



Space-Time Coordinate Metadata for the Virtual Observatory

Version 1.0

IVOA WG Internal Draft 2004-07-16

Working Group:

<http://www.ivoa.net/twiki/bin/view/IVOA/IvoaDataModel>

Authors:

Arnold Rots – Harvard-Smithsonian Center for Astrophysics

Abstract

This document provides a complete design description of the Space-Time Coordinate metadata for the Virtual Observatory. It explains the various components, highlights some implementation considerations, presents a complete set of UML diagrams, and discusses the relation between STC and certain other parts of the Data Model.

Status of this document

This is a Working Group Internal Draft

Acknowledgements

I gratefully acknowledge many helpful discussions with Wil O’Mullane, Alex Szalay, Jonathan McDowell, Ray Plante, Ed Shaya, Pat Dowler, and many others. I am indebted to SAO, ULP, CDS, and the NSF NVO grant for support.

Table of Contents

1. Introduction.....	3
2. Justification and Scope.....	3
3. Requirements and Use-Context	4
4. Design.....	5
4.1 The Metadata Components	5
4.2 Coordinate System (CoordSys).....	6
4.2.1 Coordinate Frame (CoordFrame)	6
4.3 Coordinates (Coords).....	6

STC Metadata for the VO

4.3.1 Coordinate.....	6
4.4 Coordinate Area (CoordArea).....	7
4.4.1 ScalarInterval.....	7
4.5 AstroCoordSys	8
4.5.1 TimeFrame	8
Table 1. Standard Reference Positions	8
Table 2. Timescales	9
4.5.2 SpaceFrame	10
Table 3. Standard Reference Frames.....	11
4.5.3 SpectralFrame.....	13
4.5.4 RedshiftFrame	13
4.6 AstroCoords	13
4.6.1 TimeCoord	14
4.6.2 SpatialCoord.....	14
4.6.3 SpectralCoord.....	15
4.7 AstroCoordArea	15
4.7.1 TimeInterval.....	16
4.7.2 SpatialInterval	16
4.7.3 VelocityInterval.....	16
4.7.4 SpectralInterval.....	16
4.7.5 RedshiftInterval	16
4.8 Region.....	17
4.8.1 Shapes	17
4.8.2 Operations	18
5. Projection.....	18
6. Conclusion and Usage Notes	19
6.1 Methods	19
6.2 Usage	19
6.2.1 Resource Profile	19
6.2.2 Query Constraint.....	20
6.2.3 Catalog Entry.....	20
6.2.4 Observational Data	20
7. Implementation	20
7.1 XML Schema Implementation.....	20
7.2 Simplifications	21
7.2.1 Descriptions.....	21
7.2.2 Parsers.....	21
Appendix A – UML Representation	21
Appendix B – Examples.....	31
B.1 STCResourceProfile	31
B.2 Galaxy Catalog	32
B.3 ROSAT Image.....	34
B.4 Blue M81 Image from KPNO	36
B.5 Query Specification	38
Appendix C - XInclude Library of Coordinate Systems.....	40
Appendix D - Changes from Previous Versions.....	40

1. Introduction

This document attempts to explain and document the design and implementation of the Space-Time (and related coordinate axes) metadata for the Virtual Observatory. In order to make this specification optimally useful for the VO, we have generalized the Coordinate System concept and we have added a generic Coordinate Frame. Derived from these is the AstroCoordSys and the time, space, redshift, spectral frames that together make up the STC coordinate system. Coordinate and Coordinate Area also have generic as well as Astro versions. There are five companion files to this document:

- http://hea-www.harvard.edu/~arots/nvometa/STC_UML.pdf
presents the STC metadata design described in this document as UML diagrams; it is also reproduced in the appendix to this document.
- <http://www.ivoa.net/xml/STC/v1.0>
<http://hea-www.harvard.edu/~arots/nvometa/stc.xsd>
is the top-level schema of the XML implementation, containing the coordinate system and coordinate area definitions.
- <http://www.ivoa.net/xml/STCcoords/v1.0>
<http://hea-www.harvard.edu/~arots/nvometa/coords.xsd>
is the schema of the XML implementation with the coordinate value definitions.
- <http://www.ivoa.net/xml/STCregion/v1.0>
<http://hea-www.harvard.edu/~arots/nvometa/region.xsd>
is the schema of the XML implementation for the *region* specification.
- <http://hea-www.harvard.edu/~arots/nvometa/STCPoster.pdf>
is a poster showing a slightly outdated, but still useful, version of the entire STC schema.

The schemas are now fully up-to-date.

In the following sections we shall discuss the justification and scope for this metadata specification; the requirements and use context; a detailed design description; and some implementation considerations. Some concepts are only provided in their simplest form and we expect that more sophisticated classes will be derived (and will fit in seamlessly) as work progresses. The emphasis of this document is on the framework of the STC metadata.

Note:

In order to simplify the XML schemata, we decided as of version 0.5 to forego the capability of specifying Units for each component of the Coordinate element (see Sections 4.3.1, 4.4, and 4.6) and the option of specifying spatial position coordinates in sexagesimal format (see sections 4.6.2 and 4.6.2.3).

2. Justification and Scope

Before we plunge into the requirements and design of these metadata structures, it may be helpful to ask what is driving this exercise and to provide some context to the design. Our intent in this is to explain why the design needs to be fully general and extensible, leading to its alleged complexity.

The basic justification for considering the following coordinate axes in a single metadata specification is that they are closely intertwined:

- Space (including its time-derivatives, i.e., velocities)
- Time
- Spectral (frequency, wavelength, energy)
- Redshift (Doppler velocity)

It should be emphasized that the STC design distinguishes between the spectral coordinate and the redshift coordinate (one might, after all, have to deal with a dataset containing Doppler velocities measured in different parts of the spectrum) and between redshifts and velocities (the latter are physical quantities – derivatives of spatial positions; the former represent a spectral property that is transformed via one of several formalisms to what is commonly referred to as a Doppler velocity – when we get to relativistic situations the connection with physical velocities becomes tenuous).

We shall refer to these as the *STC coordinates*. The issue is that time is bound to a position and positions are time-variable. Similarly, spectral and redshift data are tied to reference frames that may or may not be time-variable.

In the past, most data archives have not been extremely concerned with dotting every i and crossing every t , in a meticulous specification of all the details of all coordinate axes. They assumed sets of coordinate system-related defaults. Such context-dependent defaults are quite acceptable: issues are generally well-defined, obvious, and clear for single-observatory observations, even when not all is explicitly specified. However, there are no global defaults in the VO. All implicit assumptions need to be made explicit since they will not be “obvious” anymore. One must be able to transform the coordinates of two observations to a common coordinate system; this means that every little tidbit of information needs to be documented in the metadata.

Here are some simple examples of situations and questions that need an answer, when such context-dependent defaults do not work anymore:

- Are Doppler velocities optical or radio?
- What is the spatial reference position for the time stamps; if corrected for pathlength, what direction was assumed?
- Does “J2000” mean “FK5/J2000” or “ICRS”?
- At which spatial position was this observation made?
- How can we get information on Galactic background objects from solar system observations, and vice versa?

3. Requirements and Use-Context

The top-level requirement for the design of the Space-Time Coordinate metadata can be formulated in a very simple way. It is that they:

- provide sufficient and necessary information
- are self-consistent

STC Metadata for the VO

That is really all. However, to fit it into the larger VO Data Model context, we should leverage two more requirements:

- the system should be extensible to include other coordinates and accommodate coordinate metadata descriptions that include axes other than the STC ones
- it must be possible to define relocatable frames to accommodate, for instance, simulation data

From this one can derive secondary requirements: the metadata must include complete descriptions of the following three concepts:

- the coordinate frames that anchor the STC coordinate axes, collectively referred to as the *Coordinate System*
- a specific position in these frames, i.e., a coordinate object, that includes not only coordinate values, but also essential attributes such as errors and resolutions
- the volume in coordinate space that is taken up by the data object that these metadata are attached to

In all, there are four different contexts for the use of the STC metadata:

- STC Resource profile – to specify the spatial, temporal, redshift, spectral coverage of a data collection which is ultimately used by clients to determine whether or not the resource is of interest for their purposes
- Query – to specify the STC constraints of a particular request
- Catalog information – clearly, the response to a catalog query may contain coordinate information (e.g., positions), but it should also contain coverage information, specifically, the intersection of the catalog's coverage and the query's STC constraints
- Observational data – here we need the volume in coordinate space that is occupied by the observational data, as well as the coordinates of the observatory; note that these may not necessarily be given in the same Coordinate System

We will return to the use of STC in these contexts in Section 6. As commented before, the emphasis of this document is on the structure of the STC metadata. Hence, though the classes and attributes are laid out in detail, little is said about methods.

4. Design

It may be helpful to refer to the UML diagrams in the Appendix, while reading this section.

4.1 The Metadata Components

There are three metadata components in the design, as already noted in the derived requirements:

- **Coordinate system**
This object contains the coordinate frames that together define a complete coordinate system. There may be one or more per document.

- **Coordinate values**
This is an object that specifies a specific position in the space defined by the coordinate system it refers to. The object contains values, units, errors, resolution, size, and pixel size, though not all of these need be present.
- **Coordinate areas or ranges**
This is an object that defines a volume in the coordinate space specified by the coordinate system it refers to. Regions are a special case, specifically designed for 2-D (and 3-D unit-sphere) spatial coordinates.

4.2 Coordinate System (CoordSys)

The CoordSys class is nothing more than a collection of one or more Coordinate Frames, where each frame describes a set of one or more coordinate axes that belong together.

4.2.1 Coordinate Frame (CoordFrame)

A CoordFrame object provides the metadata on one or more coordinate axes. Usually, we will be dealing with one axis. The exception is the spatial frame which may control up to three position coordinate axes, as well as their time-derivatives (velocity). CoordFrame is really the base class for coordinate frames. To ensure its full generality, this base class should only contain a name. Real-life coordinate frames should be derived from this class.

4.3 Coordinates (Coords)

The Coords object specifies a particular position in coordinate space. It requires a reference to a CoordSys object and it is made up of one or more Coordinate objects, corresponding to the Coordinate Frames defined in the Coordinate System. All objects are optional. However, in order to constitute a meaningful Coords object, one would expect at least one of the Coordinate objects to be present.

4.3.1 Coordinate

A Coordinate is an aggregate of up to 6 objects: CoordName, CoordValue, CoordError, CoordResolution, CoordSize, and CoordPixSize. Of these, only CoordName is required to be present. The others, which in their simplest forms consist of a scalar numerical Value, are all optional, though one may doubt whether a Coordinate object that contains none of these has any meaning. Although at first sight it would seem that at least CoordValue is indispensable, we shall show that there are legitimate cases where the value may be absent. Coordinate has a single Unit attribute that applies to all components. Classes that are derived from Coordinate may differ in the data types of the objects, their structure and meaning, as well as restrict allowable values for Unit. [Initially we intended allowing each component to have its own Unit, the rationale being that this is the way users treat them, a practice that is often followed in catalogs; for instance, while positional coordinate values are usually expressed in degrees, their errors and sizes are commonly quoted in arcseconds.]

STC Metadata for the VO

4.3.1.1 CoordName

This required member of `Coordinate` is a simple string, presumably taken from a limited enumerated list. We may want to consider using UCDs for the Name.

4.3.1.2 CoordValue

The value of the `Coordinate` consists of a scalar numerical or string value and a Unit string.

4.3.1.3 CoordError

Again, this consists of a scalar Value, representing an rms error. However, we expect a variety of derived classes to appear with more sophisticated error descriptions. And clearly, the multi-dimensional coordinates require more information.

4.3.1.4 CoordResolution

The resolution along the `Coordinate` is expressed as a FWHM Value. In cases where this is a poor description, one should derive a more sophisticated class.

4.3.1.5 CoordSize

The size along the `Coordinate` is also expressed as a FWHM Value, though one would expect derived classes with more precise definitions (e.g., Holmberg diameter). This object is used in three contexts: in a resource profile it indicates the typical size (say, FOV) of datasets in the resource; in queries it serves the same purpose; and in catalog datasets it allows the server to provide `Coordinate` positions of objects as well as their sizes.

4.3.1.6 CoordPixSize

The pixel size (scalar Value) is often a useful figure to characterize resources, queries, and datasets, even if it only provides a typical value.

4.4 Coordinate Area (`CoordArea`)

The `Coordinate Area` class defines the volume in coordinate space that is occupied by the object it is attached to. It is an aggregate of one or more ranges in individual coordinates and is required to have a reference to a `CoordSys`. Multiple ranges along each coordinate axis are OR-ed together; the ranges of different axes are logically AND-ed together.

4.4.1 `ScalarInterval`

A `ScalarInterval` consists of a `LoLimit` and/or a `HiLimit`, both of which contain a double value and a unit. Note that they do not both have to be present, but at least one of them needs to be. `ScalarInterval` has Boolean attributes `LoLimInclude` and `HiLimInclude` and a `FillFactor` attribute with a value between 0 and 1 that indicates what fraction of the interval is really covered.

4.5 AstroCoordSys

The AstroCoordSys is derived from CoordSys. It still is a collection of coordinate frames, but it is required to contain specific frame classes derived from the CoordFrame base class, in addition to zero or more additional (generic) frames. The required frames are TimeFrame, SpaceFrame, and SpectralFrame; DopplerFrame is optional.

4.5.1 TimeFrame

The Time Frame contains two or three objects: a reference position, a time scale, and, optionally, a time reference direction.

4.5.1.1 ReferencePosition

The reference position, or spatial coordinate origin, may be chosen from a standard list of such origins (StdRefPosition) or specified as a particular coordinate position (CoordOrigin). In the latter case, one may nest the origins, but eventually they should be referenced to a known standard origin. The list of standard origins includes (see Table 1): GEOCENTER, BARYCENTER, HELIOCENTER, TOPOCENTER, LSR, GALACTIC_CENTER, SUPER_GALACTIC_CENTER, EMBARYCENTER, MOON, MERCURY, VENUS, MARS, JUPITER, SATURN, URANUS, NEPTUNE, PLUTO, RELOCATABLE; additional values (e.g., planetary satellites, non-planet solar system objects) are allowed, provided they are identified in a referenced ephemeris or other authoritative source (see: Section 4.5.2.4). Ultimately, one must be able to tie the coordinate origin down to the geocenter, be it through a geographic position, an IAU resolution, an orbit ephemeris, or a solar system ephemeris – unless one deals with simulation data that have a RELOCATABLE origin. Planetary Ephemeris is required for any position related to a solar system entity other than the geocenter. ReferencePosition is a standard class that is being used for all four STC frames, though there are different restrictions for the four usages. For the Time Frame LSR and RELOCATABLE are not allowed values; for simulations one may want to use a custom position at the origin of the relocatable spatial frame.

Table 1. Standard Reference Positions

Reference Position	Description	Comments
GEOCENTER	Center of the earth	
BARYCENTER	Center of the solar system barycenter	
HELIOCENTER	Center of the sun	
TOPOCENTER	“Local”; in most cases this will mean: the location of the telescope	
LSR	Local Standard of Rest: 19.5 km s^{-1} in the direction of FK4(B1900.0:270,+30) or GALACTIC_II(56,+23)	Only to be used for redshifts and Doppler velocities
GALACTIC_CENTER	Center of the Galaxy	

STC Metadata for the VO

Reference Position	Description	Comments
SUPER_GALACTIC_CENTER	Center of the Local Super Cluster	At the center of the Virgo Cluster, i.e., M87
EMBARYCENTER	Earth-moon barycenter	
MOON	Center of the moon	
MERCURY	Center of Mercury	
VENUS	Center of Venus	
MARS	Center of Mars	
JUPITER	Center of Jupiter	
SATURN	Center of Saturn	
URANUS	Center of Uranus	
NEPTUNE	Center of Neptune	
PLUTO	Center of Pluto	
RELOCATABLE	Relocatable center; for simulations	Only to be used for spatial coordinates

4.5.1.2 TimeScale

The time scale must be chosen from among the list recognized by the IAU (see Table 2): TT, TDT, ET, TAI, IAT, UTC, TDB, TCG, TCB, LST. TT is the default; TDT and ET are obsolete synonyms for TT; IAT is an unofficial synonym for TAI; and use of LST is to be discouraged. At some point other time scales may need to be added as they become recognized, such as planet- or moon-based time scales. For now we will also allow the value LOCAL, only to be used with RELOCATABLE space frames (see: Section 4.5.1.1). For a handy explanation of time scales, see: <http://tycho.usno.navy.mil/systime.html>

Table 2. Timescales

Timescale	Description	Comments
TT	Terrestrial Time	
TDT	Terrestrial Dynamic Time; synonym for TT	
ET	Ephemeris Time: predecessor of, and continuous with, TT	
TAI	International Atomic Time; 32.184 s behind TT	
IAT	Synonym for TAI	
UTC	Coordinated Universal Time; 32 s behind TAI in 2004	Includes leap seconds

Timescale	Description	Comments
TDB	Barycentric Dynamical Time; synchronous with TT, except for variations in earth's orbital motion	
TCG	Geocentric Coordinate Time; properly relativistic time, running about a factor 10^{-10} slower than TT	
TCB	Barycentric Coordinate Time; properly relativistic time, running about a factor 10^{-8} slower than TDB	
LST	Local Siderial Time	Ground-based observations only
LOCAL	"Local" time	Only to be used for simulations, in conjunction with RELOCATABLE spatial coordinates

4.5.1.3 TimeRefDirection

If the Time Frame's Reference Position is not TOPOCENTER, times have clearly been transformed from observed time at the location where the observation was made to some other spatial location. In order to effect such a transformation, a direction of origin must have been assumed for the observed phenomenon; that direction is given in TimeRefDirection.

4.5.2 SpaceFrame

The Space Frame contains a reference position, a reference frame, and a coordinate flavor object.

4.5.2.1 ReferencePosition

This object is of the same class as described in Section 4.5.1.1, with the following restrictions. The value RELOCATABLE is allowed, but LSR is not. Obviously, if an explicit coordinate origin is provided, those coordinates should be referenced to a different CoordSys; one may still nest them, but eventually they should be referenced to a known standard origin.

4.5.2.2 CoordFlavor

This class is peculiar to spatial coordinates. It indicates the intrinsic dimensionality of the coordinate system (1, 2, or 3), and the type of coordinates: CARTESIAN, SPHERICAL, UNITSPHERE, or GEOGRAPHIC (longitude, latitude, and altitude). For this last one we allow the special Units value "deg deg m". This list will likely be expanded with, for instance, polar and cylindrical coordinates.

4.5.2.3 SpaceRefFrame

While the Reference Position pins down the origin of the spatial coordinate system, the Space Reference Frame specifies its orientation. In most cases this will be a StdFrame, consisting of a RefSystem (FK4, FK5, ECLIPTIC, ICRS, GALACTIC_I,

STC Metadata for the VO

GALACTIC_II, SUPER_GALACTIC, AZ_EL, BODY; the first three require a CoordEquinox). The Standard Frame may be BODY if the Reference Position is specified as a solar system object. In addition, we allow solar, lunar, planetary, and planetary satellite coordinate frames (planetocentric and planetographic) as defined by Sections 7.2-7.5 of the *Explanatory Supplement to the Astronomical Almanac* (University Science Books 1992, Ed. P. K. Seidelmann); see also Fränz & Harper (2002, Plan & Sp Sci 50, 217). See Table 3.

Alternatively, a CustomFrame may be specified by way of the pole (Z) axis and the longitude zero-point (X-axis) directions.

Table 3. Standard Reference Frames

Reference Frame	Description	Comments
FK4	Fundamental Katalog, system 4; Besselian	Requires Equinox; default B1950.0
FK5	Fundamental Katalog, system 5; Julian	Requires Equinox; default J2000.0
ECLIPTIC	Ecliptic coordinates	
ICRS	International Celestial Reference System	
GALACTIC_I	Old Galactic coordinates	
GALACTIC_II	“New” Galactic coordinates	
SUPER_GALACTIC	Super galactic coordinates; pole at GALACTIC_II(47,+5)	
AZ_EL	Local azimuth and elevation	Ground-based observatories
BODY	Generic “BODY” coordinates	
GEO	Geographic coordinates	
GEOD	Geodetic coordinates	
MAG	Geomagnetic coordinates	
GSE	Geocentric Solar Ecliptic coordinates	
GSM	Geocentric Solar Magnetic coordinates	
SM	Solar Magnetic coordinates	
HGC	Heliographic coordinates	See Explanatory Supplement, Section 7.2
HEE	Heliocentric Earth Ecliptic coordinates	
HEEQ	Heliocentric Earth Equatorial coordinates	
HCI	Heliocentric Inertial coordinates	
HCD	Heliocentric coordinates of Date	
MERCURY_C	Planetocentric coordinates on Mercury	See Explanatory Supplement, Section 7.4

STC Metadata for the VO

Reference Frame	Description	Comments
VENUS_C	Planetocentric coordinates on Venus	See Explanatory Supplement, Section 7.4
LUNA_C	Lunacentric coordinates	See Explanatory Supplement, Section 7.3
MARS_C	Planetocentric coordinates on Mars	See Explanatory Supplement, Section 7.4
JUPITER_C_III	Planetocentric coordinates on Jupiter, system III	See Explanatory Supplement, Section 7.4
SATURN_C_III	Planetocentric coordinates on Saturn, system III	See Explanatory Supplement, Section 7.4
URANUS_C_III	Planetocentric coordinates on Uranus, system III	See Explanatory Supplement, Section 7.4
NEPTUNE_C_III	Planetocentric coordinates on Neptune, system III	See Explanatory Supplement, Section 7.4
PLUTO_C	Planetocentric coordinates on Pluto	See Explanatory Supplement, Section 7.4
MERCURY_G	Planetographic co-rotating coordinates on Mercury	See Explanatory Supplement, Section 7.4
VENUS_G	Planetographic co-rotating coordinates on Venus	See Explanatory Supplement, Section 7.4
LUNA_G	Lunagraphic co-rotating coordinates	See Explanatory Supplement, Section 7.3
MARS_G	Planetographic co-rotating coordinates on Mars	See Explanatory Supplement, Section 7.4
JUPITER_G_III	Planetographic co-rotating coordinates on Jupiter, system III	See Explanatory Supplement, Section 7.4
SATURN_G_III	Planetographic co-rotating coordinates on Saturn, system III	See Explanatory Supplement, Section 7.4
URANUS_G_III	Planetographic co-rotating coordinates on Uranus, system III	See Explanatory Supplement, Section 7.4
NEPTUNE_G_III	Planetographic co-rotating coordinates on Neptune, system III	See Explanatory Supplement, Section 7.4
PLUTO_G	Planetographic co-rotating coordinates on Pluto	See Explanatory Supplement, Section 7.4

4.5.2.4 Planetary ephemeris (if needed)

The planetary ephemeris object `PlanetaryEphem` indicates which solar system ephemeris was used in the `AstroCoordSys`. In general, this will be JPL's DE200 or DE405 (default). It should only be present if a solar system ephemeris was used, for instance because of planet-based coordinate systems or the application of barycenter corrections. In addition, the information in Sections 7.2-7.5 of the *Explanatory Supplement to the Astronomical*

Almanac (University Science Books 1992, Ed. P. K. Seidelmann) may be considered a known authoritative source for the purpose of spatial reference frames. If the SpaceRefFrame (see: Section 4.5.2.3) is defined on a body, ephemeris information on its coordinate system (pole, primary meridian) must be included in the PlanetaryEphem.

4.5.3 SpectralFrame

The Spectral Frame only contains a reference position.

4.5.3.1 ReferencePosition

This object is of the same class as described in Section 4.5.1.1, without any restrictions.

4.5.4 RedshiftFrame

The Redshift Frame contains two objects: a reference position and a Doppler definition object.

4.5.4.1 ReferencePosition

This object is of the same class as described in Section 4.5.1.1, without any restrictions.

4.5.4.2 DopplerDefinition

This object specifies what the definition of redshift is and how it should be translated to Doppler velocity. Allowed values are OPTICAL, RADIO, and RELATIVISTIC.

$$\text{OPTICAL:} \quad V_{opt} = c \cdot \frac{\Delta l}{l_0} = -c \cdot \frac{\Delta n}{n}$$

$$\text{RADIO:} \quad V_{rad} = -c \cdot \frac{\Delta n}{n_0} = c \cdot \frac{\Delta l}{l}$$

$$\text{RELATIVISTIC:} \quad V_{rel} = c \cdot \frac{l^2 - l_0^2}{l^2 + l_0^2} = -c \cdot \frac{n^2 - n_0^2}{n^2 + n_0^2}$$

It should be emphasized that Doppler velocities are formal velocities; i.e., defined by a formalism and not necessarily physical in nature. One should also note here that RELATIVISTIC is not strictly correct and we should, at some point, probably move toward compliance with the third FITS WCS paper.

4.6 AstroCoords

The AstroCoords object specifies a particular position in STC coordinate space. It requires a reference to an AstroCoordSys object and it is made up of 1 to 5 Coordinate objects, at most one each of the following:

- Time of class TimeCoord
- Position of class SpatialCoord
- Velocity of class SpatialCoord
- Redshift of class SpectralCoord
- Spectrum of class SpectralCoord

Note that all objects are optional. However, in order to constitute a meaningful AstroCoords object, one would expect at least one of them to be present.

The three specialized STC Coordinate classes, TimeCoord, SpatialCoord, and SpectralCoord, are all derived from a generic base-class Coordinate (see: Section 4.3.1). Instead of providing the actual coordinate elements, the object may refer to a FITS file that provides some or all of them.

4.6.1 TimeCoord

The error, resolution, size, and pixel size components of this class (derived from Coordinate) are fully generic, i.e., consisting of a scalar value (double) and unit string. The allowed time units are 's' (second), 'd' (day = 86400 s), 'yr' or 'a' (Julian year = 365.25 d), 'cy' (Julian century = 36525 d).

The CoordValue object in TimeCoord is of class AstronTime.

4.6.1.1 AstronTime

A time (i.e., a particular instant in history) is an aggregate of two or three objects: a TimeScale (Section 4.5.1.2), an AbsoluteTime, and, optionally, a RelativeTime (or ElapsedTime). Note that, in principle, all times are really elapsed times. The difference is that for absolute times there is an implicit agreement as to what the reference instant in time is.

4.6.1.1.1 AbsoluteTime

AbsoluteTime contains is instantiated as one of its four subclasses:

1. ISOTime: a time instant expressed in a subset of the ISO-8601 format; i.e., a string of format `yyyy-mm-ddThh:mm:ss.sss...`
2. JDTime: a decimal number (note: in principle unlimited precision!) representing the Julian Day of a particular instant.
3. MJDTime: a decimal number (note: in principle unlimited precision!) representing the Modified Julian Day of a particular instant ($MJDTime = JDTime - 2,400,000.5$).
4. TimeOrigin: a string with the value RELOCATABLE, primarily for simulation work, to be used in conjunction with TimeScale LOCAL and a RELOCATABLE SpaceFrame.

4.6.1.1.2 RelativeTime

Relative time is a decimal number representing the amount of time elapsed since AbsoluteTime.

4.6.2 SpatialCoord

Spatial coordinates are special in that they can be explicitly multi-dimensional and that they serve positions as well as their first time derivatives (physical velocities – not to be confused with redshifts or Doppler velocities). Not only does this mean that scalar values are to be replaced by vectors, but components like errors, resolutions, and sizes become more complicated. SpatialCoord is an aggregate of CoordName, Dimensionality, CoordValue (class SCValue), CoordError, CoordResolution, CoordSize, CoordPixSize (all class MultiCoord). We stress that Dimensionality should force all components of SpatialCoord to be consistent in type and content.

It may well be that there is merit in sub-classing a VelocityCoord class, since sexagesimal formats are not needed and a time unit needs to be added. Either way, we suggest that the units for velocity be the same as for position, but that a (standard) time unit be added for the denominator.

4.6.2.1 Dimensionality

A single object of this class controls all components of the SpatialCoord class by specifying how many coordinate axes need to be represented. Note that this is distinct from the CoordFlavor object described in Section 4.5.2.2: CoordFlavor specifies the number of spatial axes defined in the AstroCoordSys; Dimensionality specifies how many of those are actually provided in the AstroCoords object – one could envision a catalog with only Right Ascensions, where the CoordFlavor would be spherical, 2-dimensional, but the dimensionality just be 1.

4.6.2.2 CoordName

This object now contains a list of <Dimensionality> strings with the names of all axes present.

4.6.2.3 SCValue

The spatial coordinate value is a vector (length determined by Dimensionality) of decimal values (SValue).

4.6.2.4 MultiCoord

Expressing errors and sizes for vectors is slightly more complicated. If the errors are independent for each vector component, then a simply SValue object (vector of decimal values with a Unit) will do. If they are not independent (or if the resolution is tilted), one can either add a PosAngle to the SValue object, or specify the MultiCoord object through a Matrix of decimal values with a Unit.

4.6.3 SpectralCoord

The SpectralCoord is used for the spectral as well as the redshift components of the AstroCoords object. It is a completely generic Coordinate class (see: Section 4.3.1) where Name is a string and all other components consist of a double Value and a string Unit.

4.7 AstroCoordArea

The Coordinate Area class defines the volume in coordinate space that is occupied by (or: the coverage of):

- Resource
- Query
- Catalog data set
- Observation

It is an aggregate of one or more ranges in individual coordinates and is required to have a reference to an AstroCoordSys. Multiple ranges along each coordinate axis are OR-ed together; the ranges of different axes are logically AND-ed together.

4.7.1 TimeInterval

A TimeInterval consists of a StartTime and/or a StopTime, both of which are instantiations of AstronTime. Note that they do not both have to be present, but at least one of them needs to be. TimeInterval has Boolean attributes LoLimInclude and HiLimInclude, and attribute FillFactor.

4.7.2 SpatialInterval

The spatial position area has more options:

- Simple intervals on a per-axis basis
- Circle or sphere
- Regions
 - Shapes: polygon, sector, circle, ellipse, convex, convex hull, sky index
 - Operations: intersection, union, negation

4.7.2.1 CInterval

CInterval is the vector equivalent of ScalarInterval (see: Section 4.4.1). LoLimit and HiLimit are of type SCValue (see Section 4.6.2.3).

4.7.2.2 Sphere

Since spherical regions are quite common (cone search!) these are made available as special cases. The class is an aggregate of a Center position (of type CValue; see: Section 4.6.3.3) and a Radius (consisting of a double and a unit). The spatial coordinates of the center should be in the AstroCoordSys associated with AstroCoordArea, of course. Note that the 2-D equivalent (circle) is available in the Region class (see: Section 4.8).

4.7.2.3 SpatialRegion

This is an object of class Region (see: Section 4.8).

4.7.3 VelocityInterval

The velocity interval is an instantiation of CInterval (see: Section 4.7.2.1).

4.7.4 SpectralInterval

The SpectralInterval is derived from the generic ScalarInterval class (see: Section 4.4.1), restricting Unit to appropriate values.

4.7.5 RedshiftInterval

The RedshiftInterval is derived from the generic ScalarInterval class (see: Section 4.4.1), restricting Unit to appropriate values.

4.8 Region

The Region specification is a rather elaborate construct that allows a very flexible definition of arbitrary 2-dimensional spatial shapes. It has 9 derived classes: 5 shapes and 3 operations. Note that the result of these operations performed on one or more regions is a region. All spatial coordinates should be in the AstroCoordSys associated with AstroCoordArea (or Region), of course. The Region object has an important attribute, `fill_factor` (between 0.0 and 1.0; default 1.0) that indicates what fraction of the region is really filled or covered. This is particularly useful when describing a resource's coverage.

4.8.1 Shapes

The first four shapes are 2-dimensional, while the remaining two are 3-dimensional, associated with the unit sphere. All boundaries are considered part of the inside of a shape.

4.8.1.1 Circle

A Circle (2-dimensional) shape (region) is an aggregate of a Center position (type `SCValue`; see: Section 4.6.2.3) and a Radius (consisting of a double and a unit).

4.8.1.2 Ellipse

The Ellipse (2-dimensional) is derived from the Circle by adding a minor radius and a position angle. The inheritance could, of course, have gone the other way (Circle being a special case of an Ellipse), but we felt that it is important to retain the simplest definition of a circle.

4.8.1.3 Polygon

A Polygon (2-dimensional) is an ordered list of one or more vertices. Its area is defined as that contained within the lines connecting neighboring (that is, in the ordered list) vertices. If the `CoordFlavor` is `CARTESIAN`, these lines are truly lines. If the `CoordFlavor` is `SPHERICAL`, the lines are, by default, great-circles, unless the Vertex object contains a `SmallCircle` object; in that case the line connecting that vertex with its predecessor is a small-circle (parallel). The curvature of the parallel is determined by the pole of the `SpatialFrame` in the current `AstroCoordSys`, unless a `PolePosition` is explicitly specified in the Vertex object. It is the responsibility of the server to ensure that the positions of the two sequential vertices actually lie on a parallel that is consistent with the implied or specified pole.

The boundaries are considered part of the region. The inside of the region is defined as that part of coordinate space that is encircled by the polygon in a counter-clockwise sense. Direction ambiguities in longitude are solved by allowing the longitude coordinate to become negative and to exceed 360°.

4.8.1.4 Sector

A Sector (2-dimensional) is defined by a position of type `SCValue` (see: Section 4.6.2.3) and two position angles. The inside of the sector is that part of the plane or sphere that is

swept by a half-line or great-circle rotating counter-clockwise from the first to the second position angle, centered on the specified position.

4.8.1.5 Convex

A Convex is the convex region that is defined by an aggregate of one or more Constraints. It is defined on the surface of the unit sphere and is hence in 2-dimensional spherical space. Each Constraint selects a circular region (or point) on the unit sphere. The Convex is therefore the intersection of a collection of circles and its sides are great-circles and/or small-circles. A Constraint is specified by a 3-dimensional vector (a point on the unit sphere) and an Offset along that vector. The Constraint's circular area is defined by the intersection of the unit sphere and the plane normal to the specified vector, intersecting it at the Offset distance from the origin.

4.8.1.6 ConvexHull

A ConvexHull is defined by an unordered list of one or more points. It is the smallest convex polygon that contains all points in the list. It may be constructed as the union of the triangles that can be formed from all possible triplets of points in the list.

4.8.1.7 SkyIndex

This is currently a place holder to allow integration of the specification of regions through sky indexing schemes.

4.8.2 Operations

There are three operations defined on or between Regions. The result of each of those operations is also a Region.

4.8.2.1 Intersection

The Intersection of two Regions is the Region that is common to both (logical AND). We might consider allowing Intersections of more than two Regions.

4.8.2.2 Union

The Union of two Regions is the Region that is contained in either or both of them (logical OR). We might consider allowing Unions of more than two Regions.

4.8.2.3 Negation

The negation of a Region is the Region that is not contained in the original (logical NOT). Note that this logically leads to a Region that does not include its boundaries.

5. Projection

The metadata described in this document specify a position or the volume occupied by a dataset. In order to be able to transform the coordinates for individual Data Model components (such as pixels) one needs a Projection object as well. Such a Projection metadata object is very much like the meat in the FITS WCS. It will be specified in a separate document. David Berry is working on this.

6. Conclusion and Usage Notes

VO needs a stringent specification in order to not get into trouble later on when we want to, for instance:

- use far field data for near field objects
- mix data in equatorial coordinates with data in Jovian coordinates
- combine pulsar observations from ground-based, spacecraft, and lunar observatories

6.1 Methods

The metadata objects described in Section 4 provide sufficient and necessary information, and ensure self-consistency. As commented before, this document focuses on the structure of the design, not on the methods. However, we can make some general observations. First, most of the client-level methods will probably appear as methods on the projection class. Second, robust design and implementation favors access to the various elements through the top-level objects. In other words, public methods should appear on the `AstroCoordSys`, `AstroCoords`, and `AstroCoordArea` classes, while most methods on the component classes ought to be restricted to the friend classes in the whole of the STC design structure.

Examples of methods that are obviously required:

- `AstroCoords`: get {spatial, temporal, spectral, redshift} coordinates of object
- `AstroCoords`: return new `AstroCoords` object which references a specified (presumably different) `AstroCoordSys` object (i.e., coordinate transformation)
- `AstroCoordArea`: same methods as for `AstroCoords`, above
- `AstroCoordArea`: return (measured) area of object.
- `AstroCoordArea`: is a given `AstroCoords` point inside or outside the region?
- `AstroCoordArea`: constructor methods that implement the three operations
- `AstroCoordArea`: provide largest inscribing and smallest circumscribing circles

6.2 Usage

Here are some notes on how the various components are intended to be used in the different use contexts.

6.2.1 Resource Profile

The STC Resource Profile, or Coverage, describes what part of STC space is potentially covered by the datasets available from the resource. This is a potential coverage since there is no guarantee that the entire volume is covered (in particular, if the resource contains a collection of pointed observations); note that this is where the `Region` fill-factor is a useful attribute. Examples of resources are depositories of pointed observations, survey atlases, and catalogs. The profile consists of an `AstroCoordSys` object that represents the resource's native coordinate system; an `AstroCoordArea` object that represents the actual coverage in the four standard STC frames; and an `AstroCoords` object that gives some specifics on the products in the resource. The purpose of the first two will be obvious, but the third may need some explanation. The `AstroCoords` object

do not intend to convey a particular position, but the typical properties of datasets in the resource. The Coordinate objects will not contain a CoordValue, but CoordError, CoordResolution, CoordSize, and CoordPixSize represent typical values for the datasets, where CoordSize refers to the field-of-view.

6.2.2 Query Constraint

A query constraint is, in a way, a mirror image of a resource profile, and its structure and the meaning of its components is very much the same: the AstroCoordArea defines the coordinate volume that the query requests to be filled, while the AstroCoords components (again, with missing CoordValue), as far as present, represent desirable values for errors, resolution, pixelsize, and field-of-view (if the whole area cannot be filled at once).

6.2.3 Catalog Entry

The STC object that accompanies the return of catalog information (typically a table of selected catalog entries) is set up slightly differently. The AstroCoordSys provides the usual coordinate system information, of course. The AstroCoordArea describes the coverage of the returned information; that will usually be the intersection of the query coverage with the resource (catalog) coverage. The AstroCoords object will generally refer to the catalog data that are being returned; i.e., its components are likely to refer to columns in the table that the STC object is associated with (such as galaxy positions, sizes, and redshifts) or have specific values if they happen to be constant for all entries (for instance, errors or resolution).

6.2.4 Observational Data

In this context two STC objects are required. One describes the location of the observatory (AstroCoordSys and AstroCoords; note that this may refer to an orbit ephemeris file for spacecraft). The other STC object describes the volume in coordinate space that is occupied by the observational data: AstroCoordSys, AstroCoordArea (the field-of-view), and AstroCoords (to convey errors, resolution, and pixel size). The information about the observatory's location is essential, in particular for near-field/far-field spatial transformations and time transformations. This is a typical case where metadata that are "obvious" (and often implied) in the context of a particular observatory's archive needs to be made explicit.

7. Implementation

7.1 XML Schema Implementation

The design of Section 4 has been implemented in three XML schemata, as stated in Section 1. The present document will gradually be expanded with more examples in the appendix to this document.

One important thing to note is that the schema implementation will allow all elements to contain actual values or to refer to another element; this will allow integration into VOTable where such references would be to table columns. Another notable implementation detail is that coordinate components are allowed to be provided in FITS

file. The schemas include a mechanism to associate particular elements with specific binary table columns. This is particularly of interest in conjunction with spacecraft orbit ephemerides.

7.2 Simplifications

Simplifications are allowed as long as the requirements are not compromised. They come in two categories.

7.2.1 Descriptions

Since it was decided to allow the use of XInclude in the schemas, we will create a library of common element instances with distinctive ID values. Consequently, it will not be necessary to create, for instance, commonly used coordinate systems in each XML document; instead, a simple XInclude to a library entry will perform the equivalent task.

7.2.2 Parsers

One option is to design schemas that contain subsets, in such a way that instantiations are upward compatible with the full schema. However, the simpler solution is to provide the canned elements in the XInclude library with unique and distinctive ID values, so that parsers do not need to process the entire coordinate system description but can simply rely on the ID.

Appendix A – UML Representation

The following pages contain the STC design in UML diagrams.

Space-Time Coordinate Metadata for the Virtual Observatory

Arnold Rots, CfA/SAO

The UML diagrams on the following pages specify the design of the VO Space-Time Coordinate metadata. Specifically, these metadata describe the volume in coordinate space that is occupied by the data object that they are attached to (or the query, or the resource), with additional information on actual or desired properties of the data along those coordinate axes.

The coordinates that are covered by these metadata are:

- Time
- Spatial position (1-D, 2-D, 3-D; spherical, Cartesian, or unit vector coordinates)
- Spatial velocity (1-D, 2-D, 3-D)
- Redshift (or Doppler velocity)
- Spectrum

The reason for lumping these 5 coordinates together is that their reference frames are intertwined.

The coordinate axis properties included are:

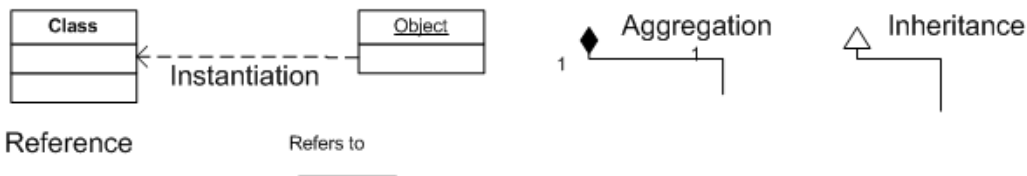
- Name
- Value
- Error
- Resolution
- Size
- Pixel size

Note that mapping metadata objects are still required to tie pixels to these axes (and vice versa).

The basic building block classes (aggregated into all-encompassing STC class) are:

- *AstroCoordSys*: defines the coordinate systems and reference frames for all coordinates in a fully general way, allowing inclusion of, e.g., terrestrial, solar system, and planetary frames. Note that *AstroCoordSys* is inherited from a generalized *CoordSys* which is a collection of one or more *CoordFrames*. The coordinate frames required in *AstroCoordSys* are all derived from *CoordFrame*.
- *AstroCoords*: defines a position in coordinate space, referenced to an *AstroCoordSys*, with error, resolution, pixel, etc., information. Inherited from a generalized *Coords*.
- *AstroCoordArea*: (derived from *CoordArea*) defines a volume in coordinate space, again tied to an *AstroCoordSys*; this includes the more sophisticated *Region* specification for 2-D spatial areas.

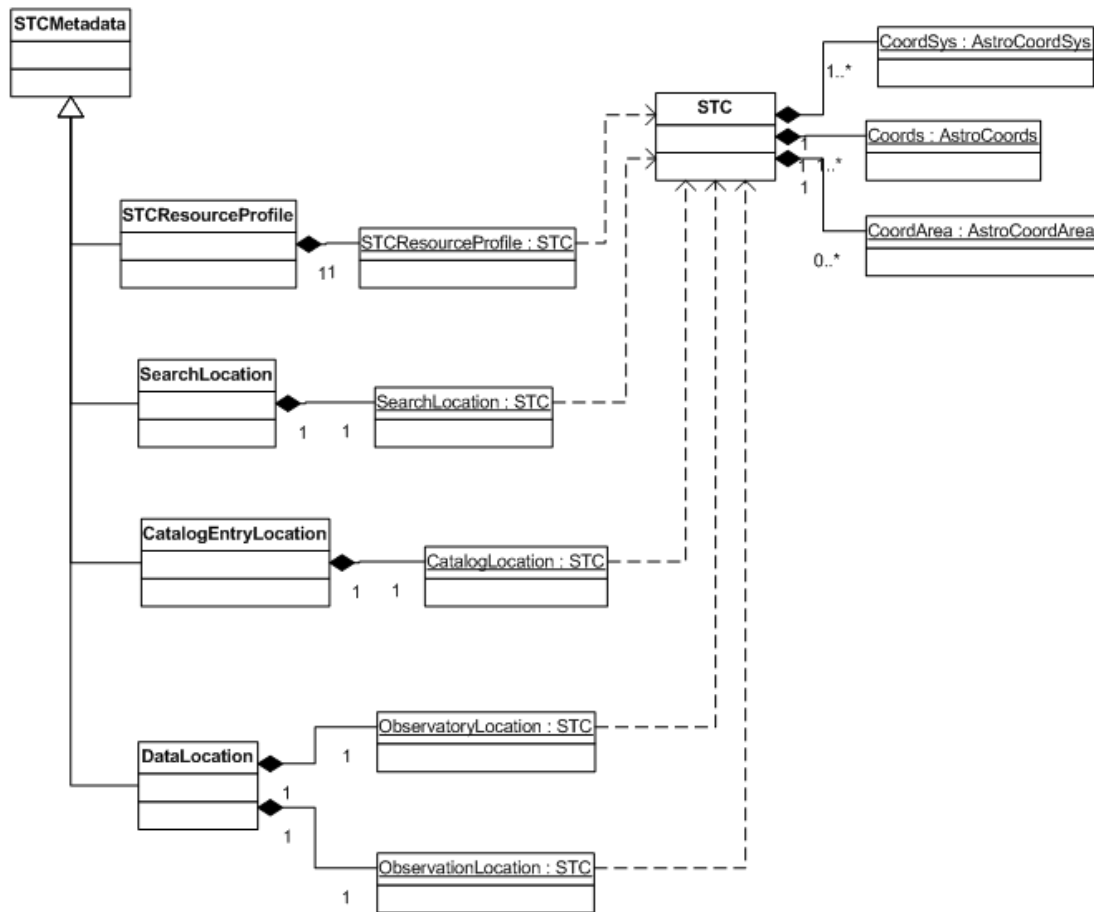
The following shapes are being used in these UML diagrams:



We grant that one might argue that some of the classes should be aggregates of objects, rather than aggregates of classes, but feel that this does not really affect the essence of this design which is, admittedly, somewhat fuzzy in this respect.

Note that for each of the Coordinate classes derived from the Generic Coordinate base class, the classes that are aggregated into the coordinate class (with the exception of *CoordName*) are constrained by the derived class. I.e., the *CoordValue*, *CoordError*, etc., for *TimeCoord*, *SpatialCoord*, *RedshiftCoord* should probably be classes that are multiply inherited from their generic classes as well as from the types that are shown in their UML diagrams. Similarly, one might wonder about the implied multiple inheritance of *CoordInterval* in the Coordinate Area diagram.

Space-Time Metadata Top-level Classes

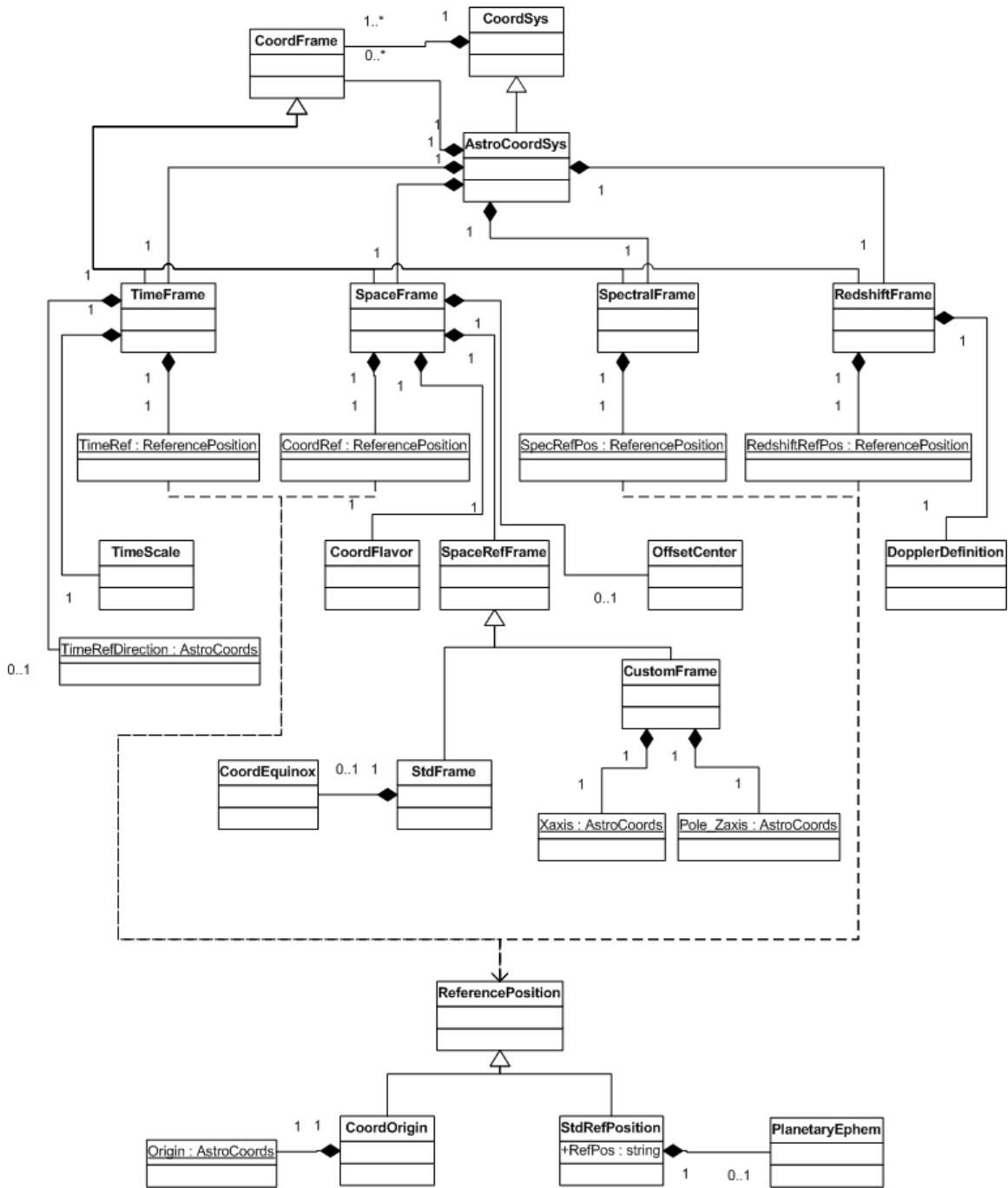


The *STCMetadata* class has four derived forms, each of which consists of one or two instantiations of *STC*:

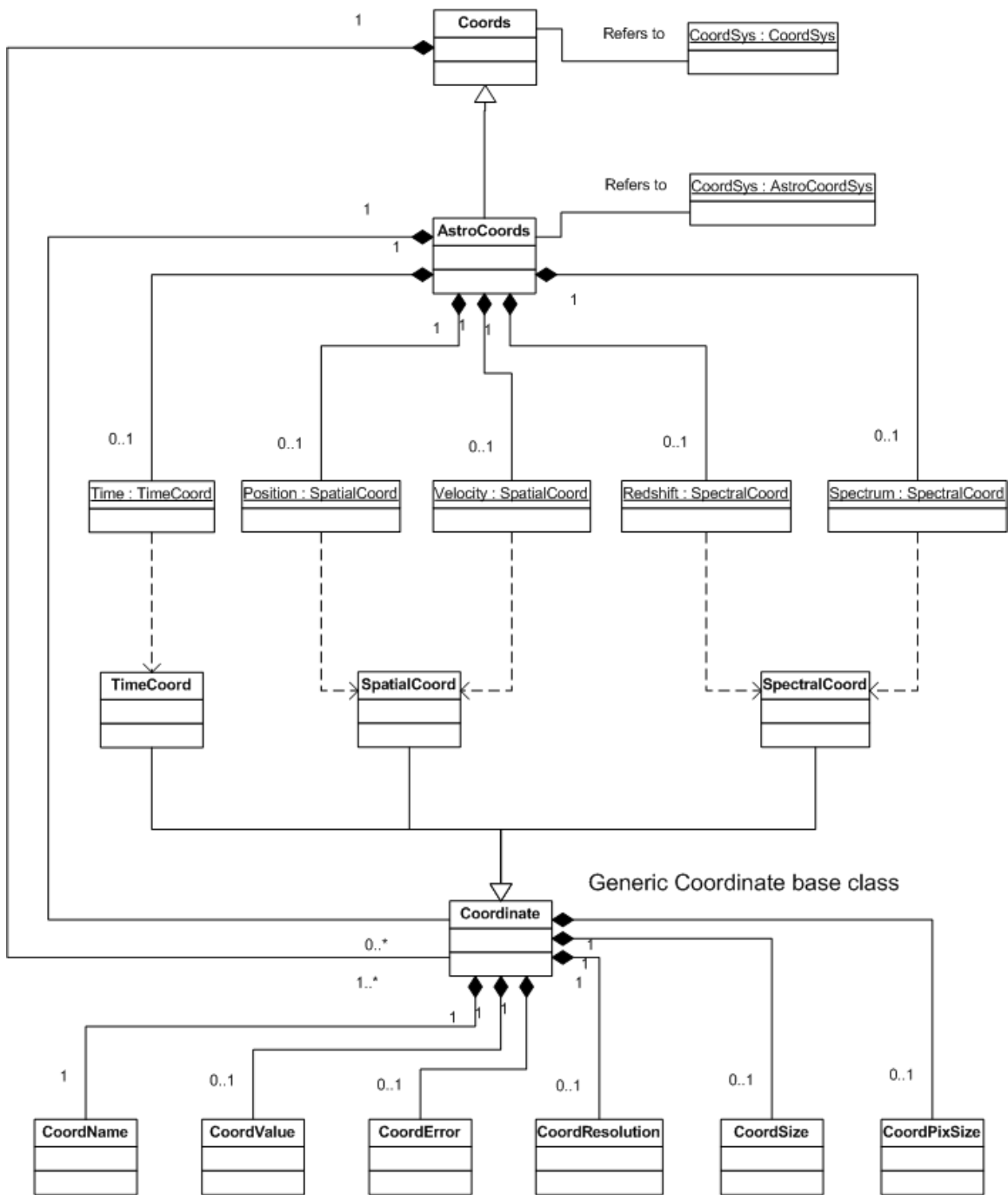
- *STCResourceprofile*: describes the coverage of a resource.
- *SearchLocation*: describes the coordinate volume requested in a query, as well as desired resolutions, errors, etc.
- *CatalogEntryLocation*: describes the coordinate coverage of a set of catalog entries, as well as the coordinates, errors, etc., when these are part of the entries.
- *DataLocation*: describes the coordinate volume occupied by a particular dataset (*ObservationLocation*), as well as the position of the observatory that produced the dataset (*ObservatoryLocation*); both are needed, in all coordinates, to facilitate transformations.

STC contains one or more instantiations of *CoordSys*, *Coords*, and (optionally) *CoordArea*.

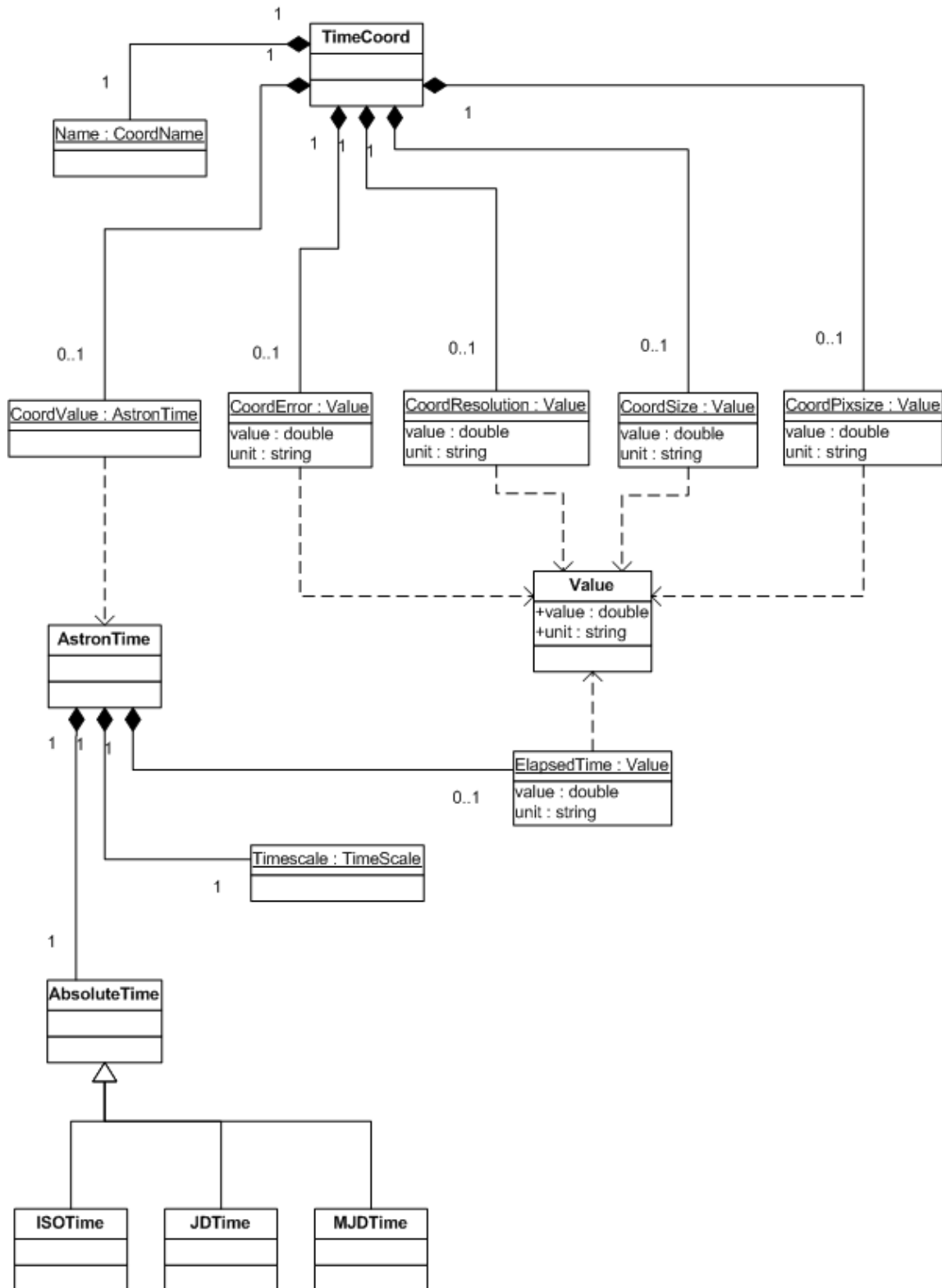
The Coordinate System



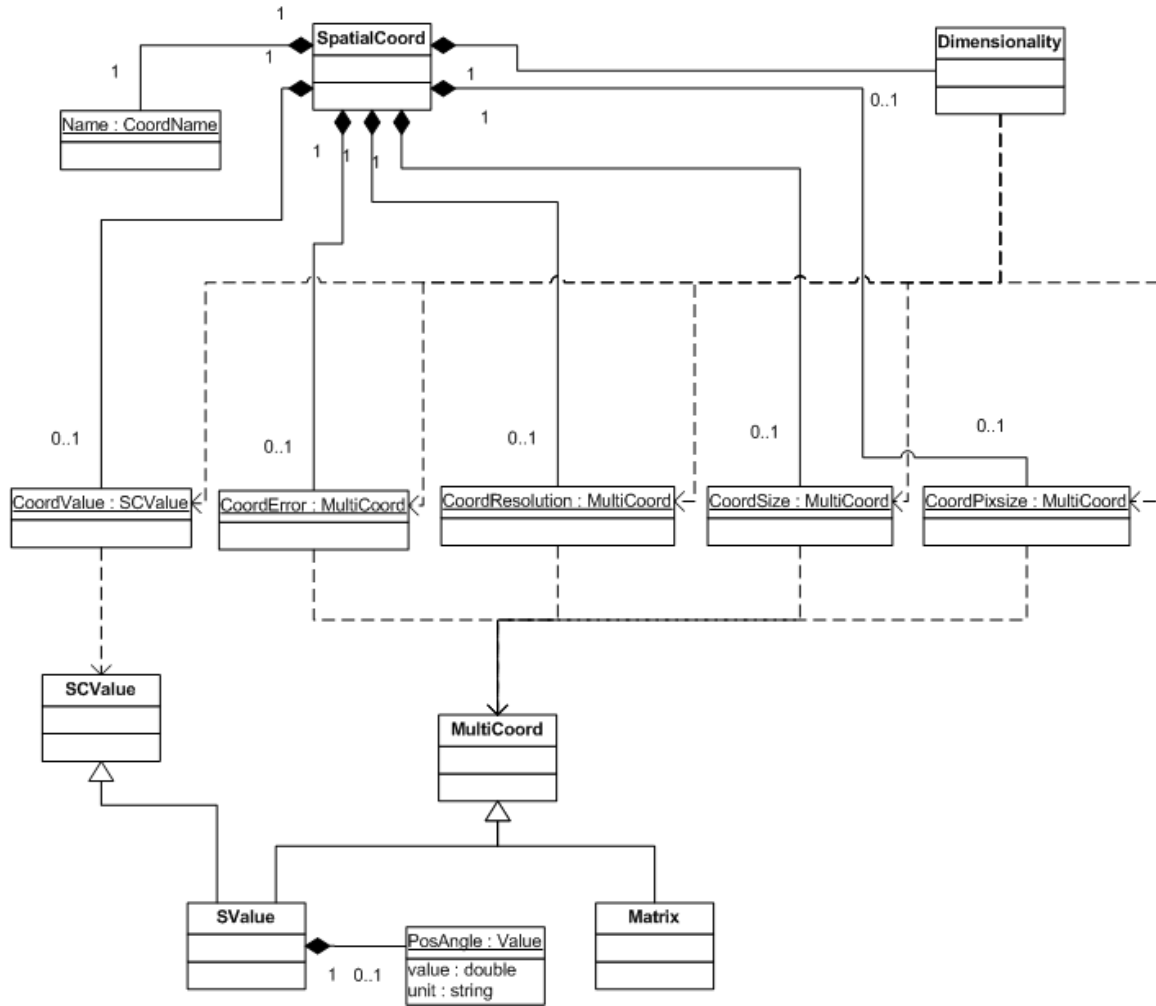
Coords Class - for full coordinate description
 AstroCoords - for astronomical coordinates



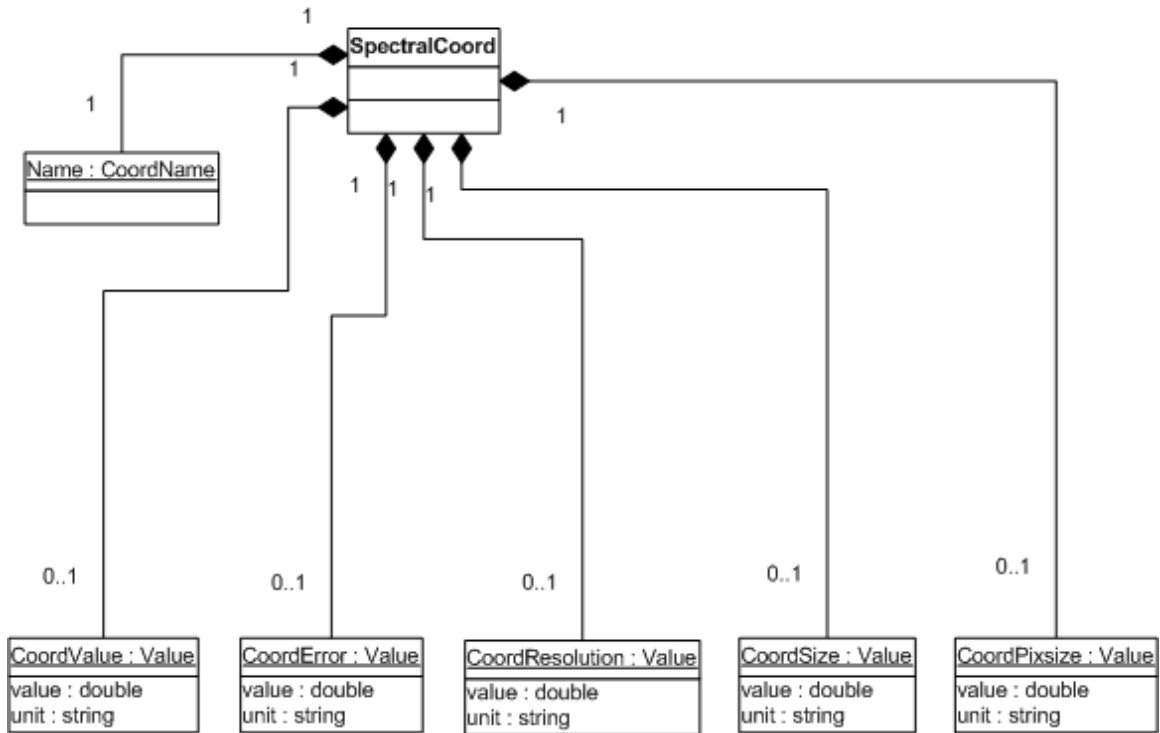
TimeCoord Class - for Time Coordinate



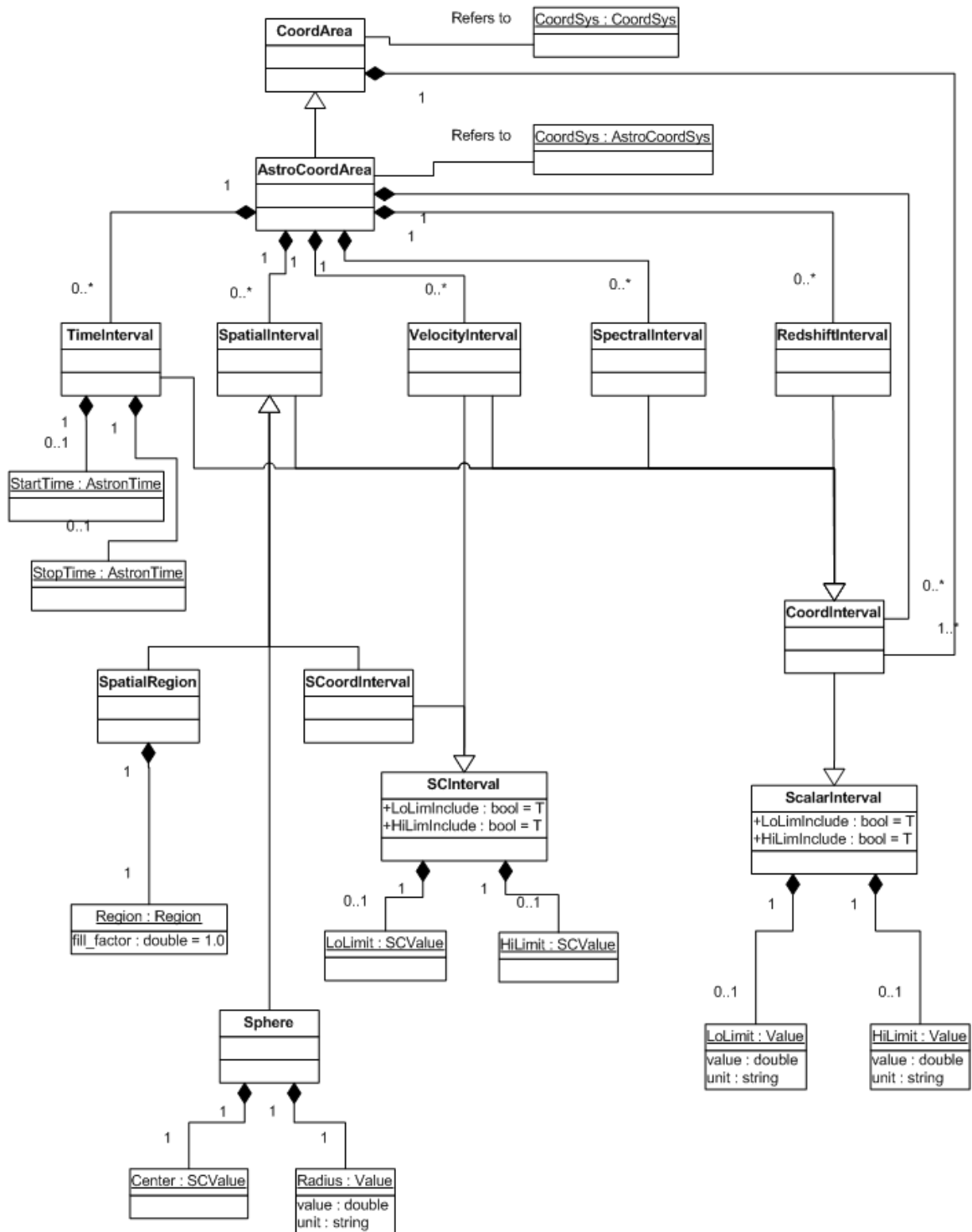
SpatialCoord Class - for Spatial Position and Velocity Coordinates



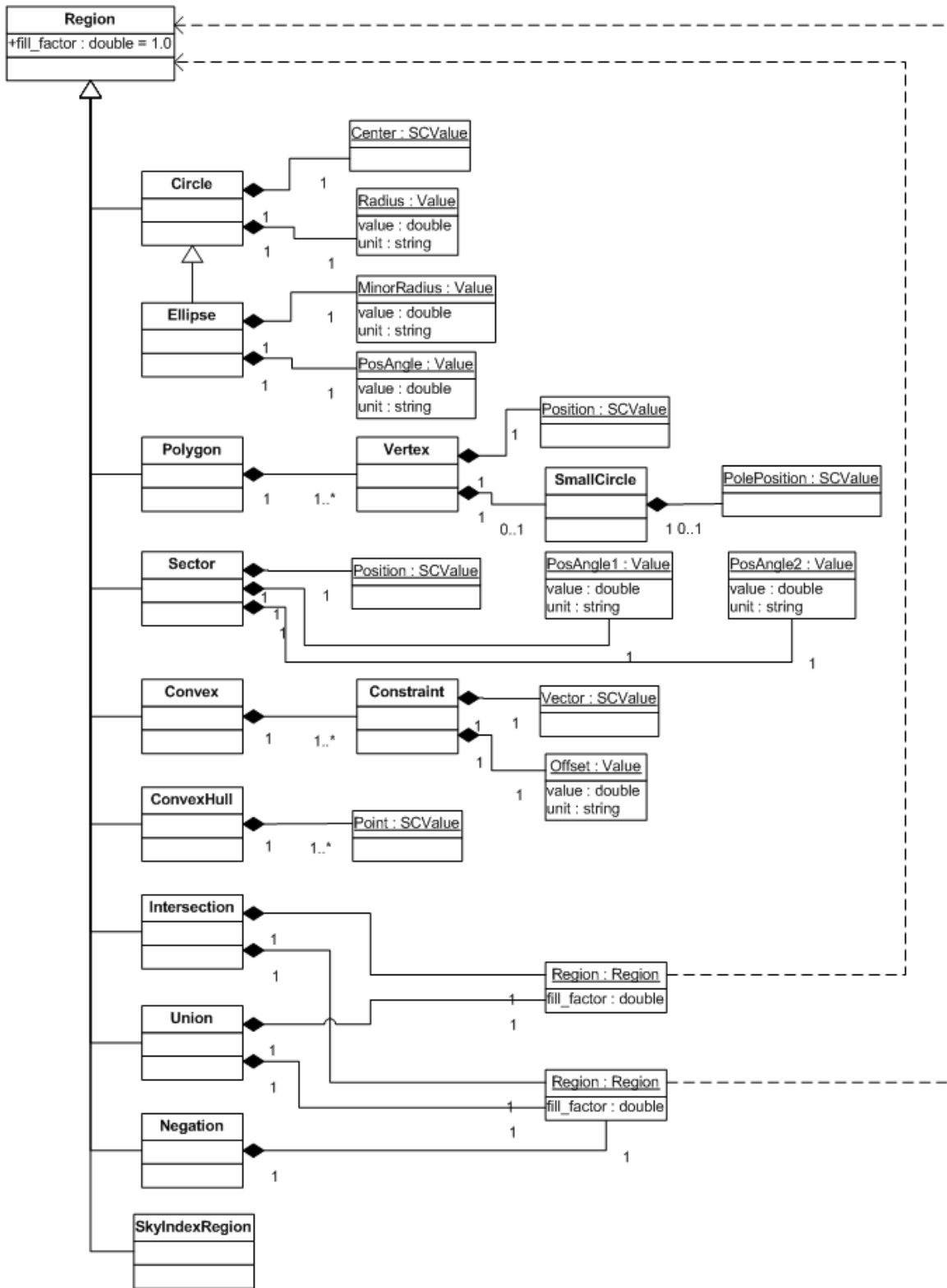
SpectralCoord Class - for Redshift and Spectral Coordinates



Coordinate Area - the coordinate volume that is taken up



Region (spatial)



Appendix B – Examples

B.1 STCResourceProfile

Resource profile for the Chandra X-ray Observatory Data Archive

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v2004 rel. 3 U (http://www.xmlspy.com) by
Arnold Rots (Smithsonian Observatory) -->
<!-- Sample XML file generated by XMLSPY v2004 rel. 3 U
(http://www.xmlspy.com)-->
<STCResourceProfile xmlns="urn:nvo-stc" xmlns:crd="urn:nvo-coords"
xmlns:reg="urn:nvo-region" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:nvo-stc
http://www.ivoa.net/xml/STC/v1.0">
  <AstroCoordSystem ID="ICRS-TT-CXO">
    <TimeFrame>
      <TimeScale>TT</TimeScale>
      <TOPOCENTER/>
    </TimeFrame>
    <SpaceFrame>
      <ICRS/>
      <TOPOCENTER/>
      <SPHERICAL coord_naxes="2" />
    </SpaceFrame>
    <SpectralFrame>
      <TOPOCENTER/>
    </SpectralFrame>
  </AstroCoordSystem>
  <AstroCoordArea ID="AllSky-CXO" coord_system_id="ICRS-TT-CXO">
    <TimeInterval>
      <StartTime>
        <crd:Timescale>TT</crd:Timescale>
        <crd:ISOTime>1999-07-23T16:00:00</crd:ISOTime>
      </StartTime>
    </TimeInterval>
    <PositionInterval unit="deg">
      <Coord2VecInterval fill_factor="0.02">
        <LoLimit2Vec>0.0 -90.0</LoLimit2Vec>
        <HiLimit2Vec>360.0 90.0</HiLimit2Vec>
      </Coord2VecInterval>
    </PositionInterval>
    <SpectralInterval unit="keV">
      <LoLimit>0.12</LoLimit>
      <HiLimit>10.0</HiLimit>
    </SpectralInterval>
  </AstroCoordArea>
  <CoordSpec coord_system_id="ICRS-TT-CXO">
    <crd:Time unit="s">
      <crd:Name>Time</crd:Name>
      <crd:Error>0.0001</crd:Error>
      <crd:Resolution>0.000016</crd:Resolution>
      <crd:Size>100000</crd:Size>
    </crd:Time>
  </CoordSpec>
</STCResourceProfile>
```

STC Metadata for the VO

```
</crd:Time>
<crd:Position2D unit="arcsec">
  <crd:Name>Position</crd:Name>
  <crd:Error2>1.0 1.0</crd:Error2>
  <crd:Resolution2>0.5 0.5</crd:Resolution2>
  <crd:Size2>1000 1000</crd:Size2>
</crd:Position2D>
<crd:Spectral unit="keV">
  <crd:Name>Energy</crd:Name>
  <crd:Error>0.1</crd:Error>
  <crd:Resolution>0.02</crd:Resolution>
  <crd:Size>10</crd:Size>
</crd:Spectral>
</CoordSpec>
</STCResourceProfile>
```

B.2 Galaxy Catalog

This example shows the STC specification for a fictitious catalog of galaxies that covers RA from 9 to 18 hours and Dec from 20 to 70 degrees, observed between JD 2440000 and 2441000, in a band between 5000 and 6500 Å, providing (RA,Dec; FK4:B1950.0) with errors and sizes, barycentric radial velocities, as well as Supergalactic coordinates and Galactocentric radial velocities, with errors; these quantities are supposed to be provided in columns 3-12 of an associated table.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v2004 rel. 3 U (http://www.xmlspy.com) by
Arnold Rots (Smithsonian Observatory) -->
<!--Sample XML file generated by XMLSPY v2004 rel. 3 U
(http://www.xmlspy.com)-->
<CatalogEntryLocation xmlns="urn:nvo-stc" xmlns:crd="urn:nvo-coords"
xmlns:reg="urn:nvo-region" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:nvo-stc
http://www.ivoa.net/xml/STC/v1.0">
  <AstroCoordSystem ID="B1950-OPTICAL-ET">
    <TimeFrame>
      <TimeScale>ET</TimeScale>
      <TOPOCENTER/>
    </TimeFrame>
    <SpaceFrame>
      <FK4>
        <Equinox>B1950.0</Equinox>
      </FK4>
      <BARYCENTER/>
      <SPHERICAL coord_naxes="2" coord_vel="true"/>
    </SpaceFrame>
    <SpectralFrame>
      <TOPOCENTER/>
    </SpectralFrame>
    <RedshiftFrame>
      <DopplerDefinition>OPTICAL</DopplerDefinition>
      <BARYCENTER/>
    </RedshiftFrame>
```

STC Metadata for the VO

```
</AstroCoordSystem>
<AstroCoordSystem ID="SGC-OPTICAL-ET">
  <TimeFrame>
    <Name>Time</Name>
    <TimeScale>ET</TimeScale>
    <TOPOCENTER/>
  </TimeFrame>
  <SpaceFrame>
    <Name>SGC</Name>
    <SUPER_GALACTIC/>
    <BARYCENTER/>
    <SPHERICAL coord_naxes="2" coord_vel="true"/>
  </SpaceFrame>
  <SpectralFrame>
    <Name>Optical</Name>
    <TOPOCENTER/>
  </SpectralFrame>
  <RedshiftFrame>
    <Name>DopplerVelocity</Name>
    <DopplerDefinition>OPTICAL</DopplerDefinition>
    <GALACTIC_CENTER/>
  </RedshiftFrame>
</AstroCoordSystem>
<Coords coord_system_id="B1950-OPTICAL-ET">
  <crd:Position2D unit="deg">
    <crd:Name>RA,Dec</crd:Name>
    <crd:Value2Ref>Column3</crd:Value2Ref>
    <crd:Error2Ref>Column4</crd:Error2Ref>
    <crd:Size2Ref>Column5</crd:Size2Ref>
  </crd:Position2D>
  <crd:Redshift unit="km" vel_time_unit="s">
    <crd:Name>Vrad(barycenter)</crd:Name>
    <crd:ValueRef>Column6</crd:ValueRef>
    <crd:ErrorRef>Column7</crd:ErrorRef>
  </crd:Redshift>
</Coords>
<Coords coord_system_id="SGC-OPTICAL-ET">
  <crd:Position2D unit="deg">
    <crd:Name>SGLong, SGLat</crd:Name>
    <crd:Value2Ref>Column8</crd:Value2Ref>
    <crd:Error2Ref>Column9</crd:Error2Ref>
    <crd:Size2Ref>Column10</crd:Size2Ref>
  </crd:Position2D>
  <crd:Redshift unit="km" vel_time_unit="s">
    <crd:Name>Vrad(Galcenter)</crd:Name>
    <crd:ValueRef>Column11</crd:ValueRef>
    <crd:ErrorRef>Column12</crd:ErrorRef>
  </crd:Redshift>
</Coords>
<AstroCoordArea coord_system_id="B1950-OPTICAL-ET"
  ID="RA6-18hDec20-70deg">
  <TimeInterval>
    <StartTime>
      <crd:Timescale>ET</crd:Timescale>
      <crd:JDTime>2440000</crd:JDTime>
    </StartTime>
    <StopTime>
```

STC Metadata for the VO

```
        <crd:Timescale>ET</crd:Timescale>
        <crd:JDTime>2441000</crd:JDTime>
    </StopTime>
</TimeInterval>
<Region>
    <reg:Polygon coord_system_id="B1950-OPTICAL-ET"
        unit="deg">
        <reg:Vertex>
            <reg:Position>270 20</reg:Position>
        </reg:Vertex>
        <reg:Vertex>
            <reg:Position>90 20</reg:Position>
            <reg:SmallCircle/>
        </reg:Vertex>
        <reg:Vertex>
            <reg:Position>90 70</reg:Position>
        </reg:Vertex>
        <reg:Vertex>
            <reg:Position>270 70</reg:Position>
        </reg:Vertex>
    </reg:Polygon>
</Region>
<SpectralInterval unit="A">
    <LoLimit>5000</LoLimit>
    <HiLimit>6500</HiLimit>
</SpectralInterval>
<RedshiftInterval unit="km" vel_time_unit="s">
    <HiLimit>10000</HiLimit>
</RedshiftInterval>
</AstroCoordArea>
</CatalogEntryLocation>
```

B.3 ROSAT Image

The following document describes the STC metadata for a ROSAT image. Note that the observatory position is contained in an orbit ephemeris file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v2004 rel. 3 U (http://www.xmlspy.com) by
Arnold Rots (Smithsonian Observatory) -->
<!-- Sample XML file generated by XMLSPY v2004 rel. 3 U
(http://www.xmlspy.com)-->
<ObsDataLocation xmlns="urn:nvo-stc" xmlns:crd="urn:nvo-coords"
xmlns:reg="urn:nvo-region" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:nvo-stc
http://www.ivoa.net/xml/STC/v1.0">
    <ObservatoryLocation ID="ROSAT">
        <AstroCoordSystem ID="FK5-UTC-VEL">
            <TimeFrame>
                <TimeScale>UTC</TimeScale>
                <TOPOCENTER/>
            </TimeFrame>
            <SpaceFrame>
                <FK5>
```

STC Metadata for the VO

```
        <Equinox>J2000.0</Equinox>
    </FK5>
    <TOPOCENTER/>
    <CARTESIAN coord_naxes="3" coord_vel="true"/>
</SpaceFrame>
</AstroCoordSystem>
<Coords coord_system_id="FK5-UTC-VEL">
  <crd:CoordFile>
    <crd:FITSFile hdu_num="1">
      http://MySpace.edu/OrbitEphemeris.fits
    </crd:FITSFile>
    <crd:FITSTime>
      <crd:Name>Time</crd:Name>
      <crd:Value>TIME</crd:Value>
    </crd:FITSTime>
    <crd:FITSPosition>
      <crd:Name>Position</crd:Name>
      <crd:Value>X,Y,Z</crd:Value>
    </crd:FITSPosition>
    <crd:FITSVelocity>
      <crd:Name>Velocity</crd:Name>
      <crd:Value>VX,VY,VZ</crd:Value>
    </crd:FITSVelocity>
  </crd:CoordFile>
</Coords>
</ObservatoryLocation>
<ObservationLocation ID="US701411P.N1">
  <AstroCoordSystem ID="FK5-UTC-Energy">
    <TimeFrame>
      <TimeScale>UTC</TimeScale>
      <TOPOCENTER/>
    </TimeFrame>
    <SpaceFrame>
      <FK5>
        <Equinox>J2000.0</Equinox>
      </FK5>
      <TOPOCENTER/>
      <SPHERICAL coord_naxes="2"/>
    </SpaceFrame>
    <SpectralFrame>
      <TOPOCENTER/>
    </SpectralFrame>
  </AstroCoordSystem>
  <Coords coord_system_id="FK5-UTC-Energy">
    <crd:Time unit="s">
      <crd:Name>Time</crd:Name>
      <crd:TimeInstant>
        <crd:Timescale>UTC</crd:Timescale>
        <crd:MJDTime>49192.57</crd:MJDTime>
      </crd:TimeInstant>
      <crd:Error>0.1</crd:Error>
      <crd:Resolution>22955</crd:Resolution>
      <crd:PixSize>22955</crd:PixSize>
    </crd:Time>
    <crd:Position2D unit="deg">
      <crd:Name>RA,Dec</crd:Name>
      <crd:Value2>233.73 23.49</crd:Value2>
```

STC Metadata for the VO

```
<crd:Error2>0.005 0.005</crd:Error2>
<crd:Resolution2>0.01 0.01</crd:Resolution2>
<crd:PixSize2>0.0041667 0.0041667
</crd:PixSize2>
</crd:Position2D>
<crd:Spectral unit="keV">
  <crd:Name>Energy</crd:Name>
  <crd:Value>1.0</crd:Value>
  <crd:Error>0.1</crd:Error>
  <crd:Resolution>2.3</crd:Resolution>
  <crd:PixSize>2.3</crd:PixSize>
</crd:Spectral>
</Coords>
<AstroCoordArea ID="ROSATFIELD"
  coord_system_id="FK5-UTC-Energy">
  <TimeInterval fill_factor="0.025">
    <StartTime>
      <crd:Timescale>UTC</crd:Timescale>
      <crd:ISOTime>1993-07-24T13:47:39
      </crd:ISOTime>
    </StartTime>
    <StopTime>
      <crd:Timescale>UTC</crd:Timescale>
      <crd:ISOTime>1993-08-03T21:17:42
      </crd:ISOTime>
    </StopTime>
  </TimeInterval>
  <Region>
    <reg:Circle coord_system_id="FK5-UTC-Energy"
      unit="deg">
      <reg:Center>233.73 23.49</reg:Center>
      <reg:Radius>1.0</reg:Radius>
    </reg:Circle>
  </Region>
  <SpectralInterval unit="keV">
    <LoLimit>0.1</LoLimit>
    <HiLimit>2.4</HiLimit>
  </SpectralInterval>
</AstroCoordArea>
</ObservationLocation>
</ObsDataLocation>
```

B.4 Blue M81 Image from KPNO

The following example describes a blue 1000 s exposure of M81 taken at Kitt Peak. IT is half a degree in size, has 0.3" pixels, a resolution of 0.8", and an absolute astrometric uncertainty of 1". The metadata values specific to KPNO and to this image are printed in **bold**.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v2004 rel. 3 U (http://www.xmlspy.com) by
Arnold Rots (Smithsonian Observatory) -->
<!--Sample XML file generated by XMLSPY v2004 rel. 3 U
```

STC Metadata for the VO

```
(http://www.xmlspy.com)-->
<ObsDataLocation xmlns="urn:nvo-stc" xmlns:crd="urn:nvo-coords"
xmlns:reg="urn:nvo-region" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:nvo-stc
http://www.ivoa.net/xml/STC/v1.0">
  <ObservatoryLocation ID="KPNO">
    <AstroCoordSystem ID="ICRS-TT-TOPO">
      <TimeFrame>
        <TimeScale>TT</TimeScale>
        <TOPOCENTER/>
      </TimeFrame>
      <SpaceFrame>
        <GEO/>
        <TOPOCENTER/>
        <GEOGRAPHIC coord_naxes="3"/>
      </SpaceFrame>
    </AstroCoordSystem>
    <Coords coord_system_id="KPNO">
      <crd:Position3D unit="deg deg m">
        <crd:Name>LongLatAlt</crd:Name>
        <crd:Value3>111.5944 31.9586 2158</crd:Value3>
      </crd:Position3D>
    </Coords>
  </ObservatoryLocation>
  <ObservationLocation ID="M81">
    <AstroCoordSystem ID="ICRS-TT-WAVELENGTH-TOPO">
      <TimeFrame>
        <TimeScale>TT</TimeScale>
        <TOPOCENTER/>
      </TimeFrame>
      <SpaceFrame>
        <ICRS/>
        <TOPOCENTER/>
        <SPHERICAL coord_naxes="2"/>
      </SpaceFrame>
      <SpectralFrame>
        <TOPOCENTER/>
      </SpectralFrame>
    </AstroCoordSystem>
    <Coords coord_system_id="ICRS-TT-WAVELENGTH-TOPO">
      <crd:Time unit="s">
        <crd:Name>Time</crd:Name>
        <crd:TimeInstant>
          <crd:Timescale>TT</crd:Timescale>
          <crd:ISOTime>2004-07-15T08:23:56
          </crd:ISOTime>
        </crd:TimeInstant>
        <crd:PixSize>1000</crd:PixSize>
      </crd:Time>
      <crd:Position2D unit="deg">
        <crd:Name>RA,Dec</crd:Name>
        <crd:Value2>148.88821 69.06529</crd:Value2>
        <crd:Error2>0.0003 0.0003</crd:Error2>
        <crd:Resolution2>0.00025 0.00025
        </crd:Resolution2>
        <crd:PixSize2>0.0001 0.0001</crd:PixSize2>
      </crd:Position2D>
    </Coords>
  </ObservationLocation>
</ObsDataLocation>
```

STC Metadata for the VO

```
<crd:Spectral unit="A">
  <crd:Name>Lambda</crd:Name>
  <crd:Value>4600</crd:Value>
  <crd:Resolution>400</crd:Resolution>
  <crd:PixSize>400</crd:PixSize>
</crd:Spectral>
</Coords>
<AstroCoordArea ID="M81Image"
  coord_system_id="ICRS-TT-WAVELENGTH-TOPO">
  <TimeInterval fill_factor="1.0">
    <StartTime>
      <crd:Timescale>TT</crd:Timescale>
      <crd:ISOTime>2004-07-15T08:17:36
    </crd:ISOTime>
    </StartTime>
    <StopTime>
      <crd:Timescale>TT</crd:Timescale>
      <crd:ISOTime>2004-07-15T08:30:16
    </crd:ISOTime>
    </StopTime>
  </TimeInterval>
  <PositionInterval unit="deg">
    <Coord2VecInterval>
      <LoLimit2Vec>148.18821 68.81529
    </LoLimit2Vec>
      <HiLimit2Vec>149.58821 69.31529
    </HiLimit2Vec>
    </Coord2VecInterval>
  </PositionInterval>
  <SpectralInterval unit="A">
    <LoLimit>4400</LoLimit>
    <HiLimit>4800</HiLimit>
  </SpectralInterval>
</AstroCoordArea>
</ObservationLocation>
</ObsDataLocation>
```

B.5 Query Specification

This example provides the STC specifications for a query that is looking for 40 arcmin images of the field 2 degrees around M81, taken since 1900, with a resolution of 1" and a pixel size of 0.5", broadband between 4000 and 7000 Å.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v2004 rel. 3 U (http://www.xmlspy.com) by
Arnold Rots (Smithsonian Observatory) -->
<!--Sample XML file generated by XMLSPY v2004 rel. 3 U
(http://www.xmlspy.com)-->
<SearchLocation xmlns="urn:nvo-stc" xmlns:crd="urn:nvo-coords"
xmlns:reg="urn:nvo-region" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:nvo-stc
http://www.ivoa.net/xml/STC/v1.0">
  <AstroCoordSystem ID="ICRS-TT-BARY">
    <TimeFrame>
```

STC Metadata for the VO

```
        <TimeScale>TT</TimeScale>
        <BARYCENTER/>
    </TimeFrame>
    <SpaceFrame>
        <ICRS/>
        <BARYCENTER/>
        <SPHERICAL coord_naxes="2"/>
    </SpaceFrame>
    <SpectralFrame>
        <BARYCENTER/>
    </SpectralFrame>
</AstroCoordSystem>
<AstroCoordArea ID="M81" coord_system_id="ICRS-TT-BARY">
    <TimeInterval>
        <StartTime>
            <crd:Timescale>TT</crd:Timescale>
            <crd:ISOTime>1900-01-01T00:00:00</crd:ISOTime>
        </StartTime>
    </TimeInterval>
    <Region>
        <reg:Circle coord_system_id="ICRS-TT-BARY"
            unit="deg">
            <reg:Center>148.9 69.1</reg:Center>
            <reg:Radius>2</reg:Radius>
        </reg:Circle>
    </Region>
    <SpectralInterval unit="A">
        <LoLimit>4000</LoLimit>
        <HiLimit>7000</HiLimit>
    </SpectralInterval>
</AstroCoordArea>
<CoordSpec coord_system_id="ICRS-TT-BARY">
    <crd:Position2D unit="deg">
        <crd:Name>RA,Dec</crd:Name>
        <crd:Resolution2>0.0003 0.0003</crd:Resolution2>
        <crd:Size2>0.67 0.67</crd:Size2>
        <crd:PixSize2>0.00015 0.00015</crd:PixSize2>
    </crd:Position2D>
    <crd:Spectral unit="A">
        <crd:Name>Lambda</crd:Name>
        <crd:Resolution>500</crd:Resolution>
    </crd:Spectral>
</CoordSpec>
</SearchLocation>
```

Appendix C - XInclude Library of Coordinate Systems

[TBD]

Appendix D - Changes from Previous Versions

From V0.5 to V0.51:

- Added STCResourceProfile example in Appendix B.

From V0.51 to V0.6:

- Finished schemas resulting in some tweaks to the design.

From V0.6 to V1.0:

- Added solar and planetary reference frames.
- Enumerated spatial reference positions, reference frames, and time scales in tables.
- Added mechanism to reference coordinate values in FITS files.
- Added examples
- Release as Working Draft