International

Virtual

Observatory

Alliance

IVOA Spectral Data Model

Version 0.96d

IVOA Working Draft
2006-05-12

This version:

http://www.ivoa.net/Documents/WD/SpectrumDM/SpectrumDM-2005XXXX.html

Latest version:

http://www.ivoa.net/Documents/latest/SpectrumDM.html

Previous versions:

http://hea-www.harvard.edu/~jcm/vo/docs/spec0.93.html

Note: ivoa.net links are not yet active; the above are placeholders.

Editors:

Jonathan McDowell, Doug Tody

Authors:

Jonathan McDowell, Doug Tody, Tamas Budavari, Markus Dolensky, Frank Valdes, Pavlos Protopapas, Arnold Rots, and the IVOA Data Access Layer Working Group.

Abstract

We present a data model describing the structure of spectrophotometric datasets with spectral and temporal coordinates and associated metadata. This data model may be used to represent spectra, time series data. and segements of SED (Spectral Energy Distributions).

Status of this document

This is a Working Draft, developed with the intention to support the Simple Spectral Access Protocol. This is the first global release of the document. The working group seeks confirmation that comments have been addressed to the satisfaction of the community.

Comments on this document are due 1 May 2006 for consideration in the next version of this document. They should be sent to the dal@ivoa.net mailing list.

This is an IVOA Working Draft for review by IVOA members and other interested parties. It is a draft document and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use IVOA Working Drafts as reference materials or to cite them as other than "work in progress." A list of current IVOA Recommendations and other technical documents can be found at http://www.ivoa.net/Documents.

This document has been developed with support from the National Science Foundation's http://www.nsf.gov/ Information Technology Research Program under Cooperative Agreement AST0122449 with The Johns Hopkins University, from the UK Particle Physics and Astronomy Research Council (PPARC) http://www.pparc.ac.uk, and from the Eurpean Commission's Sixth Framework Program http://fp6.cordis.lu/fp6/home.cfm via the Optical Infrared Coordination Network (OPTICON), http://www.astro-opticon.org.

The **Virtual Observatory (VO)** is general term for a collection of federated resources that can be used to conduct astronomical research, education, and outreach.

The **International Virtual Observatory Alliance (IVOA)** (http://www.ivoa.net) is a global collaboration of separately funded projects to develop standards and infrastructure that enable VO applications.

Contents

1 Introduction and Motivation

Spectra are stored in many different ways within the astronomical community. In this document we present a proposed abstraction for spectral data and serializations in VOTABLE, FITS, and XML, for use as a standard method of spectral data interchange.

We distinguish in several places between the implementation proposed in this document, referred to as Version 1, and capabilities proposed for possible later implementation.

```
Changes in V0.96d:
    * Corrected errors in schema.

Changes in V0.96:
    * Changed structure to emphasize single spectrophotometric segment.

Changes in V0.95:
    * Significant changes in  red italics.
    * Updated UCDs with July 2005 approved list.


Changes in V0.94:

    * Renamed old SED.CreatorID to SED.DocURL; added new
      DataID.CreatorID at Doug's suggestion
    * Updated UCDs like instr.beam and em.wavelength.
    * Proposed phot.flux.uncal

Changes in V0.93:

    * Revised defaults
    * Added version number in serializations.
    * Added FlatPoint with short attribute names as an option within the
      XML schema, to reduce XML verbosity.
    * Added Confidence to accuracy model
    * Added SEDQuantity to XML schema and derived Flux and Redshift from
      it, allowing Confidence as part of Accuracy.
    * Added per-segment Redshift element to DerivedData
    * Added Description to Target.
    * Rev (a): typo corrections
    * Rev (b): added segment size and FITS SIZE column

Changes in V0.92:

    * Attempted to include WCS Paper 3 keywords in FITS example
    * Various explanations improved (e.g. distinction between Target
      RA,Dec,z and per-segment values; role of BackgroundModel;
      consistency of units).
    * Coverage.Location fields changed to be of type SEDCoord, for
      consistency and to provide a way to store the spatial location error.
    * "Spatial" fields changed to "Sky"
    * Added SED.Type field
    * Added Curation.Reference field
```

2 Requirements

We need to represent a single 1-dimensional spectrum in sufficient detail to understand the differences between two spectra of the same object and between two spectra of different objects.

We need to represent time series photometry, with many photometry points of the same object at different times.

Finally, we need to represent spectral energy distributions (SED) which consist of multiple spectra and photometry points, usually for a single object.

3 Spectral Data Model summary

3.1 Model Components

Our model for a spectrum is a set of one or more data points (photometry) each of which share the same contextual metadata (aperture, position, etc.).

Specifically, a spectrum will have arrays of the following values:

- Spectral coordinate (e.g. wavelength), central and bin min and max
- Time coordinate, convertible to MJD UTC
- Flux value, with upper and lower statistical (uncorrelated) errors
- Quality mask
- Spectral resolution

and will have associated metadata including:

- Data collection and Dataset ID
- Instrument and filter ID
- Exposure time in seconds
- Position of aperture center, given as J2000 ICRS degrees.
- Aperture in degrees
- Systematic (correlated) error
- Bibcode

In section 4 we elaborate these concepts in detail, including some complications that we explicitly do not attempt to handle in this version. The data model fields and possible values are listed. We distinguish between optional and required fields in the text, as well as via a "Req" column in the tables which has values of R (Required) and O (Optional). Where appropriate we list those values of the physical units which interoperable implementations are required to recognize.

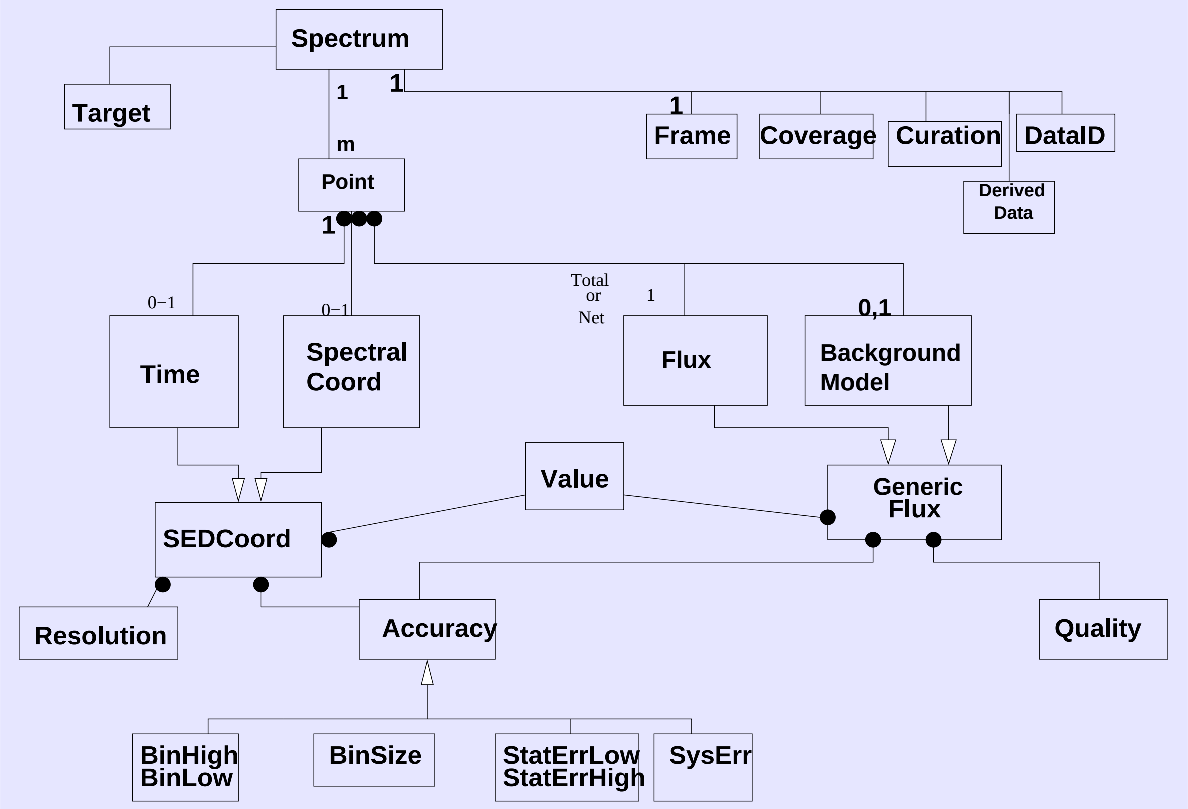

Figure 2: UML class diagram for the spectral data model.
The Frame, Coverage, Curation, DataID classes are shown in detail below in Section 4, both in diagram form and text descriptions. The text descriptions are definitive.
The minimal required content is:

- Target name
- Coverage.Location and Coverage.Extent descriptions of the location and extent of the data in the RA, Dec, time and spectral domains
- the Curation.Publisher field
- the descriptions of the spectral coordinate and flux fields including UCD and units
- For each point: the values of the spectral coordinate and flux.

3.2 Units

We adopt the WCS/OGIP convention for units: Document OGIP 93-001 (http://legacy.gsfc.nasa.gov/docs/heasarc/ofw

Briefly, units are given in the form

```
10**(-14) erg/cm**2/s/Hz, 10**3 Jy Hz
```

i.e. with exponents denoted by **, division by /, multiplication by a space.

This format is mostly consistent with the AAS standards for online tables in journals (http://grumpy.as.arizona.edu/∼gschwarz/unitstandards.html) except for the use of space rather than "." for multiplication and the fact that we do not require the use of SI units.

SI prefixes for units are to be recognized; for instance, the listing of "m" as a known unit for wavelength implies that "cm", "nm", and "um" (with "u" the OGIP convention for rendering "micro") are also acceptable.

Until IVOA generic unit conversion software is mature and widely deployed, it is helpful to interoperable applications to include a representation of the units in "base SI form", including only the base units kg, m, s (and possibly A, sr) with a numeric prefix. Pedro Osuna and Jesus Salgado have proposed a representation in the spirit of dimensional analysis, using the symbols M, L, T to signify kg, m, s respectively and omitting the ** for powers, so that

```
10**3 Jy Hz
```

which is equivalent to

```
10**-23 kg s**-2
```

is written compactly as

```
10-23MT-2
```

This alternate representation is supported for the main model fields (time, spectral coordinate and flux) only and is supported in the Frame field of the model.

Although the spectral model is flexible enough to permit different units for each field, as a matter of style we strongly recommend that whenever possible the same units should be used for compatible fields (e.g. flux and error on flux).

3.3 UCDs

UCDs or Uniform Content Descriptors are the IVOA's standardized vocabulary for astronomical concepts. In this document we use UCDs as field attributes (for example, element attributes in XML) to distinguish alternate physics within the same data model roles - for example, to distinguish frequency versus wavelength on the spectral coordinate 'X-axis'.

The current list of UCDs is http://cdsweb.u-strasbg.fr/UCD/ucd1p-words.txt with syntax defined in the UCD recommendation http://www.ivoa.net/Documents/latest/UCD.html.

UCDs should be case insenstive.

3.4 UTYPEs

UTYPE was a concept introduced in VOTABLE to label fields of a hierarchical data model. The word is now used generally to mean a standard identifier for a data mode field. They are also case-insensitive and are of the form a.b.c.d where the dots indicate a 'has-a' hierarchy with the leftmost element being the containing model and the rightmost element being the lowest level element referred to. This is quite close to a simple XPATH in an XML schema, but we chose not to use slash instead of dot to emphasize that we are only specifiying the element type, not the exact position in an instance (so no sophisticated query syntax).

4 Spectral data model Measurement objects

4.1 Spectral coordinate

Astronomers use a number of different spectral coordinates to label the electromagnetic spectrum. The cases enumerated by Greisen et al. (2003) are listed below with their UCDs.

REQUIRED: Exactly one SpectralCoord.Value field should be present, with units and one of the UCD values listed below. We distinguish between the VO data model field name (which might be used for VOTABLE UTYPE), the FITS WCS name (provided for comparison only), and the UCD1+ names.

Note 1: For this version, only the first four entries, Wavelength, Frequency, Energy, and spectral channel, should be used for interoperable transmission of data - implementations are not required to understand (convert) the other UCD values.

Note 2: For the velocity cases, the UCD uses a spect.dopplerVeloc tree rather than a *src.veloc* tree, because the velocity here is really a labelling of a spectral coordinate, and the link to the physical radial velocity of the different emission sources contributing to the spectrum is rather indirect.

Field	FITS WCS	UCD1+	Meaning	Units
PREFERRED CHOICES				
SpectralCoord.Value	**WAVE**	em.wl	Wavelength	Angstrom, m
SpectralCoord.Value	**FREQ**	em.freq	Frequency of photon	Hz
SpectralCoord.Value	**ENER**	em.energy	Photon energy	erg, eV, J
SpectralCoord.Value	-	instr.pixel;em.wl	Instrumental spectral bin	chan
ALTERNATIVE CHOICES				
SpectralCoord.Value	**WAVN**	em.wavenumber	Wavenumber	m**(-1)
SpectralCoord.Value	**AWAV**	em.wl;*obs.atmos*	Air wavelength	Angstrom, m
SpectralCoord.Value	**WAVE-LOG**	em.wl	Log wavelength	
SpectralCoord.Value	**FREQ-LOG**	em.freq	Log frequency of photon	
SpectralCoord.Value	**ENER-LOG**	em.energy	Log photon energy	
SpectralCoord.Value	**VELO**	*spect.dopplerVeloc*	Apparent radial velocity	m/s
SpectralCoord.Value	**VRAD**	*spect.dopplerVeloc.radio*	Radio velocity	m/s
SpectralCoord.Value	**VOPT**	*spect.dopplerVeloc.opt*	Optical velocity	m/s
SpectralCoord.Value	**BETA**	*spect.dopplerVeloc*	Velocity (c=1)	-

4.2 Flux (Spectral Intensity) Object

Two alternate representations of the flux are supported: Flux (the background-subtracted net flux) and TotalFlux (the source+background), possibly with the BackgroundModel (see below). Since there are no current UCDs to distinguish between these cases, we use data model field names.

REQUIRED: There should be exactly one Flux.Value or TotalFlux.Value field present. It is not currently permitted to have both Flux and TotalFlux in the same Spectrum. The field must include units and UCD.

For each of these cases, there are many slightly different physical quantities covered by the general concept of Flux; we distinguish them by their UCD.

Note in particular the distinction between the unit **count** (an instrumental value) and the unit **photon** (used in the photon number flux, i.e. the number of photons incident; photon number flux = energy flux divided by photon energy).

Note: The concept of the "nu L-nu" or "lambda L-lambda" flux, or equivalently the luminosity per logarithmic energy interval L(log nu), is a distinct concept in the world of spectral energy distributions - and it's a different concept from the bolometric luminosity, which has the same units. The UCD board has not yet approved a UCD expressing this concept; we have to use phys.luminosity and infer the concept from the units. My solution for brightness temperature is also rather questionable.

Flux Value choices

Field	UCD1+	Meaning	Unit
Flux.Value		(or...)	
TotalFlux.Value			
Flux.Value	*phot.fluDens*.uncal;em.wl	Flux density per unit wave, not absolutely calibrated.	
Flux.Value	*phot.fluDens*;em.wl	Flux density per unit wave.	erg cm$^{-2}s^{-1}\mathring{A}^{-1}$ Wm^{-2}m^{-1}, keVcm$^{-2}s^{-1}\mathring{A}^{-1}$
Flux.Value	*phot.fluDens*;em.freq	Flux density per unit freq.	erg cm$^{-2}s^{-1}$Hz^{-1} Jy, Wm^{-2}Hz^{-1}
Flux.Value	*phot.fluDens*;em.energy	Flux density per energy interval	keV cm$^{-2}s^{-1}$keV^{-1}
Flux.Value	*phot.fluDens*;em.energy; meta.number	Photons per unit area, time, energy	photon cm$^{-2}s^{-1}$keV^{-1}
Flux.Value	*phot.fluDens*,em.wl	Flux density per log wave interval (nu fnu)	Jy Hz
Flux.Value	*phot.fluDens*.sb;em.wl	Surface brightness per unit wavelength	erg cm$^{-2}s^{-1}\mathring{A}^{-1}$arcsec^{-2}
Flux.Value	*phot.fluDens*.sb;em.freq	Surface brightness per unit frequency	Jy sr^{-1}
Flux.Value	phot.count	Counts in spectral channel	count
Flux.Value	arith.rate;phot.count	Count rate in spectral channel	count/s
Flux.Value	arith.ratio;*phot.fluDens*	Flux ratio of two spectra	-
Flux.Value	phys.luminosity;em.wl	Luminosity per unit wave	erg$s^{-1}\mathring{A}^{-1}$, W/m
Flux.Value	phys.luminosity;em.freq	Luminosity per unit freq	erg$s^{-1}Hz^{-1}$, W/Hz
Flux.Value	phys.luminosity;em.energy	Luminosity per unit energy	ergs^{-1}keV^{-1}
Flux.Value	phys.*luminosity;em.energy*	Luminosity per log frequency	erg s^{-1}, W
Flux.Value	*phys.energDensity*	Radiation energy density per unit volume, per unit wave etc.	erg cm^{-3}, W m^{-3}
Flux.Value	phys.luminosity; phys.angArea;em.wl	Flux per unit solid angle (at source)	erg cm$^{-2}s^{-1}sr^{-1}\mathring{A}^{-1}$
Flux.Value	instr.antenna-temp	Antenna temperature	K
Flux.Value	*phot.fluDens;phys.temperature*	Brightness temperature	K
Flux.Value	phot.mag	Magnitude in defined band	mag
Flux.Value	phot.mag	AB (spectrophotometric) magnitude	mag
Flux.Value	*phot.fluDens*; instr.beam	Flux per resolution element (e.g. Jy/beam)	Jy/beam
Flux.Value	phot.mag.sb	Surface brightness in magnitudes	mag arcsec^{-2}

4.3 BackgroundModel Object

We optionally allow a BackgroundModel value for each Flux or TotalFlux value. Note that we define Flux = TotalFlux - BackgroundModel; thus, you don't need both Flux and TotalFlux. The name BackgroundModel, rather than Background, reminds us that it is an estimate: often, the BackgroundModel will be generated by taking a flux measurement at another location and rescaling it for any difference in exposure time or extraction aperture.

The BackgroundModel array is required to have the same UCD and units as the Flux array.

It represents a model for the expected flux values if the Target had zero flux.

OPTIONAL: There may be at most one BackgroundModel.Value field present. It must have the same UCD as the Flux or TotalFlux.

4.4 Time coordinate

The time coordinate is given by an elapsed time in some physical units (e.g. seconds or days) relative to a reference time. It refers to the midpoint of the sample interval. See the Space-Time Coordinates document for details of time coordinate complexity, and the Frame section of this document for discussion of the reference time Frame.Time.Zero.

The time unit is specified by a string, and the only valid values for this unit are 's' (seconds) and 'd' (days).

OPTIONAL: The Time.Value field is optional and defaults to 0.0 (relative to the reference time, which defaults to the Coverage.Location.Time as described in the Frame section).

Time Fields

Field	UCD1+	Meaning	Unit	Req
Time.Value	time	Photon arrival time since ref. time	s, d	O

4.5 Position coordinate

In general we may consider position coordinates as part of the measurement, but this is not included in the current document. The position of the spectrum is given in the Coverage.Location.Sky field. The Coverage.Location.Sky.Value field is an RA,Dec pair expressed in decimal degrees.

4.6 Accuracy Fields

We include accuracy models for both the coordinates (spectral, spatial and temporal) and the fluxes. (Note that the spatial coordinate appears only in the Coverage.Location). In the table below we have collapsed the model field names so that we have e.g. 'SpectralCoord.BinLow' rather than 'SpectralCoord.Accuracy.BinLow'.

We express the bandpass for each spectral bin as a low and high value for the spectral coordinate, or as a width. The same is done for photometry points, which amounts to approximating a filter by a rectangular bandpass. Time bins are also given as low and high values or as a width.

In addition to the binning, we allow the model to express uncertainties (which may be larger than the bin width), both statistical and systematic. We allow two-sided statistical errors but only one-sided systematic errors, and for position we have a single statistical error - eventually we may want a full error ellipse but this is too complicated for the present model.

We also use a very simple error model for the fluxes: we include plus and minus flux errors, and a quality flag. The errors are understood as 1 sigma gaussian errors which are uncorrelated for different points in the spectrum. If the data provider has only upper limit information, it should be represented by setting the flux value and the lower error value equal to the limit, and the upper error value equal to zero (e.g. 5 (+0,-5)). In general applications may choose to render measurements as upper limits if the flux value is less than some multiple (e.g. 3) of the lower error. We also allow a systematic error value, assumed constant across a given spectrum and fully correlated (so that, e.g. it does not enter into estimating spectral slopes). We also include a trivial resolution model: a single number nominally representing a FWHM spectral or time resolution expressed in the same units as the spectral or time coordinate. The spatial (sky) resolution may be useful to know if it exceeds the aperture size.

CLARIFICATION: the ErrorLow and ErrorHigh are the plus/minus ERRORS, not the (value+error, value-error). In other words, if Value = 10 and there is a symmetric uncertainty of 3, the ErrorLow and ErrorHigh are both +3.0, and NOT 7.0, 13.0. This is different from the sampling description BinLow and BinHigh, which give the VALUES at the low and high end of the bin. Thus if the central wavelength of the bin is 4200.0, and the bin size is 10, then the BinLow and BinHigh values are 4195.0, 4205.0 and NOT 10.0, 10.0. Note that because of this, 0.0 is NOT an acceptable default for BinLow and BinHigh, while it IS acceptable (albeit unlikely) for ErrorLow and ErrorHigh.

We add a further measure of accuracy, the Confidence, which expresses a probability between 0 and 1 that the quoted errors do apply. This measure is used in the Sloan spectral service to provide a way of describing the estimated probability that the redshift is completely in error because the lines have been misidentified - in general, such a Confidence could be useful for any measurement where the error probability distribution has multiple peaks in parameter space. Its default value is 1.0. This field is illustrated in the Target model (section 6.2).

The Quality model represents quality by an integer, with the following meanings: 0 is good data, 1 is data which is bad for an unspecified reason (e.g., no data in the sample interval), and other positive integers greater than 1 may be used to flag data which is bad or dubious for specific reasons. The data provider may define scalar string-valued metadata fields Quality.2, Quality.3... to define specific quality flags on a per-spectrum basis. Bitmasks, used in some archives such as SDSS, should be remapped to such independent Quality fields.

We also introduce a Calibration field which can have the values ABSOLUTE, RELATIVE or UNCALIBRATED. This is expected to be particularly useful to describe the flux. ABSOLUTE indicates that the values in the data are expected to be correct within the given uncertainty. RELATIVE indicates that although an unknown systematic error is present, the ratio of any two values will be correct. UNCALIBRATED indicates that although the values reflect a measurement of the given UCD, they are modified by an unspecified coordinate-dependent correction. Such values may be useful in the case of a spectrum with ABSOLUTE calibration on the wavelengths but UNCALIBRATED fluxes; the wavelengths of discontinuous features such as spectral lines can be measured on the assumption that the missing calibration function has no sharp discontinuities in the region of interest.

All the Accuracy fields are optional (both in the table fields and in the Coverage instances).

OPTIONAL: (SpectralCoord, Time) Only one of BinSize, or both BinLow and BinHigh, must be present (possibly as a header parameter implying a constant value for each flux point). If there is only one coordinate value (e.g. a time series with only one spectral coordinate value) the bin limits should be the entire range given in Coverage.Extent given in the Coverage model, else the bin limits are assumed to be halfway between the spectral coordinate values.

OPTIONAL: (SpectralCoord, Time) Resolution: - default is BinSize.

OPTIONAL: Sky.Resolution - default is the size of the aperture, given in Coverage.Extent.

OPTIONAL: StatErrLow, StatErrHigh, SysError fields for SpectralCoord, Time, Sky and Flux ; omitting these fields indicates that the errors are unknown. Data providers are STRONGLY encouraged to provide explicit error measures whenever possible.

Note that the errors on Sky are 1-dimensional; we might want to use error ellipses in a later release. StatErrorLow and StatErrorHigh, however, would be pairs of numbers for a 2-dimensional coordinate like Sky. We don't implement them in this version.

OPTIONAL: Flux.Quality field , default is 0 (good data).

Accuracy Fields

Field	UCD1+	Meaning	Req
SpectralCoord.BinLow	em.*;stat.min	Wavelength etc	O
SpectralCoord.BinHigh	em.*;stat.max	Wavelength etc	O
SpectralCoord.BinSize	em.*;?	Wavelength bin size	O
SpectralCoord.Resolution	instr.spectr.resolution	Spectral resolution FWHM	O
SpectralCoord.StatErrLow	em.*; stat.error	Spectral coord measurement error	O
SpectralCoord.StatErrHigh	em.*; stat.error	Spectral coord measurement error	O
SpectralCoord.SysError	em.*; stat.error	Spectral coord measurement error	O
SpectralCoord.Calibration		Type of coord calibration	O
Time.BinLow	em.*;stat.min	Time bin start	O
Time.BinHigh	em.*;stat.max	Time bin stop	O
Time.BinSize	em.*;?	Time bin size	O
Time.Resolution	time.resolution	Temporal resolution FWHM	O
Time.StatErrLow	time; stat.error	Time coord measurement statistical error	O
Time.StatErrHigh	time; stat.error	Time coord measurement statistical error	O
Time.SysError	time; stat.error.sys	Time coord measurement systematic error	O
Time.Calibration		Type of coord calibration	O
Sky.Resolution	instr.ang-res	Spatial resolution of data	O
Sky.StatError	pos.eq;stat.error	Astrometric statistical error	O
Sky.SysError	pos.eq;stat.error.sys	Systematic error	O
Sky.Calibration		Type of coord calibration	O
Flux.StatErrLow	phot.flux;em....;stat.error; *stat.min*	Negative error	O
Flux.StatErrHigh	phot.flux;em....;stat.error; *stat.max*	Positive error	O
Flux.SysError	phot.flux;stat.error.sys	Systematic error	O
Flux.Quality	meta.code.qual;phot.flux,em...	Quality mask	O
Flux.Quality.n		String value, for n = 0,1,2...; meaning of quality value	O
Flux.Calibration		Type of flux calibration	O

5 Associated Metadata Fields

5.1 Coverage Fields

The coverage fields will have a constant value for a given spectrum.

We give a value giving the effective exposure time (useful for selecting among multiple spectra from the same instrument). The aperture is important to determine what part of an extended object is contributing to the spectrum; we allow a simple aperture description consisting of a single number representing the aperture diameter in decimal degrees; it is anticipated that a full region description based on the CoordArea object will be supported in later versions.

The Coverage.Location.Sky element might have an HTMID (Hierarchical Triangular Mesh) integer value as well as an RA and Dec; a placeholder for this is in the XSD but we haven't really addressed it yet.

The units of Coverage.Extent.Spectral and Coverage.Region.Spectral should be the same as those of SpectralCoord.Value.

The Coverage.Region.Time is a pair of values giving the start and stop time; in principle it could be a whole array of start-stop pairs indicating data accumulated over a series of intervals, but we don't support that yet.

REQUIRED: Coverage.Location.Sky, Coverage.Location.Time, Coverage.Extent.Sky, Coverage.Extent.Time .

REQUIRED: Coverage.Location.Spectral; Coverage.Extent.Spectral . These are required to be present at the level of the abstract model. However, in the serializations of the model we break the symmetry between the spectral and other axes because in SED applications we'll often have a spectrum (SED segment) with just one spectral point, so it's easier to derive these values directly from the data.

OPTIONAL: Coverage.Region (defaults to full region of Coverage.Extent centered on Coverage.Location) **OPTIONAL: Coverage.Fill**

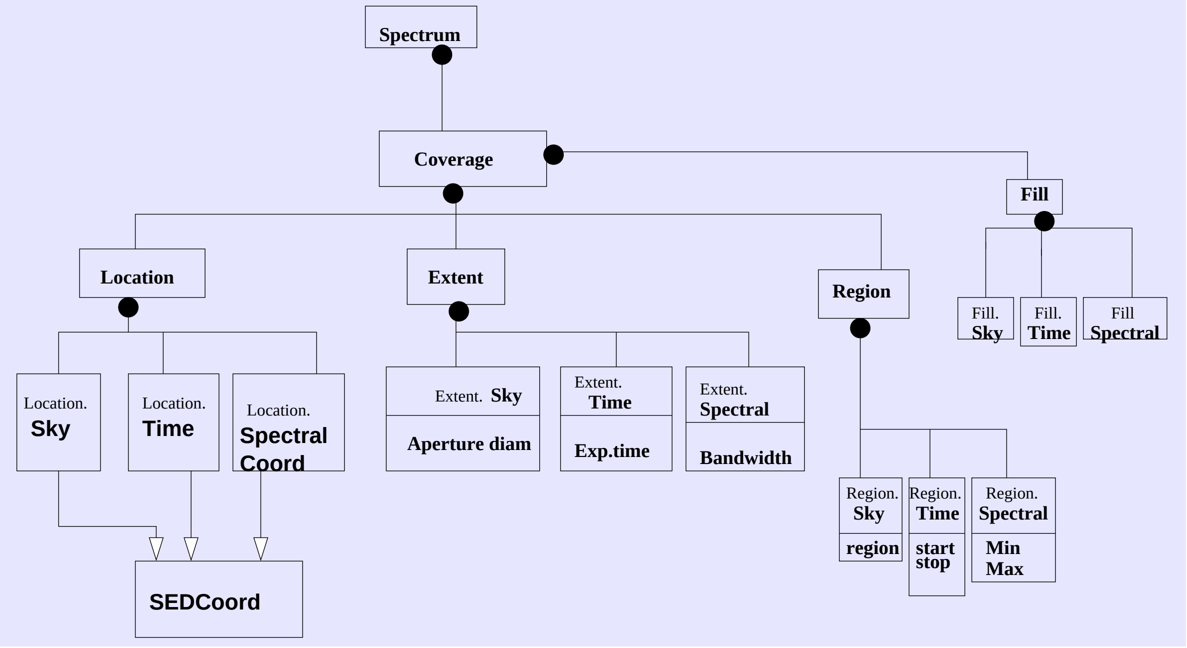

Figure 3: Diagram for Coverage object

Coverage Fields

Field	UCD1+	Meaning	Unit	Req
Coverage.Location.Sky.Value	pos.eq	RA and Dec	deg, (or sexagesimal string)	R
Coverage.Location.Time.Value	*time.epoch*	Midpoint of exposure, on MJD scale	d	R
Coverage.Location.SpectralCoord.Value	instr.bandpass	Band, consistent with RSM		R
Coverage.Extent.Sky	instr.fov	Aperture angular diameter	deg	R
Coverage.Extent.Time	time.expo	Exposure time	s	R
Coverage.Extent.Spectral	instr.bandwidth	Width of spectrum in A or other spec. coord.	(See text)	R
Coverage.Region.Sky		Aperture region	-	O
Coverage.Region.Time	time.expo.start, time.expo.end	Start and stop times	s	O
Coverage.Region.Time.Start	time.expo.start	Start time	s	O
Coverage.Region.Time.Stop	time.expo.end	Stop time	s	O
Coverage.Region.Spectral	em.*;stat.min,stat.max	Start/Stop in spectral coordinate	(see text)	O
Coverage.Fill.Sky	stat.*fill*;pos.eq	Sampling Filling factor	-	O
Coverage.Fill.Time	time;stat.*fill*;time	Sampling Filling factor	-	O
Coverage.Fill.Spectral	stat.*fill*;em.*	Sampling Filling factor	-	O

5.2 Frame Fields

The Frame object is a simplified instance of the STC CoordSystem object.

For all the numeric fields, units must be supplied. In addition, we add an SIDim field for each axis giving the SI units of the values in the Osuna-Salgado dimensional format.

For the spectral coordinate, position and time we need to add further metadata to define the frames used. Here we try and match existing FITS conventions (TIMESYS is a convention used in the X-ray community).

One standard reference time in astronomy is the origin of Julian Day Number on the TT (Terrestrial Time) timescale, BC 4713 Nov 24 at 11:59:27.81 (Gregorian). Using TT is preferable to UTC because it does not contain leap seconds, so the elapsed time in days is just equal to the difference in JD values.

The ISO-8601 calendar format standard does not support dates before AD 1, so cannot express this reference time. Therefore, it is not a suitable format for internal representations of such reference times. However, non-default choices of reference time may be specified in external serializations by a date in ISO-8601 format, e.g. "2004-11-30T11:59:00.01".

In this version of the model we require use of MJD as the time type for absolute times. (ISO dates and JD are other possibilities covered by the STC document).

(Note that in the FITS serialization, the MJDREF keyword allows definition of reference times in decimal days relative to MJD 0.0 = JD 2400000.5.)

Other frame information needed for velocity spectral coordinates include the observation-fixed spectral frame, the observatory location, the source redshift, and the velocity zero point (in Greisen et al, SSYSOBS, OBSGEO, VELOSYS, RESTFRQ/RESTWAV).

For Flux, the only entry is the SIDim field; we may later add a Type field to make finer distinctions in the type of flux scale than is possible using the Flux.Value's UCD.

Notes on compatibility with, and differences from, STC 1.0:

- We include the types (e.g. MJD, Cartesian) with the Frame (STC's CoordFrame), while STC puts them with the individual Coords instances. Implicitly, this requires that within a spectrum, all coordinate values for a particular coordinate be of the same type.
- Frame.Sky (STC SpaceFrame): we restrict the allowed expression of coordinates to a few simple cases.
  - the allowed values of Frame.Sky.Type (STC SpaceRefFrame) are ICRS or FK5. STC 1.0 allows additional cases.
  - For ICRS the equinox may be omitted; if present it should be J2000.0 (ICRS is not parameterized by equinox, but it does happen to coincide with the J2000 equator). For FK5 the equinox is required to be J2000.0.
  - the RefPos is not included and is assumed to be BARYCENTER (the solar system barycenter). This only matters for objects near enough to have measurable parallax; we do not support description of the position of solar system objects in this document, to make implementation simpler.
  - the Flavor is not included. It is implicitly required to be UNITSPHERE.
- TimeFrame: if you have a light curve whose times are corrected to the solar system barycenter, the results depend on the assumed source direction. RefPos = BARYCENTER and a RefDir with the assumed RA and Dec allow you to describe this. STC1.0 gives many possibilities for RefPos, but we support only TOPOCENTER, BARYCENTER, HELIOCENTER, GEOCENTER.

  The Frame.Time.Zero element defines the reference time (zero point) for all elapsed times in the spectrum, i.e. those in Coverage.Location.Time, Coverage.Extent.Time, Coverage.Region.Time, Point.Time.Value.
- SpectralFrame: the Frame.SpectralCoord.RefPos indicates the frame of the spectral coordinates, as in STC SpectralFrame ReferencePosition.
- RedshiftFrame: we don't include this - in this edition of the document, velocity axes (as opposed to spectral axes) are deprecated and their description is not supported.

OPTIONAL: All Frame values are optional , but data providers should take special care to check whether or not the defaults are appropriate for their data. The implications of the defaults are:

- Positions are given in ICRS RA,Dec in degrees and are heliocentric values (i.e. corrected for annual parallax and aberration, as normally found in source catalogs).
- Times are given in MJD days and represent times of photon arrival at the telescope.
- Spectral coordinates are as observed at the telescope, and not corrected for redshift, the motion of the Earth, etc.
- Note: we are considering making the Frame values required, particularly the SIDIM entries.

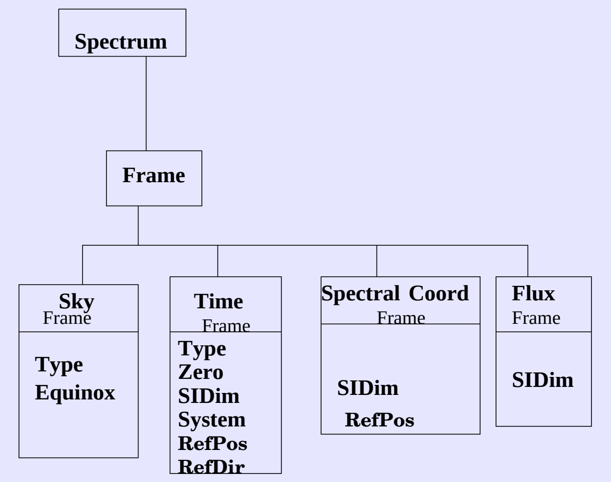

Figure 4: Diagram for Frame object

Frame Fields

Field	UCD1+	Meaning	Req
Frame.Sky.Type	?	ICRS (default) or FK5	O
Frame.Sky.Equinox	time.equinox;pos.frame	Default: 2000.0	O
Frame.Time.Type	time.scale	Time type, always equal to MJD	O
Frame.Time.Zero	time;arith.zp	Zero point of timescale in MJD, default 0	O
Frame.Time.SIDim	time;arith.zp	SI factor and dimensions	O
Frame.Time.System	time.scale	Timescale - default TT	O
Frame.Time.RefPos	time.scale	Times of photon arrival are at this location: default TOPOCENTRIC	O
Frame.Time.RefDir	time.scale	RA and Dec of assumed direction, omit if RefPos is TOPOCENTRIC	O
Frame.SpectralCoord.RefPos	?	default TOPOCENTRIC (use Greisen et al values)	O
Frame.SpectralCoord.SIDim	-	SI factor and dimensions	O
Frame.Flux.SIDim	-	SI factor and dimensions	O

5.3 Derived Data Fields

The Derived Data object has useful, and optional, summary information about the spectrum. For now, we include the option of adding signal-to-noise and variability indicators.

OPTIONAL: DerivedData (all fields)

Derived data Fields

Field	UCD1+	Meaning	Req
Derived.SNR	stat.snr	Signal-to-noise for spectrum	O
Derived.redshift		Measured redshift for spectrum	O
Derived.VarAmpl	src.var.amplitude;stat.ratio	Variability amplitude as fraction of mean (0 to 1)	O

5.4 Curation model

The Curation is an object consistent with the Curation information in the document "Resource Metadata for the Virtual Observatory Version 1.01", although we have added a Reference field for a bibliographic reference.

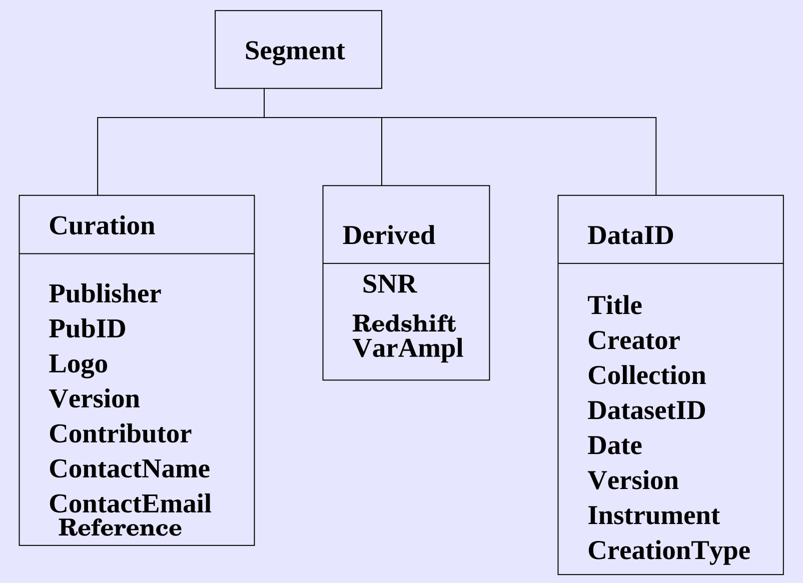

Figure 5: Diagram for Curation, DataID, Derived objects

REQUIRED: Curation.Publisher. As with the VO Resource Metadata document, this is the only required field in the Curation object. All other fields are optional.

Field	UCD1+	Meaning	
Curation.Publisher	*meta.curation*	Publisher	R
Curation.Reference	*meta.bib.bibcode*	URL or Bibcode for documentation	O
Curation.PubID	*meta.ref.url;meta.curation*	URI for VO Publisher	O
Curation.Logo	*meta.ref.url*	URL for creator logo	O
Curation.Version	*meta.version;meta.curation*	Version info	O
Curation.Contributor		Contributor (may be many)	O
Curation.ContactName	*meta.bib.author*;meta.curation	Contact name	O
Curation.ContactEmail	*meta.ref.url*;meta.*email*	Contact email	O

5.5 Data Identification model

The Data Identification model gives the dataset ID for a particular spectrum, and its membership of larger collections.

The DataID.DatasetID is the IVOA/ADEC/ADS dataset identifier. By agreement between the AAS journals, the ADS and the ADEC (NASA data centers), dataset identifiers, described in http://vo.ads.harvard.edu/dv/, will be used to link journal articles back to the archival datasets containing the relevant observational data.

OPTIONAL: All DataId fields are optional.

We introduce the concept of an dataset creation type, which can have one of the following three values:

- Archival, indicating that it is one of a collection of datasets (in this case spectra) generated in a systematic, homogeneous way and stored statically (or at least versioned). It will be possible to regenerate this dataset at a later date.
- Dynamic, indicating that the dataset was created 'on-the-fly', possibly by assembling the latest available data and applying a possibly changeable processing algorithm. A future attempt to generate this dataset in the same way may give different results.
- Custom, indicating that the dataset was created by manual analysis and processing, whether by data center staff or as part of a scientist's research project. Traceability of the processing may be incomplete.

DataID Fields

Field	UCD1+	Meaning	Req
DataID.Title	*meta.title;meta.dataset*	Dataset Title	O
DataID.Creator		VO Creator ID	O
DataID.Collection		Collection name(s)	O
DataID.DatasetID	meta.id;meta.dataset	IVOA Dataset ID	O
DataID.CreatorID	meta.id	Creator's ID for the dataset	O
DataID.Date	time;meta.dataset	Data processing/creation date	O
DataID.Version	*meta.version;meta.dataset*	Version of dataset	O
DataID.Instrument	meta.id;instr	Instrument ID	O
DataID.CreationType		dataset creation type	O

The dataset is associated with one or more Collections (instrument name, survey name. etc.) indicating some degree of compatibility with other datasets sharing the same Collection properties. Examples of possible Collection values are: "WFC", "Sloan", "BFS Spectrograph", "MSX Galactic Plane Survey".

5.6 Segment type and size

In SED applications, the Spectrum model is reused for both Spectrum and TimeSeries and is renamed Segment.

Each Segment may have a SegmentSize attribute giving the number of flux points in the segment (in some serializations this value is deduced from the size of the data arrays, while in others it is made explicit).

Each Segment may also have a SegmentType attribute indicating whether the data is intended as a TimeSeries (data are same spectral coord, varying times), Photometry (data are different spectral coords with irregular gaps), Spectrum (data are different spectral coords in contiguous bins), or Mixed (some mixture of the above).

This attribute is **OPTIONAL** and defaults to Spectrum.

5.7 Target model

In spectral data it is particularly important to be able to specify the target of the observation, which may be an astronomical source or some other target (calibration, diffuse background, etc.). By explicitly including a target model we can not only facilitate searches on particular types of target but also support archives of model spectra for which the Coverage fields may not be relevant.

The Target.pos field gives a nominal RA and Dec for the target, for example the catalog position of the source; the Coverage.Location fields in the spectrum indicate the actual telescope pointing position for that spectrum. (An SED might have a single Target object with a known position, but many Spectrum objects with slightly different telescope pointings). Similarly, the Target.redshift is the assumed actual cosmological redshift of the astronomical object, if applicable (again, usually from a catalog, NED, etc.), while the redshifts in the DerivedData objects in the spectrum (segment) indicates a redshift measured from that spectrum.

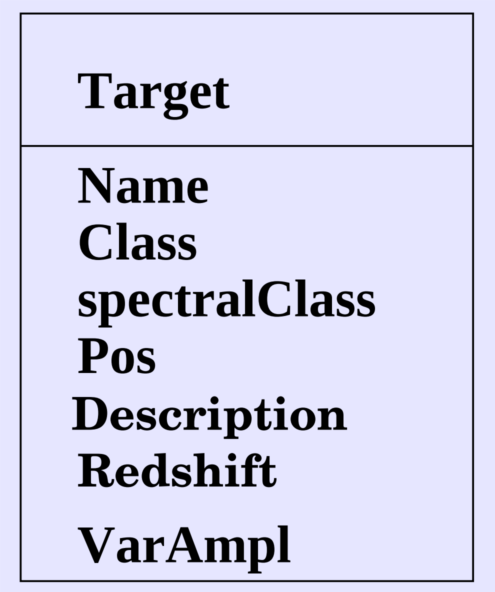

Figure 6: Diagram for Target object

REQUIRED: The Target.Name field is required.
OPTIONAL: All other Target fields are optional.

Target Fields

Field	UCD1+	Meaning	Req
Target.Name	meta.id;src	Target name	R
Target.Description		Target descriptive text	O
Target.Class	src.class	Target or object class	O
Target.spectralClass	src.spType	Object spectral class	O
Target.redshift.value	src.redshift	Target redshift	O
Target.redshift.statError	stat.error;src.redshift	Target redshift	O
Target.redshift.Confidence	stat.value;src.redshift	Target redshift	O
Target.pos	pos.eq;src	Target RA and Dec	O
Target.VarAmpl	src.var.amplitude	Target variability amplitude, typical	O

At the moment there is no international standard list of valid values for Target class and spectral class. Nevertheless an initial deployment of the VO would gain some benefit from using archive-specific classes, and provide a framework for converging on a standard list.

5.8 Packaging model

The simple Packaging model for SSA describes the format of the associated dataset. Allowed values for the format (detailed serialization for formats 4 to 7 to be specified in a separate document; only 1-3 are discussed here.)

These packaging values will be part of the SSA protocol response, and are implicit in the individual serializations.

- (1) FITS (standard BINTABLE for SED, to be defined)
- (2) VOTABLE
- (3) XML (native XML for web services and XML tools)
- (4) text (simple text table with columns of data and no markup)
- (5) text/html
- (6) graphics; a JPG, GIF etc. representation of the data
- (7) metadata; only the XML metadata.

6 Relationship to general VO data models

The Spectrum model involves objects addressed by the proposed VO Observation and Quantity data models. Although these models have not yet been fully worked out, we may note that a single Spectrum maps to the Observation model, which will include the Curation and Characterization objects. The Flux and the spectral coordinate entries together with their associated errors and quality will be special cases of the Quantity model, as will the simpler individual parameters. The field structure presented here is consistent with current drafts of the models.

7 Serializations

7.1 FITS serialization

We define a reference serialization of this data model as a FITS binary table. The table represents a spectrum or photometry point as a single row of a table.

This serialization is a special case of an SED serialization which uses one row per spectral segement; in that case, variable-length arrays may be used to contain the array quantities. In each case below where a ‘variable length array’ is specified, fixed length arrays are suitable for a single spectrum, but readers should be prepared to handle the variable length case.

Here we give the mapping of data model fields to FITS columns and keywords. For each column, the standard keywords TTYPEn, TUNITn, TFORMn should be provided. Order of keywords and columns is not significant, except that it is strongly recommended that RA and Dec be in adjacent columns or keywords.

The SPCO_UCD and FLUX_UCD keywords give the spectral coordinate and flux UCDs for the spectrum.

We add a new keyword VOCLASS to describe the VO object represented by the FITS table.

The spectral coordinate may also be identified by optional 1Sn_1 and 1CTYPn keywords as per WCS Paper 3. Table 9 of that paper implies that each data column which is a function of the spectral coord needs a pair of such keywords. Applications which implement the spectrum data model may ignore the WCS keys and interpret the file by recognizing ’by spec’ that SPCO is the spectral coordinate and that FLUX, etc. are functions of it, but the WCS keys give a general FITS application a chance at making sense of the file.

FITS keyword	Data model field	Value
VOCLASS	Spectrum	Spectrum V0.96
OBJECT	Target.Name	
SRCCLASS	Target.Class	
SPECTYPE	Target.spectralClass	
REDSHIFT	Target.redshift	
DS_IDENT	DataID.DatasetID	
TITLE	DataID.Title	
VERSION	DataID.Version	
VOPUB	Curation.Publisher	
VOPUBID	Curation.PubID	
VOREF	Curation.Reference	
VOLOGO	Curation.Logo	
CONTRIBn	Curation.Contributor	
CONTACT	Curation.ContactName	
EMAIL	Curation.ContactEmail	
RA_TARG	Target.pos	
DEC_TARG	Target.pos	

The following fields are keywords. *These keywords may instead appear as columns in an SED where their values may differ for different table rows.* The Position.Type field is not explicitly serialized: the names of the two columns used for Position are used to infer the type.

FITS keyword (or TTYPEn)	Data model field	
SPCO_UCD	SpectralCoord.ucd	
FLUX_UCD	Flux.ucd	
EQUINOX	Frame.Sky.Equinox	2000.0 (required)
RADECSYS	Frame.Sky.Type	either ICRS or FK5
TIMESYS	Frame.Time.Type	TT (required)
TIMEUNIT	(HEA convention)	
TIMECAL	Accuracy.Time.Calibration	
MJDREF	Frame.Time.Zero	(required)
TIMESDIM	Frame.Time.SIDim	(required)
SPECSYS	Frame.SpectralCoord.RefPos	(see Greisen et al)
SPECSDIM	Frame.SpectralCoord.SIDim	(required)
SPECCAL	Accuracy.SpectralCoord.Calibration	
FLUXSDIM	Frame.Flux.SIDim	(required)
FLUXCAL	Accuracy.Flux.Calibration	
INSTRUME	DataID.Instrument	
COLLECTn	DataID.Collection	
DATE-OBS	Coverage.Location.Time.Value	
RA_NOM	Coverage.Location.Sky.Value	
DEC_NOM	Coverage.Location.Sky.Value	
SPECBAND	Coverage.Location.SpectralCoord.Value	
SPECBWID	Coverage.Extent.SpectralCoord	
APERTURE	Coverage.Extent.Sky	
EXPOSURE	Coverage.Extent.Time	
SYS_ERR	Flux.SysError	
SKY_RES	Sky.Resolution	
SKY_SYE	Sky.SysError	
SKY_ERR	Sky.StatError	
SKY_CAL	Accuracy.Sky.Calibration	
TSTART	Coverage.Region.Time.Start	
TSTOP	Coverage.Region.Time.Stop	
SIZE	Number of points in spectrum (segment)	

The following fields are array columns. For a spectrum, all of these arrays have the same length - the number of points in the spectrum.

FITS TTYPEn	Data model field
SPCO	SpectralCoord
SPCO_LO	SpectralCoord.BinLow
SPCO_HI	SpectralCoord.BinHigh
TIME	Time
TIME_LO	Time.BinLow
TIME_HI	Time.BinHigh
FLUX	Flux
ERR_LO	Flux.StatErrLow
ERR_HI	Flux.StatErrHigh
QUALITY	Flux.Quality
TIME_RES	Time.Resolution
SPEC_RES	SpectralCoord.Resolution
SPEC_SYE	SpectralCoord.SysError
SPEC_SEL	SpectralCoord.StatErrLow
SPEC_SEH	SpectralCoord.StatErrHigh
TIME_SYE	Time.SysError
TIME_SEL	Time.StatErrLow
TIME_SEH	Time.StatErrHigh

Note: The ENERG_LO, ENERG_HI columns are already used in the X-ray community. We summarize this with a full sample FITS extension header.

```
XTENSION= 'BINTABLE'           / binary table extension
BITPIX  =                    8 / 8-bit bytes
NAXIS   =                    2 / 2-dimensional binary table
NAXIS1  =                57344 / width of table in bytes
NAXIS2  =                    1 / number of rows in table
PCOUNT  =                    0 / size of special data area
GCOUNT  =                    1 / one data group (required keyword)
TFIELDS =                    7 / number of fields in each row
EXTNAME = 'SPECTRUM '           / name of this binary table extension
VOCLASS = 'Spectrum V0.96'      / VO Data Model
VOCREATE= 'MMT Archive'       / VO Creator
VOCRID  = 'ivoa://cfa.harvard.edu' / VO Publisher ID URI
DATE    = '2004-08-30T14:18:17' / Date and time of file creation
DATE-OBS= '2004-06-03T21:18:17' / Date and time of observation
SPECBAND= ' '    / SED.Bandpass
RA_NOM       =     233.72789197     / [deg] Nominal RA
DEC_NOM      =      23.49792615     / [deg] Nominal Dec
OBJECT  = 'ARP 220 '           / Source name
SRCCLASS= 'Galaxy'             /
SPECTYPE= 'ULIRG'              /
RA_TARG      =     233.73791700     / Observer's specified target RA
DEC_TARG     =      23.50333300     / Observer's specified target Dec
REDSHIFT=              0.01812  / Emission redshift
DS_IDENT= 'cfa://whatever'   /
TITLE   = 'Observations of Merging Galaxies' /
VERSION =  2                     / Reprocessed 2004 Aug
VOPUB   = 'CfA Archive'        / VO Publisher authority
VOLOGO  = 'http://cfa.harvard.edu/vo/cfalogo.jpg' / VO Creator logo
SPCO_UCD= 'em.wl'                    /
FLUX_UCD= 'phot.fluDens;em.wl'     /
EQUINOX =  2.0000000000000E+03 / default
RADECSYS= 'ICRS    '           / default
TIMESYS = 'TT     '           / Time system
```

```
TIMEUNIT= 's'                / Time unit
SPECSYS = 'TOPOCENT'            / Wavelengths are as observed
TIMESDIM= 'T'                  / Time SIDim
SPECSDIM= '10-10 L'             / Spectral SIDim
FLUXSDIM= '10+7 ML-1T-3'       / Flux SDim
MJDREF  =  0.0                 / MJD zero point for times
SPEC_RES=              5.0  / [Angstrom] Spectral resolution
SKY_RES =              1.0  / [arcsec] Spatial.Resolution

TELESCOP= 'MMT '              / Telescope
INSTRUME= 'MMT/BCS '          / Instrument
FILTER  = 'G220    '          / Grating
RA      =     233.73791       / [deg] Pointing position
DEC     =      23.50333       / [deg] Pointing position
APERTURE=       2.0           / [arcsec]  Aperture diameter/Slit width
TIME     =  52984.301203      / [d] MJD
EXPOSURE =   1500.015         / [s] Effective exposure time
SYS_ERR  =      0.15          / Fractional systematic error
SIZE     =          2048      / Number of data points
COMMENT  ---------------------------
COMMENT  WCS Paper 3 Keywords
1S11_1   = 'SPCO'            / Column name with spectral coord
1CTYP11  = 'WAVE-TAB'        / Spectral coord is WAVE
1S12_1   = 'SPCO'            / Column name with spectral coord
1CTYP12  = 'WAVE-TAB'        / Spectral coord is WAVE
1S13_1   = 'SPCO'            / Column name with spectral coord
1CTYP13  = 'WAVE-TAB'        / Spectral coord is WAVE
1S14_1   = 'SPCO'            / Column name with spectral coord
1CTYP14  = 'WAVE-TAB'        / Spectral coord is WAVE
COMMENT  ---------------------------
TTYPE1 = 'SPCO'   / Wavelength
TFORM1 = '2048E'
TUNIT1 = 'Angstrom'
TTYPE2 = 'SPCO_LO' /
TFORM2 = '2048E'
TUNIT2 = 'Angstrom'
TTYPE3 = 'SPCO_HI' /
TFORM3 = '2048E'
TUNIT3 = 'Angstrom'
TTYPE4 = 'FLUX' /
TFORM4 = '2048E'
TUNIT4 = 'erg cm**(-2) s**(-1) Angstrom**(-1)'
TTYPE5 = 'ERR_LO' /
TFORM5 = '2048E'
TUNIT5 ='erg cm**(-2) s**(-1) Angstrom**(-1)'
TTYPE6 = 'ERR_HI' /
TFORM6 = '2048E'
TUNIT6 ='erg cm**(-2) s**(-1) Angstrom**(-1)'
TTYPE7 = 'QUALITY' /
TFORM7 = '2048I'
```

The data would look like

```
SPCO   SPCO_LO SPOCO_HI FLUX  ERR_LO  ERR_HI  QUALITY
3200.0 3195.0 3205.0 1.48E-12 2.0E-14 2.0E-14   0
3210.0 3205.0 3215.0 1.52E-12 3.0E-14 3.0E-14   0
3220.0 3215.0 3225.0 0.38E-12 0.38E-12 0.0      0
3230.0 3225.0 3235.0 1.62E-12 3.0E-14 3.0E-14   0
```

7.2 Alternate FITS serialization

Some implementors may find the one-row-per-spectrum approach inconvenient. An alternate 'relational normalization' serialization, considered but currently rejected by the VO group, is recorded here for information. It would consist of two binary or ASCII tables, each with only scalar columns. The Spectrum table would contain the same scalar columns as above, but the variable array columns would be moved to a second table, called SpecData, and used as scalar columns arranged vertically. A new integer column "OBS_ID" would be added to each table; the values in the Spectrum table would be unique and the corresponding values in the SpecData table would indicate which data points belonged to which rows in the Spectrum table.

Another approach would be to have one FITS HDU per spectrum or photometry point. However this was rejected as unworkable, as the overhead of 5760 bytes (2 FITS blocks) per photometry point would inflate the data for the photometry-only SED case by factors of around 50-100.

7.3 VOTable serialization

The VOTable version of Spectrum uses a single VOTable `<TABLE>`. (Note that this may appear as one of many tables within an SED VOTable). The data model fields described above as arrays map to VOTable FIELDs, while the remaining fields map to PARAM.

We use nested GROUP constructs to delimit data model objects within the main object, and PARAM and FIELD tags for attributes. Names of fields and parameters are left to the data provider. The utype and ucd attributes are used to denote data model and UCD tags. The schema and namespace for the utypes is the XML schema given in section 8.4. I have made up arbitrary NAME attributes for the PARAM and these are not to be considered standard; the name fields are free to be whatever the data provider wants, allowing compatibility with local archive nomenclature. The NAME attributes for the FIELD elements are also not standardized (of course they must be the same as in the matching FIELDrefs); it is the utype attribute which is standardized.

The one departure from the XML schema below is that the 'ArrayOfPoints' element and the individual 'Point' elements are implicitly represented by the table structure itself. Perhaps a UTYPE attribute to the TABLEDATA element could be used to make this explicit.

The example below describes a single SPECTRUM.

```
<?xml version="1.0" encoding="UTF-8"?>
<VOTABLE version="1.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://www.ivoa.net/xml/VOTable/v1.1"
  xmlns:sed="http://www.ivoa.net/xml/SpectrumModel/v0.96"
  xmlns="http://www.ivoa.net/xml/VOTable/v1.1">
<RESOURCE utype="sed:Spectrum">

<TABLE utype="sed:Spectrum">
<PARAM name="Date" utype="sed:Date" datatype="char" arraysize="*" value="2004-05-10"/>
<PARAM name="Nseg" utype="sed:NSegments" ucd="meta.number" datatype="int" value="1"/>
<PARAM name="Type" utype="sed:Type" datatype="char" arraysize="*" value="observed"/>
<GROUP utype="sed:Target">
 <PARAM name="Target" utype="sed:Target.Name" datatype="char" arraysize="*" value="Arp 220"/>
 <PARAM name="TargetPos" utype="sed:Target.pos" unit="deg" datatype="double"
                                  arraysize="2" value="233.737917 23.503330"/>
 <PARAM name="z" utype="sed:Target.redshift" datatype="float" value="0.0018"/>
</GROUP>

<!-- SegmentType can be Photometry, TimeSeries or Spectrum -->
<PARAM name="Segtype" utype="sed:Spectrum.SegmentType" datatype="char" arraysize="*"
                                                  value="Photometry" ucd="meta.code"/>
<GROUP name="Frame" utype="sed:Spectrum.Frame">
 <GROUP utype="sed:Spectrum.Frame.Sky">
   <PARAM name="System" utype="sed:Spectrum.Frame.Sky.Type" ucd="frame.pos.system"
                                         datatype="char" arraysize="*" value="ICRS"/>
   <PARAM name="Equinox" utype="sed:Spectrum.Frame.Sky.Equinox" ucd="time.equinox;pos.eq"
                                         datatype="float" value="2000.0" />
 </GROUP>
 <GROUP utype="sed:Spectrum.Frame.Time">
  <PARAM name="TimeFrame" utype="sed:Spectrum.Frame.Time.Type" ucd="frame.time.scale" datatype="char"
                                       arraysize="*" value="UTC"/>
  <PARAM name="TimeZero" utype="sed:Spectrum.Frame.Time.Zero" ucd="frame.time.zero" datatype="double" value="0.0"
 </GROUP>
 <GROUP utype="sed:Spectrum.Frame.SpectralCoord">
  <PARAM name="SpectralFrame" utype="sed:Spectrum.Frame.SpectralCoord.RefPos" ucd="frame.em.system"
                                      datatype="char" arraysize="*" value="Barycent"/>
 </GROUP>
</GROUP>
<GROUP utype="sed:Spectrum.Coverage">
 <GROUP utype="sed:Spectrum.Coverage.Location">
  <PARAM name="SkyPos" utype="sed:Spectrum.Coverage.Location.Sky.Value" ucd="pos.eq" unit="deg"
                                   datatype="double" arraysize="2" value="132.4210 12.1232"/>
  <PARAM name="TimeObs" utype="sed:Spectrum.Coverage.Location.Time.Value" ucd="time.obs"
                                   datatype="double" value="52148.3252"/>
 </GROUP>
 <GROUP utype="sed:Spectrum.Coverage.Extent">
  <PARAM name="SkyExtent" utype="sed:Spectrum.Coverage.Extent.Sky" ucd="pos.region.diameter"
                     datatype="double" unit="arcsec" value="20"/>
  <PARAM name="TimeExtent" utype="sed:Spectrum.Coverage.Extent.Time" ucd="time.expo;phot.spectrum"
                     unit="s" datatype="double" value="1500.0" />
  <PARAM name="SpectralExtent" utype="sed:Spectrum.Coverage.Extent.Spectral" ucd="instr.bandwidth"
                     unit="Angstrom" datatype="double" value="3000.0"/>
 </GROUP>
 <GROUP utype="sed:Spectrum.Coverage.Region">
  <GROUP utype="sed:Spectrum.Coverage.Region.Time">
   <PARAM name="TimeStart" utype="sed:Spectrum.Coverage.Region.Time.Start" ucd="time" unit="s"
                                     datatype="double" value="52100.000" />
   <PARAM name="TimeStop" utype="sed:Spectrum.Coverage.Region.Time.Stop" ucd="time" unit="s"
                                     datatype="double" value="52300.000" />
  </GROUP>
 </GROUP>


</GROUP>

<GROUP utype="sed:Spectrum.Curation">
 <PARAM name="Publisher" utype="sed:Spectrum.Curation.Publisher" ucd="meta.organization;meta.curation"
                       datatype="char" arraysize="*" value="SAO"/>
 <PARAM name="PubID" utype="sed:Spectrum.Curation.PubID" ucd="meta.curation.pubid" datatype="char"
                                       arraysize="*" value="ivoa://cfa.harvard.edu"/>
 <PARAM name="Logo" utype="sed:Spectrum.Curation.Logo" ucd="meta.curation.logo" datatype="char"
                                        arraysize="*" value="http://cfa-www.harvard.edu/nvo/cfalogo.jpg"/>
 <PARAM name="Contact" utype="sed:Spectrum.Curation.ContactName" ucd="meta.human;meta.curation"
                       datatype="char" arraysize="*" value="Jonathan McDowell"/>
 <PARAM name="email" utype="sed:Spectrum.Curation.ContactEmail" ucd="meta.email" datatype="char"
                                       arraysize="*" value="jcm@cfa.harvard.edu"/>
</GROUP>

<GROUP utype="sed:Spectrum.DataID">
 <PARAM name="Title" utype="sed:Spectrum.DataID.Title" datatype="char" arraysize="*" value="Arp 220 SED"/>
 <PARAM name="Creator" utype="sed:Segmenet.DataID.Creator" ucd="meta.curation.creator" datatype="char"
                                      arraysize="*" value="SAO/FLWO"/>
 <PARAM name="DateDate" utype="sed:Spectrum.DataID.Date" ucd="time;soft.dataset;meta.curation"
                      datatype="char" arraysize="*" value="2003-12-31T14:00:02Z"/>
 <PARAM name="Version" utype="sed:Spectrum.DataID.Version" ucd="soft.dataset.version;meta.curation"
                      datatype="char" arraysize="*" value="1"/>
 <PARAM name="Instrument" utype="sed:Spectrum.DataID.Instrument" ucd="inst.id" datatype="char"
                                    arraysize="*" value="BCS"/>
 <PARAM name="Filter" utype="sed:Spectrum.DataID.Collection" ucd="inst.filter.id" datatype="char"
                                   arraysize="*" value="G300"/>
 <PARAM name="CreationType" utype="sed:Spectrum.DataID.CreationType" datatype="char" arraysize="*" value="Archiva
</GROUP>

<GROUP utype="sed:Spectrum.Derived">
 <PARAM name="SNR" utype="sed:Spectrum.Derived.SNR" datatype="float" value="3.0"/>
</GROUP>

<GROUP utype="sed:Spectrum.Points">

<GROUP utype="sed:Spectrum.Points.SpectralCoord">
 <FIELDref ref="Coord"/>

 <GROUP utype="sed:Spectrum.Points.SpectralCoord.Accuracy">
  <FIELDref ref="BinLow"/>
  <FIELDref ref="BinHigh"/>
 </GROUP>
<!-- In this case Resolution is demoted from Field to Param since it is constant -->
 <PARAM name="Resolution" utype="sed:Spectrum.Points.SpectralCoord.Accuracy.Resolution"
                            unit="Angstrom" datatype="float" value="14.2"/>
</GROUP>
```

```
<GROUP utype="sed:Spectrum.Points.Flux">
 <FIELDref ref="Flux1"/>
 <GROUP utype="sed:Spectrum.Points.Flux.Accuracy">
  <FIELDref ref="ErrorLow"/>
  <FIELDref ref="ErrorHigh"/>
  <PARAM name="SysErr" utype="sed:Spectrum.Points.Flux.SysErr" unit="" datatype="float" value="0.05"/>
 </GROUP>
 <FIELDref ref="Quality"/>
</GROUP>
</GROUP>

<FIELD name="Coord" ID="Coord" utype="sed:Spectrum.Points.SpectralCoord.Value" ucd="em.wavelength"
              datatype="double" unit="Angstrom"/>
<FIELD name="BinLow" ID="BinLow" utype="sed:Spectrum.Points.SpectralCoord.BinLow" ucd="stat.min;em.wavelength"
              datatype="double" unit="Angstrom"/>
<FIELD name="BinHigh" ID="BinHigh" utype="sed:Spectrum.Points.SpectralCoord.BinHigh" ucd="stat.max;em.wavelength"
              datatype="double" unit="Angstrom"/>
<FIELD name="Flux" ID="Flux1" utype="sed:Spectrum.Points.Flux.Value" ucd="phot.flux;em.wavelength"
              datatype="double" unit="erg cm**(-2) s**(-1) Angstrom**(-1)"/>
<FIELD name="ErrorLow" ID="ErrorLow" utype="sed:Spectrum.Points.Flux.Accuracy.StatErrLow"
              datatype="double" unit="erg cm**(-2) s**(-1) Angstrom**(-1)"/>
<FIELD name="ErrorHigh" ID="ErrorHigh" utype="sed:Spectrum.Points.Flux.Accuracy.StatErrHigh"
              datatype="double" unit="erg cm**(-2) s**(-1) Angstrom**(-1)"/>
<FIELD name="Quality" ID="Quality" datatype="int" utype="sed:Spectrum.Points.Flux.Quality"/>
<DATA>
<TABLEDATA>
<!-- Note slightly nonlinear wavelength solution -->
<!-- Second row is upper limit -->
<!-- Third row has quality mask set -->
<TR><TD>3200.0</TD><TD>3195.0</TD><TD>3205.0</TD><TD>1.38E-12</TD><TD>5.2E-14</TD><TD>6.2E-14</TD><TD>0</TD></TR>
<TR><TD>3210.5</TD><TD>3205.0</TD><TD>3216.0</TD><TD>1.12E-12</TD><TD>1.12E-12</TD><TD>0</TD><TD>0</TD></TR>
<TR><TD>3222.0</TD><TD>3216.0</TD><TD>3228.0</TD><TD>1.42E-12</TD><TD>1.3E-14</TD><TD>0.2E-14</TD><TD>3</TD></TR>
</TABLEDATA>
</DATA>
</TABLE>


</RESOURCE>
</VOTABLE>
```

7.4 XML schema serialization

In the following XML schema, we implement the model fairly directly.

Within a spectrum the data points are kept together in objects called Point, with their general structure specified in a Fields object. Also, we have included a CustomParams element to allow site-specific metadata to be added.

The Coverage.Location fields have been collapsed to simple values rather than SEDCoord elements; this should perhaps be changed.

The Flux object is defined as an example of a more general SEDQuantity object, which is also used for the Sloan spectral service's redshift information.

The SED aggregation model is also included in the schema, as the top level element.

```xml
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:jxb="http://java.sun.com/xml/ns/jaxb" jxb:version="1.0">

<!-- The top level element: an SED with one target and many segments -->
<xs:element name="SED" nillable="true" type="SEDType" />
<xs:complexType name="SEDType">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="1" name="Date" type="TimeParam" />
<xs:element minOccurs="0" maxOccurs="1" name="Target" type="Target" />
<xs:element minOccurs="0" maxOccurs="1" name="CustomParams" type="ArrayOfParam" />
<xs:element minOccurs="0" maxOccurs="1" name="Type" type="TextParam" />
<xs:element minOccurs="0" maxOccurs="1" name="NSegments" type="IntParam" />
<xs:element minOccurs="0" maxOccurs="unbounded" ref="BaseSegment"/>
<xs:element minOccurs="0" maxOccurs="1" name="Creator" type="TextParam" />
<xs:element minOccurs="0" maxOccurs="1" name="CreatorID" type="TextParam" />
<xs:element minOccurs="0" maxOccurs="1" name="SpectralMinWavelength" type="DoubleParam" />
<xs:element minOccurs="0" maxOccurs="1" name="SpectralMaxWavelength" type="DoubleParam" />
</xs:sequence>
</xs:complexType>
<!-- A single segment corresponding to a spectrum or single point -->

<xs:element name="BaseSegment" type="SegmentType"/>
<xs:element name="Spectrum" type="SpectrumType" substitutionGroup="BaseSegment"/>
<xs:element name="Segment" type="SpectrumType" substitutionGroup="BaseSegment"/>
<xs:element name="TimeSeries" type="TimeSeriesType" substitutionGroup="BaseSegment"/>

<xs:complexType name="SpectrumType">
<xs:complexContent mixed="false">
<xs:extension base="SegmentType"/>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="TimeSeriesType">
<xs:complexContent mixed="false">
<xs:extension base="SegmentType"/>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="SegmentType">
<xs:complexContent mixed="false">
<xs:extension base="Group">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="1" name="SegmentType" type="TextParam" />
<xs:element minOccurs="0" maxOccurs="1" name="Frame" type="Frame" />
<xs:element minOccurs="0" maxOccurs="1" name="Coverage" type="Coverage" />
<xs:element minOccurs="0" maxOccurs="1" name="Curation" type="Curation" />
<xs:element minOccurs="0" maxOccurs="1" name="CustomParams" type="ArrayOfParam" />
<xs:element minOccurs="0" maxOccurs="1" name="DataID" type="DataID" />
<xs:element minOccurs="0" maxOccurs="1" name="Derived" type="DerivedData" />
<xs:element minOccurs="0" maxOccurs="1" name="Fields" type="Point" />
<xs:element minOccurs="0" maxOccurs="1" name="Target" type="Target" />
<xs:element minOccurs="0" maxOccurs="1" ref="Points"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
```

```
<!-- Define the UCDs etc for the SED coordinate and the flux coordinate,
and include a global to specify accuracy etc which happens to be
constant for the entire segment (note that in SEDCoord,
value has minOccurs=0 so it can be omitted) -->
<!-- A single SEDCoord (time or spectral coord) value. -->
<xs:complexType name="SEDCoord">
<xs:complexContent mixed="false">
<xs:extension base="Group">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="1" name="Value" type="DoubleParam" />
<xs:element minOccurs="0" maxOccurs="1" name="Accuracy" type="Accuracy" />
<xs:element minOccurs="0" maxOccurs="1" name="Resolution" type="DoubleParam" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="SEDQuantity">
<xs:complexContent mixed="false">
<xs:extension base="Group">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="1" name="Value" type="DoubleParam" />
<xs:element minOccurs="0" maxOccurs="1" name="Accuracy" type="Accuracy" />
<xs:element minOccurs="0" maxOccurs="1" name="Quality" type="IntParam" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!-- The error model. Bin entries will usually be omitted for the flux coordinate -->

<xs:complexType name="Accuracy">
<xs:complexContent mixed="false">
<xs:extension base="Group">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="1" name="BinLow" type="DoubleParam" />
<xs:element minOccurs="0" maxOccurs="1" name="BinHigh" type="DoubleParam" />
<xs:element minOccurs="0" maxOccurs="1" name="BinSize" type="DoubleParam" />
<xs:element minOccurs="0" maxOccurs="1" name="StatErrLow" type="DoubleParam" />
<xs:element minOccurs="0" maxOccurs="1" name="StatErrHigh" type="DoubleParam" />
<xs:element minOccurs="0" maxOccurs="1" name="SysErr" type="DoubleParam" />
<xs:element minOccurs="0" maxOccurs="1" name="Confidence" type="DoubleParam" />
<xs:element minOccurs="0" maxOccurs="1" name="Calibration" type="TextParam" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
```

```
<!-- A set of useful types to add UCDs and units to base types; like BasicQuantity -->
<xs:complexType name="Group" />

<xs:complexType name="TextParam">
<xs:simpleContent>
<xs:extension base="Param" />
</xs:simpleContent>
</xs:complexType>

<xs:complexType name="Param">
<xs:simpleContent>
<xs:extension base="xs:string">
<xs:attribute name="name" type="xs:string" />
<xs:attribute name="ucd" type="xs:string" />
</xs:extension>
</xs:simpleContent>
</xs:complexType>

<xs:complexType name="DateParam">
<xs:simpleContent>
<xs:extension base="Param" />
</xs:simpleContent>
</xs:complexType>

<xs:complexType name="PositionParam">
<xs:simpleContent>
<xs:extension base="Param">
<xs:attribute name="unit" type="xs:string" />
</xs:extension>
</xs:simpleContent>
</xs:complexType>

<xs:complexType name="DoubleParam">
<xs:simpleContent>
<xs:extension base="Param">
<xs:attribute name="unit" type="xs:string" />
</xs:extension>
</xs:simpleContent>
</xs:complexType>

<xs:complexType name="TimeParam">
<xs:simpleContent>
<xs:extension base="Param">
<xs:attribute name="unit" type="xs:string" />
</xs:extension>
</xs:simpleContent>
</xs:complexType>

<xs:complexType name="IntParam">
<xs:simpleContent>
<xs:extension base="Param">
<xs:attribute name="unit" type="xs:string" />
</xs:extension>
</xs:simpleContent>
</xs:complexType>
```

```
<!-- Define the space-time frame; eventually replace with STC -->
<xs:complexType name="Frame">
<xs:complexContent mixed="false">
<xs:extension base="Group">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="1" name="Sky" type="SkyFrame" />
<xs:element minOccurs="0" maxOccurs="1" name="Time" type="TimeFrame" />
<xs:element minOccurs="0" maxOccurs="1" name="Spectral" type="SpectralFrame" />
<xs:element minOccurs="0" maxOccurs="1" name="Flux" type="FluxFrame" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="SpectralFrame">
<xs:complexContent mixed="false">
<xs:extension base="Group">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="1" name="SpectralRefPos" type="TextParam" />
<xs:element minOccurs="0" maxOccurs="1" name="SIDim" type="TextParam" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="FluxFrame">
<xs:complexContent mixed="false">
<xs:extension base="Group">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="1" name="SIDim" type="TextParam" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="TimeFrame">
<xs:complexContent mixed="false">
<xs:extension base="Group">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="1" name="TimeType" type="TextParam" />
<xs:element minOccurs="0" maxOccurs="1" name="TimeZero" type="DoubleParam" />
<xs:element minOccurs="0" maxOccurs="1" name="TimeSystem" type="TextParam" />
<xs:element minOccurs="0" maxOccurs="1" name="TimeRefPos" type="TextParam" />
<xs:element minOccurs="0" maxOccurs="1" name="TimeRefDir" type="PositionParam" />
<xs:element minOccurs="0" maxOccurs="1" name="SIDim" type="TextParam" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="SkyFrame">
<xs:complexContent mixed="false">
<xs:extension base="Group">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="1" name="Type" type="TextParam" />
<xs:element minOccurs="0" maxOccurs="1" name="Equinox" type="DoubleParam" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
```

```
<!-- The Field type allows us to define what our axes are -->
<xs:complexType name="ArrayOfField">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="unbounded" name="Field" nillable="true" type="Field" />
</xs:sequence>
</xs:complexType>

<xs:complexType name="Field">
<xs:attribute name="Name" type="xs:string" />
<xs:attribute name="Unit" type="xs:string" />
<xs:attribute name="Ucd" type="xs:string" />
</xs:complexType>

<!-- The Point type groups a single set of time, spectral and flux values -->
<xs:element name="Points" type="ArrayOfGenPoint"/>

<xs:complexType name="ArrayOfGenPoint"/>

<xs:complexType name="ArrayOfPointType">
<xs:complexContent mixed="false">
<xs:extension base="ArrayOfGenPoint">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="unbounded" name="Point" nillable="true" type="Point" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:element name="ArrayOfPoint" type="ArrayOfPointType" substitutionGroup="Points"/>

<xs:complexType name="Point">
<xs:sequence>
<xs:element name="Time" minOccurs="0" maxOccurs="1" type="SEDCoord" />
<xs:element name="SpectralCoord" minOccurs="0" maxOccurs="1" type="SEDCoord" />
<xs:element name="Flux" minOccurs="0" maxOccurs="1" type="SEDQuantity" />
<xs:element name="BackgroundModel" minOccurs="0" maxOccurs="1" type="SEDQuantity" />
</xs:sequence>
</xs:complexType>

<xs:element name="ArrayOfFlatPoint" type="ArrayOfFlatPointType" substitutionGroup="Points"/>

<xs:complexType name="ArrayOfFlatPointType">
<xs:complexContent mixed="false">
<xs:extension base="ArrayOfGenPoint">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="unbounded" name="Point" nillable="true" type="FlatPoint" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
```

```
<xs:complexType name="FlatPoint">
<xs:attribute name="T" type="xs:double"/>
<xs:attribute name="T_BinL" type="xs:double" />
<xs:attribute name="T_BinH" type="xs:double" />
<xs:attribute name="T_Size" type="xs:double" />
<xs:attribute name="T_Res" type="xs:double" />
<xs:attribute name="SP" type="xs:double" />
<xs:attribute name="SP_BinL" type="xs:double" />
<xs:attribute name="SP_BinH" type="xs:double" />
<xs:attribute name="SP_Size" type="xs:double" />
<xs:attribute name="SP_Res" type="xs:double" />
<xs:attribute name="F" type="xs:double" />
<xs:attribute name="F_ErrL" type="xs:double" />
<xs:attribute name="F_ErrH" type="xs:double" />
<xs:attribute name="F_Sys" type="xs:double" />
<xs:attribute name="F_Qual" type="xs:int" />
<xs:attribute name="B" type="xs:double" />
<xs:attribute name="B_ErrL" type="xs:double" />
<xs:attribute name="B_ErrH" type="xs:double" />
<xs:attribute name="B_Sys" type="xs:double" />
<xs:attribute name="B_Qual" type="xs:int" />
</xs:complexType>


<!-- A special Position type including support for RA/Dec and HTM -->
<xs:complexType name="Position">
<xs:complexContent mixed="false">
<xs:extension base="Group">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="1" name="Value" type="PositionParam" />
<xs:element minOccurs="0" maxOccurs="1" name="HTMID" type="IntParam" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<!-- Now we define the higher level metadata -->
<xs:complexType name="Curation">
<xs:complexContent mixed="false">
<xs:extension base="Group">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="1" name="Publisher" type="TextParam" />
<xs:element minOccurs="0" maxOccurs="1" name="PubID" type="TextParam" />
<xs:element minOccurs="0" maxOccurs="1" name="Reference" type="TextParam" />
<xs:element minOccurs="0" maxOccurs="unbounded" name="Contributor" type="TextParam" />
<xs:element minOccurs="0" maxOccurs="1" name="Logo" type="TextParam" />
<xs:element minOccurs="0" maxOccurs="1" name="Version" type="TextParam" />
<xs:element minOccurs="0" maxOccurs="1" name="Contact" type="Contact" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
```

```
<xs:complexType name="Contact">
<xs:complexContent mixed="false">
<xs:extension base="Group">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="1" name="Name" type="TextParam" />
<xs:element minOccurs="0" maxOccurs="1" name="Email" type="TextParam" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="Coverage">
<xs:complexContent mixed="false">
<xs:extension base="Group">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="1" name="Location" type="CoverageLocation" />
<xs:element minOccurs="0" maxOccurs="1" name="Extent" type="CoverageExtent" />
<xs:element minOccurs="0" maxOccurs="1" name="Region" type="CoverageRegion" />
<xs:element minOccurs="0" maxOccurs="1" name="Fill" type="CoverageFill" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>


<xs:complexType name="CoverageLocation">
<xs:complexContent mixed="false">
<xs:extension base="Group">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="1" name="Sky" type="Position" />
<xs:element minOccurs="0" maxOccurs="1" name="Time" type="TimeParam" />
<xs:element minOccurs="0" maxOccurs="1" name="Spectral" type="TextParam" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="CoverageExtent">
<xs:complexContent mixed="false">
<xs:extension base="Group">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="1" name="Sky" type="DoubleParam" />
<xs:element minOccurs="0" maxOccurs="1" name="Time" type="DoubleParam" />
<xs:element minOccurs="0" maxOccurs="1" name="Spectral" type="DoubleParam" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
```

```
<xs:complexType name="CoverageFill">
<xs:complexContent mixed="false">
<xs:extension base="Group">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="1" name="Sky" type="DoubleParam" />
<xs:element minOccurs="0" maxOccurs="1" name="Time" type="DoubleParam" />
<xs:element minOccurs="0" maxOccurs="1" name="Spectral" type="DoubleParam" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="CoverageRegion">
<xs:complexContent mixed="false">
<xs:extension base="Group">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="1" name="Sky" type="SkyRegion" />
<xs:element minOccurs="0" maxOccurs="1" name="Time" type="Interval" />
<xs:element minOccurs="0" maxOccurs="1" name="Spectral" type="Interval" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>


<xs:complexType name="Interval">
<xs:complexContent mixed="false">
<xs:extension base="Group">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="1" name="Min" type="DoubleParam" />
<xs:element minOccurs="0" maxOccurs="1" name="Max" type="DoubleParam" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="SkyRegion">
<xs:complexContent mixed="false">
<xs:extension base="TextParam"/>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="DerivedData">
<xs:complexContent mixed="false">
<xs:extension base="Group">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="1" name="SNR" type="DoubleParam" />
<xs:element minOccurs="0" maxOccurs="1" name="VarAmpl" type="DoubleParam" />
<xs:element minOccurs="0" maxOccurs="1" name="redshift" type="SEDQuantity" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
```

```
<xs:complexType name="DataID">
<xs:complexContent mixed="false">
<xs:extension base="Group">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="1" name="Title" type="TextParam" />
<xs:element minOccurs="0" maxOccurs="1" name="Creator" type="TextParam" />
<xs:element minOccurs="0" maxOccurs="unbounded" name="Collection" type="TextParam" />
<xs:element minOccurs="0" maxOccurs="1" name="DatasetID" type="TextParam" />
<xs:element minOccurs="0" maxOccurs="1" name="Date" type="DateParam" />
<xs:element minOccurs="0" maxOccurs="1" name="Version" type="TextParam" />
<xs:element minOccurs="0" maxOccurs="1" name="Instrument" type="TextParam" />
<xs:element minOccurs="0" maxOccurs="1" name="CreationType" type="TextParam" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>


<xs:complexType name="Target">
<xs:complexContent mixed="false">
<xs:extension base="Group">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="1" name="Name" type="TextParam" />
<xs:element minOccurs="0" maxOccurs="1" name="Description" type="TextParam" />
<xs:element minOccurs="0" maxOccurs="1" name="TargetClass" type="TextParam" />
<xs:element minOccurs="0" maxOccurs="1" name="spectralClass" type="TextParam" />
<xs:element minOccurs="0" maxOccurs="1" name="redshift" type="SEDQuantity" />
<xs:element minOccurs="0" maxOccurs="1" name="pos" type="PositionParam" />
<xs:element minOccurs="0" maxOccurs="1" name="VarAmpl" type="DoubleParam" />
<xs:element minOccurs="0" maxOccurs="1" name="CustomParams" type="ArrayOfParam" />
</xs:sequence>
</xs:extension>
</xs:complexContent>

</xs:complexType>
<xs:complexType name="ArrayOfParam">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="unbounded" name="Param" nillable="true" type="Param" />
</xs:sequence>
</xs:complexType>
</xs:schema>
```

We reproduce below the VOTABLE instance example as an instance directly mapped from the schema. To go from the XML instance to the VOTABLE instance, we:

- - map the top level element to a RESOURCE
- - map all elements with simple content to PARAM
- - map all elements with complex content to GROUP
- - map the element names (with appropriate path) to values of the utype attribute,
- - but, handle the FIELDS and POINTS elements in a special way. The FIELDS element is used to define the table fields and the POINTS element is used to define the table data.

- - but, also, all the second level elements below RESOURCE except SPECTRUM map to an initial TABLE, while we map SPECTRUM to a second TABLE.

- - most of the elements extend the Param element, to which I have added an optional name attribute that I have not used in the instance. If this attribute is used, it can hold the name attributes of the PARAM and FIELD; otherwise the relevant attributes could be filled with the same value as the utype (without namespace prefix).

How can this be generalized to mapping an arbitrary data model schema to VOTABLE? The only tricky parts are

- **Spotting where the tabledata parts are.** We could require any DM schema that maps to VOTABLE to include elements called FIELDS and POINTS (perhaps ROWS would be a better name), otherwise you would get a VOTABLE with no data section.

- **Spotting where to start the main TABLE (i.e. the fact that SPECTRUM is special).** We could change the schema to have an explicit attribute, annotation or other marker to tell us this.

These issues will require further discussion for future models.

```
<?xml version="1.0" encoding="UTF-8"?>
<Spectrum xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://ivoa.net/xml/SED/SED-0.96.xsd">
<!-- xml instance example -->
  <Target>
   <name>Arp 220 </name>
   <pos>233.737917 23.503330</pos>
   <redshift>0.0018</redshift>
  </Target>
  <SegmentType ucd="meta.code">Spectrum</SegmentType>
  <Frame>
   <Sky>
    <Equinox ucd="time.equinox;pos.eq">2000.0</Equinox>
    <System ucd="frame.pos.system">ICRS</System>
   </Sky>
   <Time>
     <TimeType ucd="frame.time.system?">MJD</TimeType>
     <TimeZero ucd="frame.time.zero">0.0 </TimeZero>
     <TimeSystem ucd="frame.time.scale">UTC</TimeFrame>
     <TimeRefPos ucd="pos;frame.time">TOPOCENTER</TimeZero>
   </Time>
   <SpectralCoord>
     <SpectralRefPos ucd="frame.em.system">Barycent</SpectralRefPos>
   </SpectralCoord>
  </Frame>
  <Coverage>
  <Location>
    <Sky ucd="pos.eq" unit="deg">132.4210 12.1232</Sky>
    <Time ucd="time.obs" unit="d">52148.3252</Time>
   </Location>
   <Extent>
     <Sky ucd="pos.region.diameter" unit="arcsec">20</Sky>
     <Time ucd="time.expo;phot.spectrum" unit="s">1500.0</Time>
     <Spectral ucd="instr.bandwidth" unit="Angstrom">3000.0</Spectral>
   </Extent>
  </Coverage>
  <Curation>
   <Publisher ucd="meta.organization;meta.curation">SAO</Publisher>
   <PubID ucd="meta.curation.pubid">ivoa://cfa.harvard.edu</PubID>
   <Logo ucd=meta.curation.logo">http://cfa-www.harvard.edu/nvo/cfalogo.jpg</Logo>
   <Contact>
     <Name ucd="meta.human;meta.curation">Jonathan McDowell</Name>
     <Email ucd="meta.email">jcm@cfa.harvard.edu</Email>
   </Contact>
  </Curation>
  <DataID>
    <Title>Arp 220 SED</Title>
    <Creator ucd="meta.curation.creator">SAO/FLWO</Creator>
    <Date ucd="time;soft.dataset;meta.curation">2003-12-31T14:00:02Z</Date>
    <Version ucd="soft.dataset.version;meta.curation">1</Version>
    <Instrument ucd="inst.id">BCS</Instrument>
    <Collection>Archival</Collection>
  </DataID>
  <Derived>
    <SNR>3.0</SNR>
  </Derived>
  <!-- Define table structure -->
  <Fields>
    <SpectralCoord>
     <Value ucd="em.wavelength" unit="Angstrom"/>
     <Accuracy>
      <BinLow ucd="stat.min;em.wavelength" unit="Angstrom"/>
      <BinHigh ucd="stat.max;em.wavelength" unit="Angstrom"/>
     </Accuracy>
   </SpectralCoord>
   <Flux>
     <Value ucd="phot.flux;em.wavelength" unit="erg cm**(-2) s**(-1) Angstrom**(-1)"/>
     <Accuracy>
       <StatErrLow  unit="erg cm**(-2) s**(-1) Angstrom**(-1)"/>
       <StatErrHigh unit="erg cm**(-2) s**(-1) Angstrom**(-1)"/>
       <SysErr>0.05</SysErr>
     </Accuracy>
     </Quality>
   </Flux>
  </Fields>
  <!-- Use table structure -->
  <Points>
   <Point>
    <SpectralCoord>
     <Value>3200.0</Value>
     <Accuracy><BinLow>3195.0</BinLow><BinHigh>3205.0</BinHigh></Accuracy>
    </SpectralCoord>
    <Flux>
      <Value>1.38E-12</Value>
      <Accuracy><StatErrLow>5.2E-14</StatErrLow><StatErrHigh>6.2E-14</StatErrHigh></Accuracy>
      <Quality>0</Quality>
    </Flux>
   </Point>

   <Point>
    <SpectralCoord>
     <Value>3210.5</Value>
     <Accuracy><BinLow>3205.0</BinLow><BinHigh>3216.0</BinHigh></Accuracy>
    </SpectralCoord>
    <Flux>
      <Value>1.12E-12</Value>
      <Accuracy><StatErrLow>1.12E-12</StatErrLow><StatErrHigh>0</StatErrHigh></Accuracy>
      <Quality>0</Quality>
    </Flux>
   </Point>

   <Point>
    <SpectralCoord>
     <Value>3222.0</Value>
     <Accuracy><BinLow>3216.0</BinLow><BinHigh>3228.0</BinHigh></Accuracy>
    </SpectralCoord>
    <Flux>
      <Value>1.42E-12</Value>
      <Accuracy><StatErrLow>1.3E-14</StatErrLow><StatErrHigh>0.2E-14</StatErrHigh></Accuracy>
      <Quality>3</Quality>
    </Flux>
   </Point>
...
  </Points>
```

```
OR

  <Points>
  <FlatPoint SP=3200.0 SP_BinL=3195.0 SP_BinH=3205.0
                    F=1.48E-12 F_ErrL=2.0E-14  F_ErrH=2.0E-14/>
  <FlatPoint SP=3210.0 SP_BinL=3205.0 SP_BinH=3215.0
                    F=1.48E-12 F_ErrL=3.2E-14  F_ErrH=3.8E-14/>
  <FlatPoint SP=3220.0 SP_BinL=3215.0 SP_BinH=3225.0
                    F=1.48E-12 F_ErrL=1.48E-12  F_ErrH=0.0 />
...
 </Points>
</Spectrum>
```

7.5 Extensibility

The model and serializations defined in this document are extensible in the following sense:

- Future versions of the abstract (UML) model may add attributes or fields, and may deprecate the 'optional' property of existing fields.
- For the FITS serialization, implementors may add arbitrary additional keywords or table columns; readers must be able to handle files containing extra keywords and columns, and are encouraged to propagate such extra information when copying files. This permits local conventions to be layered on the basic definition.
- For the VOTABLE serialization, implementors may add arbitrary additional GROUP, PARAM or FIELDref/FIELD elements, with the restriction that the layering of existing elements should not be changed. (e.g. within the spectrum:DataID GROUP one may add a new GROUP containing newly defined PARAMs, but one may not move the existing Title PARAM inside the new group because that would change its indentation level). Readers must be able to handle files containing extra elements and are encouraged to propagate such extra information when copying files. This permits local conventions to be layered on the basic definition.
- For the XML object-based serialization, the CustomParams element at the top level of Spectrum is intended to allow extensibility and is equivalent to the ability to add a new GROUP at the top level. Future versions of the schema could use type extension to back-compatibly include the current schema as a special case, but apart from the CustomParams we have not currently provided for local extensibility within the current schema. (We could improve the schema by allowing the Group element to have arbitrary extra Param elements.)

References

Greisen, EW, Valdes F G, Calabretta M R and Allen S L 2003, in prep.,
http:www.aoc.nrao.edu/ egreisen/scs_112103.ps.

Hanisch, R., (ed)., Resource Metadata for the VO, Version 1.01, 2004 Apr 26.
http://www.ivoa.net/Documents/latest/RM.html

Derriere, S. et al (eds.), UCD, Moving to UCD 1+, 2004 Oct 26.
http://www.ivoa.net/Documents/latest/UCD.html