Observation Coverage and Space-Time Coordinates

IVOA DM WG Internal Note
2004-03-31

Working Group: http://www.ivoa.net/twiki/bin/view/IVOA/IvoaDataModel
Authors:
Jonathan McDowell

Abstract

The Space-Time Coordinates (STC) schema and its relationship to the Observation data model are described.

Status of this document

This is a Working Group Internal Note.

1 Introduction

In this document I summarize Arnold Rots' proposal on Space-Time Coordinates (STC) and discuss how it might be incorporated in the Observation data model.

1.1 The Observation Characterization and Coverage models

In a draft document on the IVOA DM WG site,
http://www.ivoa.net/twiki/bin/view/IVOA/IvoaDataModel
we have proposed the Observation data model. In this model we describe a Characterization model to describe the context of an observation. An observation is located within an m-dimensional parameter space; the observation itself may be an n-dimensional array, with n different from m, or some other kind of data such as a catalog or table. For example, a celestial image may

be a two-dimensional array where the dimensions are mappings from RA and Dec; but its characterization includes a Coverage object which says not only which part of the sky the image represents (the two dimensions of the image) but also which wavelengths the image covers, what time the image was taken, and so on. Furthermore, there may be coupling between these dimensions - the spatial coordinate system may be defined in terms of a time - the equinox of the equator - which is different from the time of the observation, and the flux scale may be corrected to a velocity rest frame different from the rest frame of the positional coordinate system. The STC concept attempts to cover all the nasty possibilities that can come up in the spatial, temporal, spectral and velocity axes.

1.2 Theory data and relocatability

There are other axes that may also need to be represented, and there are datasets - especially theoretical ones - which may not be representable by an absolute STC description. As an example of a theoretical image, consider an image whose axes are (1) magnitude of orbital velocity and (2) angle between velocity vector and the vector to the coordinate origin, with (3) pixel values equal to the resulting orbital eccentricity in the context of Keplerian orbits in a central field. Here we have quantities which are only ‘partially absolute’: the coordinate origin is not any particular point in the universe, the velocity has physical units and is defined to be in the inertial frame of the coordinate origin, but that inertial rest frame is again not located absolutely in space, and is entirely unanchored to any absolute time; since no photon flux is involved in the problem, the issue of a spectral frame does not even arise.

However, one could imagine taking this dataset and applying it to the ephemeris of a proposed astronomical satellite orbiting Mars. In that case the spatial coordinate frame becomes more fully defined, although there is still no particular absolute location implied (without arbitrarily choosing the remaining orbital elements). The temporal coordinates remain at best speculative; after funding and launch, the temporal coordinates become defined, but spectral coordinates remain forever irrelevant to this particular class of dataset.

These properties of relocatability, partial definition, and relevance of a limited and non-standard set of axes are common characteristics of theory data, although ancillary engineering data for observations (commonly used in space mission data analysis) also have these properties. In contrast, the bulk of near-term VO applications involve observations of photon fluxes which, in order to be fully specified, require the full STC specification of spatio-temporal, angular, spectral and Doppler frames, and so this is an important special case.

1.3 Existing STC documentation

Because some of our team have found the STC proposal complicated to follow, I try here to give my own, possibly over-simplistic, understanding of it. The most up to date official documentation is on Arnold's web site at

http://hea-www.harvard.edu/∼arots/nvometa

and consists of the XML schema files themselves:

- stc.xsd
- coords.xsd
- region.xsd,

  and the new document on a UML model for STC,

- STC_UML.pdf

In addition, there is:

- STCdoc.pdf (May 2003), a paper explaining the XML
- SpaceTime.html (Jul 2003), more text describing the XML
- STC.pdf (Aug 2003), a poster giving the XMLspy version of the model on a single sheet.
- STC.html, (May 2003), a huge auto-generated set of XMLspy documentation.

These latter documents are not quite in sync with the latest version, but the changes since last year, while important from the point of view of making the model more generalizable and coherent, are not huge.

1.4 UCDs

Any place where an STC element must have one of a predefined set of string values (e.g. the StdPole which can be one of EQUATORIAL, ECLIPTIC, etc.), we are defining a controlled vocabulary of astronomical concepts. I would argue that any such controlled vocabulary should be a subset of the UCDs, which from the official data dictionary for astronomical concepts in the VO. We should therefore review all these possible values and assign or create UCDs for them. I have marked these cases in the text as [UCD?].

2 The STC model

The STC model is intended as a description of the coverage of a dataset, survey or observation. It includes three main parts:

- The coordinate system, defining the parameter space of the coverage
- The coordinate area, defining the region within that space which contains valid data.

- The coordinate location, defining a reference point within that space that may be used to approximately characterize the data.

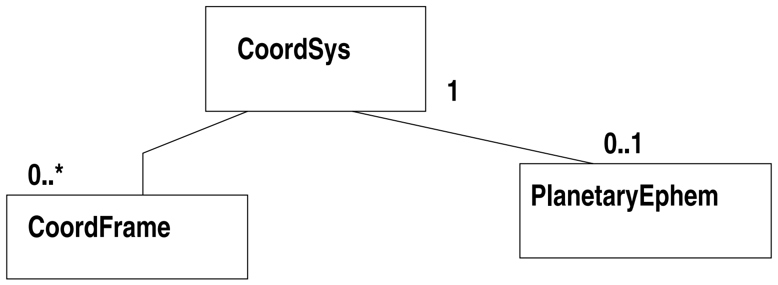
The STC model draws particular attention to the connection between time, spatial position, spatial velocity, redshift and spectral coordinates, whose definitions are intertwined.

In addition, the STC model envisages a separate instance of the STC object to describe the observation coverage (on the sky) and the observatory location.

3 Coordinate System

3.1 General CoordSys and CoordFrame

The CoordSys object consists of the aggregation of several CoordFrame objects, each for a separate set of axes. CoordFrame is subclassed for particular kinds of axis, such as Space, Time etc.



3.2 PlanetaryEphem

A coordinate system is considered to be absolutely defined in space if its coordinates can be transformed to the ITRF/ICRS geocentric standard reference frame. This means that if you are giving coordinates relative to Mars, you need to specify what assumptions you are making about the position of Mars. The CoordSys has a optional PlanetaryEphem object aggregated to it, which labels the planetary orbital ephemeris used; it should probably also provide a way to label the planetary physical ephemeris (pole and meridian versus time). Note that when barycenter-corrected times are given (see discussion on time below), the PlanetaryEphem is used to define what you mean by the solar system barycenter.

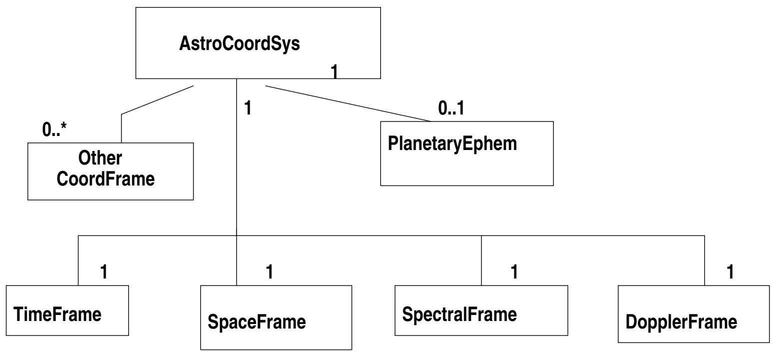
3.3 CoordFlavor

CoordSys also has a CoordFlavor saying how many spatial coordinate axes to expect, and whether the spatial coordinates are SPHERICAL, CARTESIAN, etc. [UCD?], and whether there are velocities in the data for each spatial axis.

I have argued to Arnold that the CoordFlavor is specific to the SpaceFrame derived class of CoordFrame, and does not belong in the larger CoordSys object.

3.4 AstroCoordSys

The CoordSys object specializes to an AstroCoordSys subclass which consists of at least CoordFrame objects for time, space, spectral, and doppler coordinates:



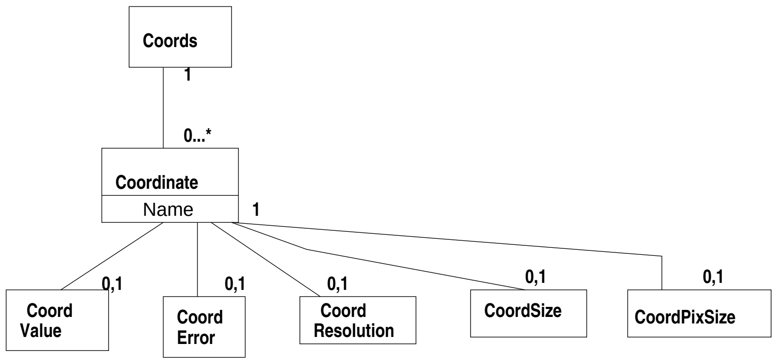
4 Coords and Coordinate classes

The general Coords class is an aggregation of individual Coordinate objects, which may include the specialized Time, Position, Velocity, Redshift and Spectrum coordinates. So a Coordinate instance describes one 'axis' (time, space,etc.) while a Coords instance describes all the axes at once.

In fact, the RedshiftCoord class is a totally generic Coordinate instance and is used for both the Redshift and Spectrum coordinates. The Velocity and Position are both described by a single SpatialCoord class. The TimeCoord and SpatialCoord specialized classes are discussed in detail later; first we describe the generic Coordinate as used for Redshift, Spectrum and arbitrary other coordinates.

4.1 General Coordinate object

The general Coordinate object is an aggregation of a CoordName and one or more of CoordValue, CoordError, CoordResolution, CoordSize, and CoordPixSize. What are all of these?



- CoordValue normally has a double for the value of the coordinate, plus a string for its unit. However the details of this value are handled specially for some subclasses such as time - see below.

- CoordError, with a value and a unit, is some kind of error on the coordinate. There is no discussion about whether the error includes systematic or statistical parts, and we can assume this class will be elaborated in the context of error modelling by the DM WG.

- CoordResolution, with a value and a unit, is in a sense a particular kind of CoordError, and represents a parameterization of the local smearing of the coordinate by the observation process. If the astronomical coordinate is x, the probability that the observing process will measure it as x' is $R(x, x')$. We can characterize the R function by calculating its full-width-half-max, R_{50}, which is the value in CoordResolution.

- CoordPixSize: again, a value and a unit. This object is only present if the Coordinate represents a quantized (pixellated or sampled) observation and its value is the extent in coordinate value of this quantization or pixellation. Let us consider a simple case of a 1D spectrum, with a regular pixel array where the Coordinate is wavelength and is being used to describe one of the pixels - the value might be "4002.5 Angstrom" and the pixel size might be "50 nm", indicating that the pixel runs from 4000.0 to 4005.0 Angstrom (Note that the pixel size and the value can be given in different units).

- CoordSize:

  I find this the most problematic part of the STC paradigm, and I express here my opposition to it. It actually represents a CoordArea expressed as a diameter - for instance, the size of a field of view in an observation or resource description. I completely don't understand it as an aggregation to the coordinate class, potentially together with pixel size, value and error. Arnold has suggested to me that it would be used as part of the description of an archive, I guess with the Coords object representing the resolution, pixel size and field of view of a typical field. I can't get past a mental block that the first two are a local (pixel level) property, even if constant for the field, but the FOV is really just the CoordArea object for the field, stashed in the Coords object for the survey. Part of this is probably because the CoordArea objects are not yet explicitly relocatable (they have absolute origins).

The intention of the Coords object is that any one instance will have only some of the member objects listed above. A Coords object describing the characterization of a dataset might only have the resolution and the error but not the value, while an instance describing an actual coordinate position might have the value but not the resolution.

5 CoordArea

The third and last main class in the STC is the CoordArea, describing the coordinate area/volume/extent taken up by an observation.

The CoordArea is an aggregation of arrays of Interval objects. Each coordinate can have such an array, including Time, Spatial, Velocity, Spectral and Redshift. Spatial, as usual, is a special 2-dimensional case and as well as simple intervals it can have arbitrary two-dimensional regions, which I won't summarize here.

The spectral and redshift coordinates are deemed to have simple scalar intervals consisting of start and stop value pairs (which are not required to have the same unit as each other!) and an interval type (lower limit closed or open, upper limit closed or open).

For velocity and spatial coordinates, the interval values can of course be multidimensional (remember astronomers, velocities are not just radial velocities!)

The time interval's coordinates are special time values as used in the time coordinate itself.

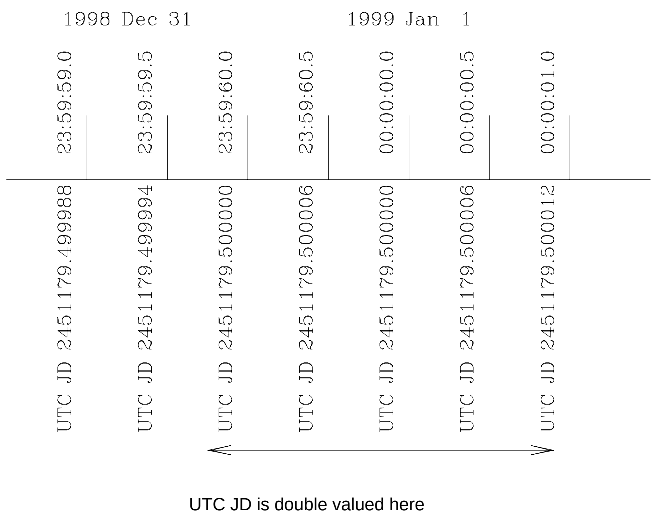
6 Time

6.1 Measuring time: instants and elapsed time

When talking about time, it's important to distinguish between the scalar quantity 'elapsed time between two instants' and the (essentially tuple-valued) 'label of an instant'.

Concepts like "Eastern Standard Time" and "Universal Time" apply to the labelling of instants rather than the elapsed time; thus the elapsed time between the instants "UTC 1998 Dec 31 23:59:59.5" and "UTC 1999 Jan 1 00:00:00.5" is 2.0 seconds, and so is the elapsed time between "TT 1999 Jan 1 00:01:02.5" and "TT 1999 Jan 1 00:01:04.5" which are the equivalent instants in the TT timescale. Sometimes I've heard people talk about "1 second of UTC" versus "1 second of TT" - there's no such thing. This leads to the worse error of calculating 86400 seconds * (difference in days), not counting the leap seconds, and calling this "elapsed UTC seconds", which is just Wrong. UTC, TT, EST, TAI, even arguably different calendars, all are alternate labellings of a single time variable, in contrast to concepts like TDB in which time runs at a different rate (TDB is defined in a different general relativistic frame). Note that we do not really use a zero point for timescales like UTC - yes, you could argue that 00:00 UTC on 1 Jan 1 BC counts as 'zero', but then is 01:00 UTC on 1 Jan 1 BC the zero of CET (European) time? I don't think in practice we use it like that. Instead, we map to a different representation - the JD.

We even have to be careful about Julian Day (JD) values. These look like a continuous floating-point measurement of elapsed time since the reference instant of BC(Old Style) 4713 Jan 1 00:00 in the particular timescale - but in the case of UTC, this is not the case because of leap seconds. The JD(UTC) corresponding to the two instants mentioned above are t1 = JD 2451179.4999942 and t2 = JD 2451179.5000058 UTC, and the subtraction doesn't work (86400*(t2-t1) = 1.0 not 2.0) because JD 2451179.5000058 is also the JD of "UTC 1998 Dec 31 23:59:60.5" - the JD(UTC) scale is doublevalued! This may seem a detail since it rarely is a significant effect, but it speaks directly to what a JD really is. Now it's true that JD(TT) is a single-valued, continuous coordinate.



6.2 Time Reference Position

There is a further complication in astronomy: we often talk not about the time an event happened, but the time an electromagnetic signal from the event reached a different position. There are several different ways this comes up:

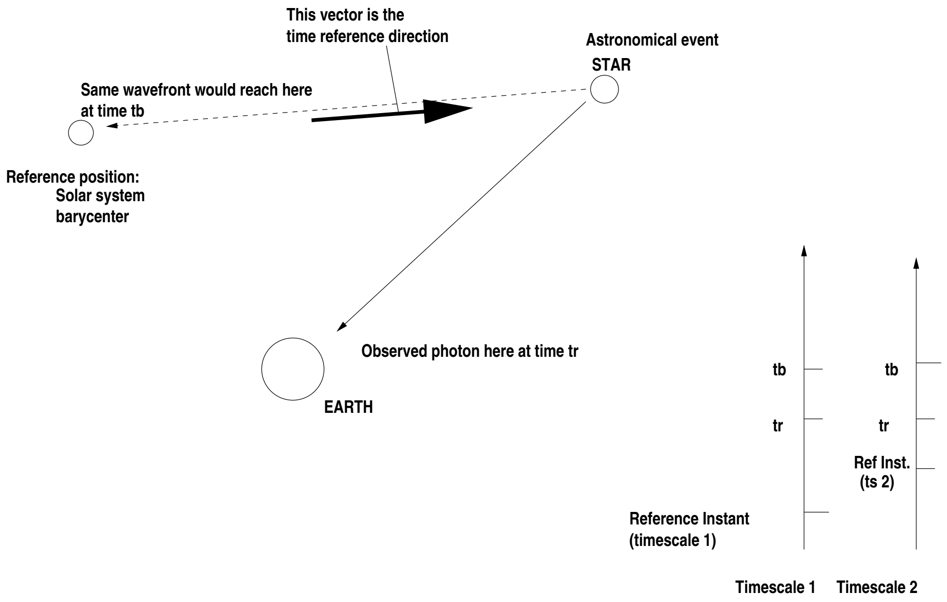
- The Spirit Rover lands on Mars, distance d away, at time t; the signal is received at JPL at time $t_r = t + d/c$, and we have to be careful when we quote the landing time to say whether we mean 'Spacecraft Event Time', t, or 'Earth Received Time', t+d/c.

- A star at a distance d from the Earth flares at time t; we detect photons from the flare on Earth at time $t_r = t + d/c$ (ignoring relativity effects). Because d is rarely known well, we almost always use t_r.

- We calculate the time that the flare would have been seen if we were sitting at the solar system barycenter, and the Sun were transparent and failing to fry us. This time, $t_b = t_r + \mathbf{r_e}.\mathbf{d}/c$ where $\mathbf{r_e}.\mathbf{d}$ is the component of the Earth-Sun vector in the direction of the source, is known as 'barycenter time'. In another source of confusion, it is often thought of as a *different timescale*, but it is actually *the time of a different event* (wavefront reaches barycenter, instead of wavefront reaches Earth) *in the same timescale.*

With that lead-in, we can understand that the time coordinate in STC represents the elapsed time in some frame, since some reference instant, that an electromagnetic signal from the event being discussed would have reached some reference position in the absence of absorption.

In simple use, the reference position and the observation position are the same and the time is just the time of observation.


This vector is the time reference direction
Astronomical event
STAR
Same wavefront would reach here at time tb
Reference position: Solar system barycenter
Observed photon here at time tr
EARTH
tb
tr
tb
tr
Ref Inst. (ts 2)
Reference Instant (timescale 1)
Timescale 1
Timescale 2

Here we illustrate the example of barycenter time; t_r and t_b are different instants, not the same instant on different timescales. As plotted, timescale 1 and 2 are in the same GR frame ($t_b - t_r$ is the same in both).

In the case of an astronomical observation of a photon wavefront, the electromagnetic signal actually exists but may or may not reach the TimeRef-Position - few such signals will arrive at the barycenter! In the case of some other kind of event being timed such as the position of the Chandra satellite in its orbit, the signal may not even exist but the corresponding time correction is still well-defined.

We also need to record the TimeReferenceDirection, the unit vector from the TimeReferencePosition to the source of the wavefront, so that we can back out the correction if needed.

6.3 Absolute and Relative Time classes

The two kinds of time in the model are AbsoluteTime and RelativeTime. Both kinds include the label of a reference instant; in the case of AbsoluteTime the time is the instant in question, while for RelativeTime the real-valued elapsed time since the instant is provided.

Since AbsoluteTime is identical to a RelativeTime with an ElapsedTime value of zero, I don't see the need for two separate subclasses.

6.4 TimeFrame

The TimeFrame consists of the TimeScale (to define the frame), and the TimeRefPosition, an instance of the ReferencePosition class (to define the reference position), and the TimeRefDirection.

- The TimeScale has values like TT, TDT, ET, TCG, UTC ... [UCD?]
- The ReferencePosition is either a labelled position, called a StdRefPosition, with values like BARYCENTER, TELESCOPE, GEOCENTER... [UCD?] or is a positional coordinate together with a spatial coordinate frame (assumed in the STC object to be the same as the main STC spatial coordinate frame).
- the TimeRefDirection is a Coords object and defines the direction of the source used in calculating the reference position correction, if the reference position is not TELESCOPE.

QUESTION to Arnold: Doesn't the ObservatoryLocation fit here?

6.5 TimeCoord

The TimeCoord is derived from the generic CoordinateClass; only the Value part of it is different.

The AbsoluteTime class defines a Timescale (repeated from the TimeFrame) and a time representation, either ISOTime, JDTime or MJDTime. This provides a way to describe a time instant.

The RelativeTime class consists of an AbsoluteTime used as a zero point and a scalar value/unit giving the elapsed time from that zero point.

7 Space

7.1 SpaceFrame

The SpaceFrame has a ReferencePosition instance giving the standard of rest and an origin. The ReferencePosition is described under Time.

- SpaceFrame also has a SpaceRefFrame, which can be a CustomFrame or a StdFrame. The CustomFrame defines the X and Z axis vectors in terms of another coordinate system via Coords objects, while the StdFrame defines the axes via a named system RefSystem and a StdPole. The SpaceRefFrame also can have a CoordEquinox (value e.g. J2000.0 or B1950.0) to allow FK4/FK5 equatorial and ecliptic sky frames to be parameterized in the old standard way.

  I think the CoordEquinox object belongs on RefSystem, as it only applies in the case of FK4/FK5.

- Allowed values for RefSystem: ICRS, FK5, FK4, SPHERICAL_BODY, AZ_EL. [UCD?]

  I don't like the name SPHERICAL_BODY; you really mean any body where we use a spherical coord system - (433) Eros is an example where the body itself is very non spherical.

- Allowed values for StdPole: EQUATORIAL, ECLIPTIC, SUPERGALACTIC, GALACTIC. [UCD?]

  I think we should include an option PLANETOGRAPHIC and PLANETOCENTRIC here. EQUATORIAL with equinox of date is equivalent to PLANETOCENTRIC when the reference position is the geocenter.

  I don't like the name StdPole, since this metadata also specifies a longitude zero point, not just the pole direction.

For AZ/EL, you need the origin (e.g. observatory) location. Seems to me that this should be just RefSystem = Earth together with a CustomFrame that gives the Pole_Zaxis in lat/long coords (instead of just allowing StdPole with RefSystem). Some STC docs say "StdRefPosition must be TELESCOPE for data in AZ_EL systems". I can imagine data in AZ_EL coordinates relative to things other than a telescope aperture.

7.2 SpatialCoord

The first difference between a SpatialCoord and a generic Coordinate is the presence of a Dimensionality class containing a single attribute n, the dimensionality of the coordinate (often 2 in astronomy, but sometimes 3 and in relativistic applications plausibly 4; hopefully it will be a while before the value $n = 11$ is required).

The other main difference is that the values in the Value, Error, Resolution, PixSize objects are themselves two-dimensional. For the main Value, the representation is called a CValue, which is either a simple scalar n-tuple or a Value60, representing a pair of numbers in sexagesimal format (only if $n = 2$, and probably not for a velocity coordinate - we measure proper motion in arcseconds/yr, but don't represent it in sexagesimal format).

It seems to me that the Value60 as currently described is a description of an external representation. This is somewhere where the UML and the XML perhaps part company. The underlying model of sexagesimal data should be a tuple of values (it does make sense sometimes to hold things as a tuple rather than the equivalent floating point value, to avoid rounding error problems during conversion). The particular mapping of the tuple to a string (with colons? with spaces? etc) should be a method on the Value60, rather than the string representation being the fundamental attribute.

For Error, Resolution and PixSize, where extent is being described, the values are MultiCoords, which are like CValues with the extra option that they can be an $n \times n$ matrix, or else can be an n-tuple with a position angle, representing an ellipse.

8 Spectral coordinates, Redshift and Velocity

- As mentioned above, the Spectral and Redshift coordinates are simple Coordinate instances with just scalars everywhere. The Velocity coordinate is a SpatialCoord with tuples.
- The Spectral frame is simply defined with a StdRefPosition defining the standard of rest (see discussion in Time above).
- The Doppler frame is like the Spectral frame but also has the aggregation of a DopplerDefinition (optical, radio or relativistic) and a type attribute saying whether velocities or redshifts are given. [UCD?]

  Both of these pieces of metadata seem to me like they would be better handled using UCDs, allowing us to unify the Doppler and Spectral frame concepts.

Note that the standard of rest for the Doppler frame can be different from the standard of rest for the Spectral frame and for the Spatial frame; one might have true velocity, doppler velocity and wavelengths all measured in different rest frames in a single STC object.

The potential confusion if there is for some reason more than one Spectral or Spatial or Doppler frame (a dataset with multiple positions in different axes?) leads me to feel that the generic CoordSys object will need an explicit ordering ('in this CoordSys instance, the third axis has Spatial frame S1, the fourth axis has Doppler frame D1, and the fifth axis has Spatial frame S2') rather than simple aggregation. Arnold thinks this should be handled by multiple CoordSys instances.

In the DopplerFrame and Spectral Frame, I'm missing a way to define 'using a standard of rest at z=1.4'.