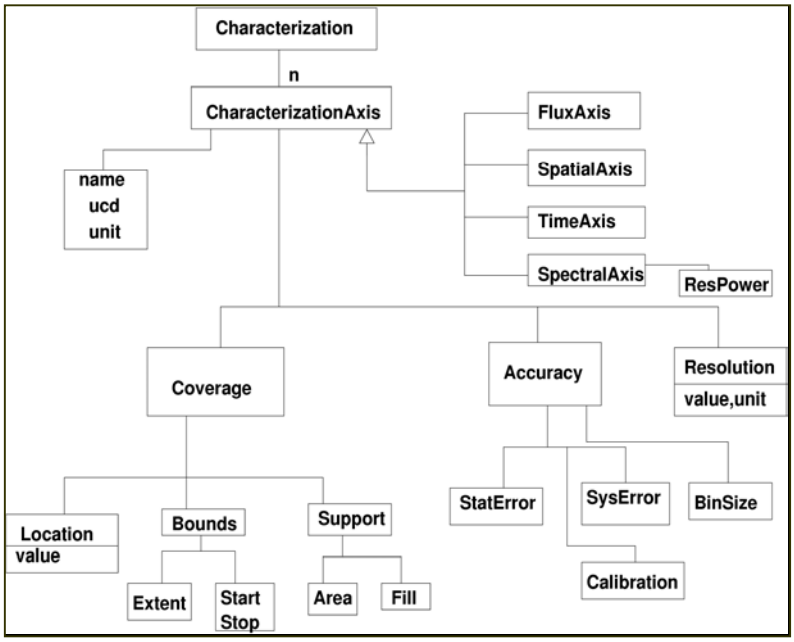

Figure 2. Data model for the simplified version of Characterization used in the Spectrum data model. Each box represents a data model field (utype name), and lines between the boxes represent substructure, with the arrow representing inheritance. Thus, the Area box for the SpatialAxis would then have the utype specified as 'Characterization.SpatialAxis.Coverage.Support.Area'.

In practical data analysis, we treat interruptions to the data in two different ways. Gross interruptions or gaps include the spatial gaps between chips, the spectral gaps between grating orders, or the temporal gaps between different orbits of a space-based observation where the Earth gets in the way. These are usually treated explic-

itly as separate observation segments. In contrast, small and often uncalibrated gaps, such as the gaps between individual pixels or the dead-time response that alters the effective exposure time of some detectors, are treated on a statistical basis using a 'filling factor'. The characterization model includes both explicit ranges for the Support and the ability to define such a filling factor.

The highest level of fidelity will be the *Sensitivity*, which gives the relative sensitivity as a function of pixel along the axis (we make the assumption that the sensitivities along each axis are independent). The idea here is that quantities such as the filter transmission curve and the exposure depth map are just more detailed versions of the bandpass limits and the field of view. The representation of the Sensitivity has not yet been standardized.

3. The Spectrum Data Model

The VO Spectrum data model is the one that is most completely developed so far. In addition to the data model itself, the standards document prescribes three possible serializations: one in FITS, one in VOTable and another in simple XML using a schema based directly on the model. For illustrative purposes I will use the VOTable example here and begin with a spectrum containing only the mandatory fields.

```
<VOTABLE version="1.1"
 xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
 xsi:noNamespaceSchemaLocation="http://www.ivoa.net/xml/VOTable/v1.1"
 xmlns:spec="http://www.ivoa.net/xml/SpectrumModel/v1.01"
 xmlns="http://www.ivoa.net/xml/VOTable/v1.1">
<RESOURCE utype="spec:Spectrum">
<TABLE utype="spec:Spectrum">
<GROUP utype="spec:Spectrum.Curation">
 <PARAM name="Publisher"
        utype="spec:Curation.Publisher"
        ucd="meta.organization;meta.curation"
        datatype="char" arraysize="*"
        value="SAO"/>
</GROUP>
<GROUP utype="spec:Spectrum.Target">
 <PARAM name="Target"
        utype="spec:Target.Name" datatype="char" arraysize="*"
        value="Arp 220"/>
</GROUP>
<GROUP utype="spec:Char">
 <GROUP utype="spec:Char.FluxAxis">
   <PARAM name="FluxAxisName" utype="Char.FluxAxis.name"
        ucd="phot.flux.density;em.wavelength"
        unit="erg cm**(-2) s**(-1) Angstrom**(-1)" value="Flux"/>
 </GROUP>
 <GROUP utype="spec:Char.SpectralAxis">
```

```
 <PARAM name="SpectralAxisName" utype="Char.SpectralAxis.name"
       ucd="em.wl" unit="Angstrom" value="Wavelength"/>
 <GROUP utype="Char.SpectralAxis.Coverage">
  <GROUP utype="Char.SpectralAxis.Coverage.Location">
   <PARAM name="NomLambda"
       utype="Char.SpectralAxis.Coverage.Location.Value" ucd="em.wl"
       value="4700.0"/>
  </GROUP>
  <GROUP utype="Char.SpectralAxis.Coverage.Bounds">
   <PARAM name="SpectralExtent"
       utype="Char.SpectralAxis.Coverage.Bounds.Extent"
       ucd="instr.bandwidth" unit="Angstrom" datatype="double"
       value="3000.0"/>
   <PARAM name="SpectralStart"
       utype="Char.SpectralAxis.Coverage.Bounds.Start"
       ucd="em.wl;stat.min" unit="Angstrom" datatype="double"
       value="3195.0"/>
   <PARAM name="SpectralStop"
       utype="Char.SpectralAxis.Coverage.Bounds.Stop"
       ucd="em.wl;stat.max" unit="Angstrom" datatype="double"
       value="6195.0"/>
  </GROUP>
 </GROUP>
</GROUP>
<GROUP utype="spec:Char.SpatialAxis">
  <GROUP utype="spec:Char.SpatialAxis.Coverage">
   <GROUP utype="Char.SpatialAxis.Coverage.Location">
    <PARAM name="SkyPos"
       utype="Char.SpatialAxis.Coverage.Location.Value"
       ucd="pos.eq" unit="deg" datatype="double" arraysize="2"
       value="132.4210 12.1232"/>
   </GROUP>
   <GROUP utype="Char.SpatialAxis.Coverage.Bounds">
    <PARAM name="SkyExtent"
        utype="Char.SpatialAxis.Coverage.Bounds.Extent"
        ucd="pos.region.diameter" datatype="double" unit="arcsec"
        value="20"/>
   </GROUP>
  </GROUP>
</GROUP>
<GROUP utype="spec:Char.TimeAxis">
 <GROUP utype="Char.TimeAxis.Coverage">
  <GROUP utype="Char.TimeAxis.Coverage.Location">
   <PARAM name="TimeObs"
       utype="Char.TimeAxis.Coverage.Location.Value" ucd="time.obs"
       datatype="double" value="52148.3252"/>
  </GROUP>
  <GROUP utype="Char.TimeAxis.Coverage.Bounds">
```

```
    <PARAM name="TimeExtent"
        utype="Char.TimeAxis.Coverage.Bounds.Extent"
        ucd="time.expo;phot.spectrum" unit="s" datatype="double"
        value="1500.0" />
   </GROUP>
  </GROUP>
 </GROUP>
</GROUP>
<GROUP utype="spec:Spectrum.Data">
  <GROUP utype="spec:Spectrum.Data.SpectralAxis">
    <FIELDref ref="Coord"/>
  </GROUP>

  <GROUP utype="spec:Spectrum.Data.FluxAxis">
    <FIELDref ref="Flux1"/>
  </GROUP>
</GROUP>
```

This VOTable header is easy to fill in: We provide a publisher name and a target name to provide human-useful identification. We then begin the Characterization section by defining the UCD and units for the flux (y) axis and the spectral coordinate (x) axis. The Coverage for the spectral axis has a nominal location of 4700A, an extent (width) of 3000A, and a start and stop wavelength of 3195-6195A. The Coverage for the spatial axis gives an RA and Dec position for the nominal location, an extent of 20 arc seconds corresponding to the effective aperture; the Coverage for the time axis gives an MJD value for the nominal location, and an extent of 1500s (the exposure time).

The next listing simply shows the remainder of the VOTable, which just specifies the FIELD column parameters and gives the data; no new metadata are involved:

```
<FIELD name="Coord" ID="Coord"
     utype="spec:Spectrum.Data.SpectralAxis.Value"
     ucd="em.wavelength" datatype="double" unit="Angstrom"/>
<FIELD name="Flux" ID="Flux1"
     utype="spec:Spectrum.Data.FluxAxis.value"
     ucd="phot.flux;em.wavelength" datatype="double"
     unit="erg cm**(-2) s**(-1) Angstrom**(-1)"/>
<DATA>
  <TABLEDATA>
    <TR><TD>3200.0</TD><TD>1.38E-12</TD></TR>
    <TR><TD>3210.5</TD><TD>1.12E-12</TD></TR>
    <TR><TD>3222.0</TD><TD>1.42E-12</TD></TR>
  </TABLEDATA>
</DATA>
</TABLE>
</RESOURCE>
</VOTABLE>
```

This minimally compliant spectrum shows that it's reasonably easy to generate a VOTable serialization of a Spectrum data model instance. But there are a lot of recommended parameters that are not included here even though they would be very useful for data consumers. The most obvious are the errors on the data; we strongly considered making these mandatory but relented because of the unfortunate prevalence of data without errors in old archives. In addition, the default values for recommended parameters may not be appropriate for your data. We strongly recommend including as many of the recommended parameters as practical.

Recommended parameters include:

- The spatial frame for the observation position (default ICRS; the default is that the observation times and spectral coordinates are uncorrected and still in the frame of the observer, and that times are given in TT, not UTC; any changes to this must be declared)
- Curation rights (default PUBLIC) and a URL or bibcode for documentation
- A target position on the sky (if this is meaningful)
- A polygon giving the extraction aperture on the sky
- The absolute times of exposure start and stop
- Typical statistical and systematic errors on the fluxes and spectral coordinates

With these metadata, both data discovery and basic data manipulation applications have the minimal information needed to do their job; data providers should consult the Spectrum DM document for details.

The DM is intended to represent an extracted one-dimensional spectrum, support for multi-segment data such as echelle spectra, or multidimensional data such as velocity cubes or unextracted long slit data, are deferred to a later standard. Within these limitations, the Spectrum DM can describe a wide variety of data, flux-calibrated or not, either as observed or with corrections to the wavelength frame. The use of UCDs allows a much more precise description of the y-axis of the data than simple units provide, and the spectrum can be represented as a function of wavelength, energy or frequency. We therefore hope that the new standard will allow a systematic approach to distributing spectral data and to developing new spectral analysis and display applications.

4. Summary

Even before the VO, the astronomical community had already to some extent standardized simple astronomical images and catalogs, so it was easy to exchange them. As the VO begins to provide more complicated kinds of data, we need a more sophisticated mechanism for data description. The VO data models will underlie second-generation publishing protocols and analysis interfaces, and provide a standard paradigm that data providers can use to organize the information they provide.

Three IVOA Data Models have now reached the Proposed Recommendation stage: the Space-Time Coordinates, Spectrum, and Characterization data models. The working group is beginning definition of a second generation of data models that will

standardize representation of spectral energy distributions and of general observation metadata.

Acknowledgements

The material presented in this chapter reflects the work of the IVOA Data Models Working Group, and in particular Mireille Louys, Francois Bonnarel, Arnold Rots, Alberto Micol, Doug Tody, Brian Thomas, Ray Plante, Anita Richards, Gerard Lemson, David Giaretta, Norman Gray, Tamas Budavari, Randy Thompson, Inga Kamp, Pedro Osuna, Jesus Salgado, Inaki Ortiz, Marie-Lise Dubernet, Markus Donlensky, and Kelly McCusker. This work has been supported by the NVO grant and by the Chandra X-ray Center, operated by Smithsonian Astrophysical Observatory for NASA under contract NAS8-03060.

Useful Links

The formal IVOA standards are at: http://www.ivoa.net/Documents/ [Accessed July 9, 2007]

IVOA Data Models Twiki at: http://www.ivoa.net/twiki/bin/view/IVOA/IvoaDataModel [Accessed July 9, 2007]