Space-Time Coordinate Metadata for the VO

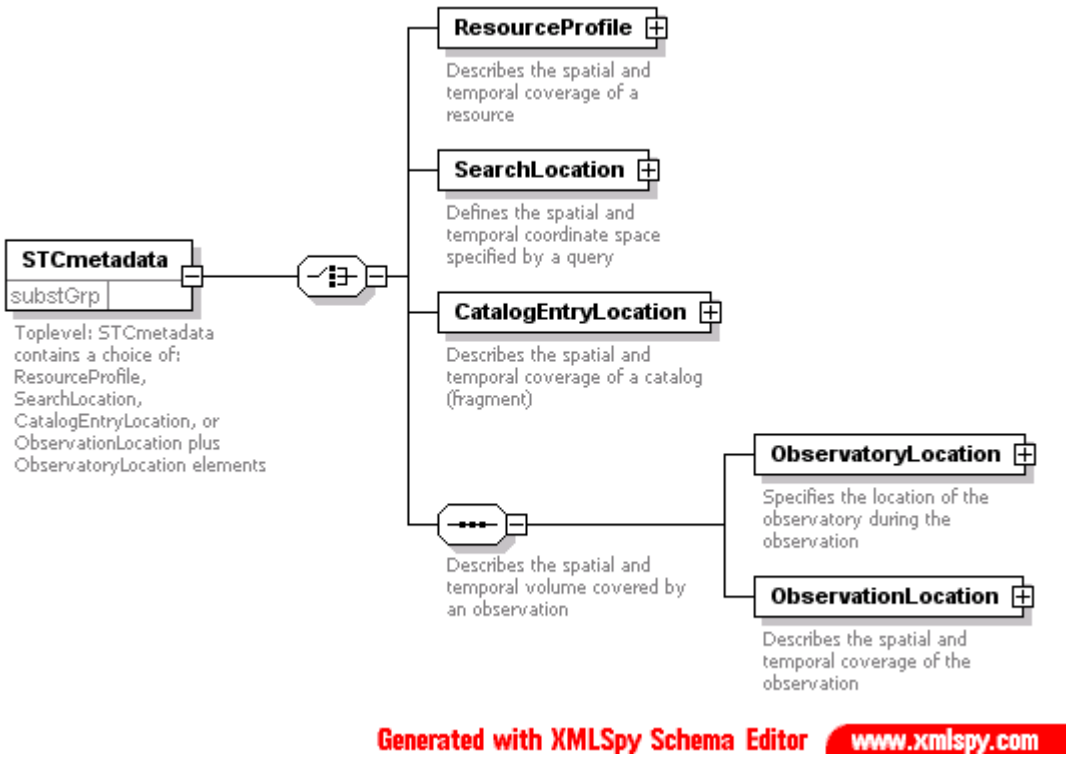
Arnold Rots
SAO

Space-Time Coordinate Metadata

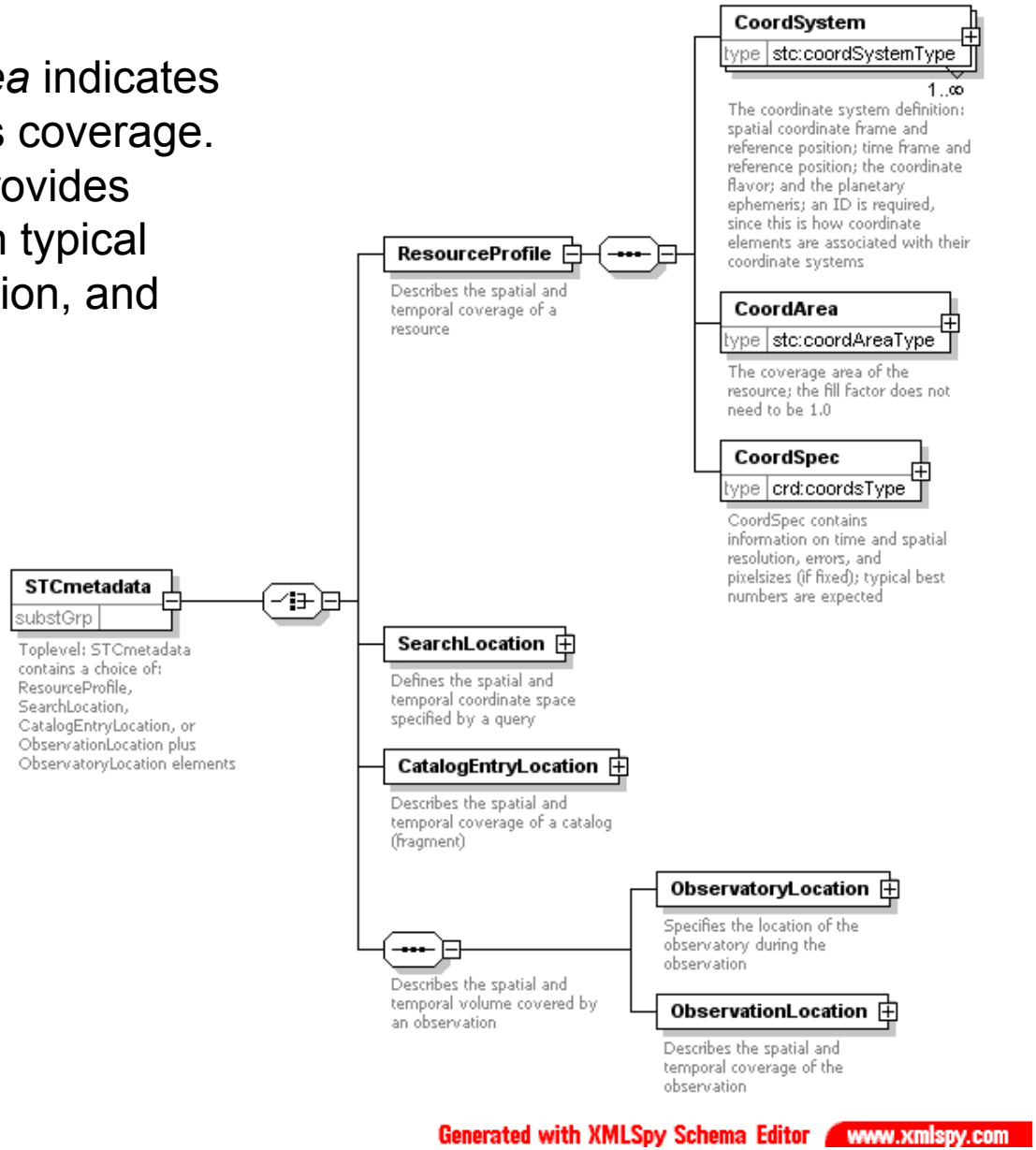
- The ***STC*** metadata describes the volume in coordinate space occupied by the data, resource, or query they are attached to
- The (intertwined) coordinates are:
  - Time
  - Space
  - Redshift / Doppler velocity
  - Spectrum
- What it does not describe is how this volume is projected/mapped onto the dataset (e.g., pixel array)

STC Structure

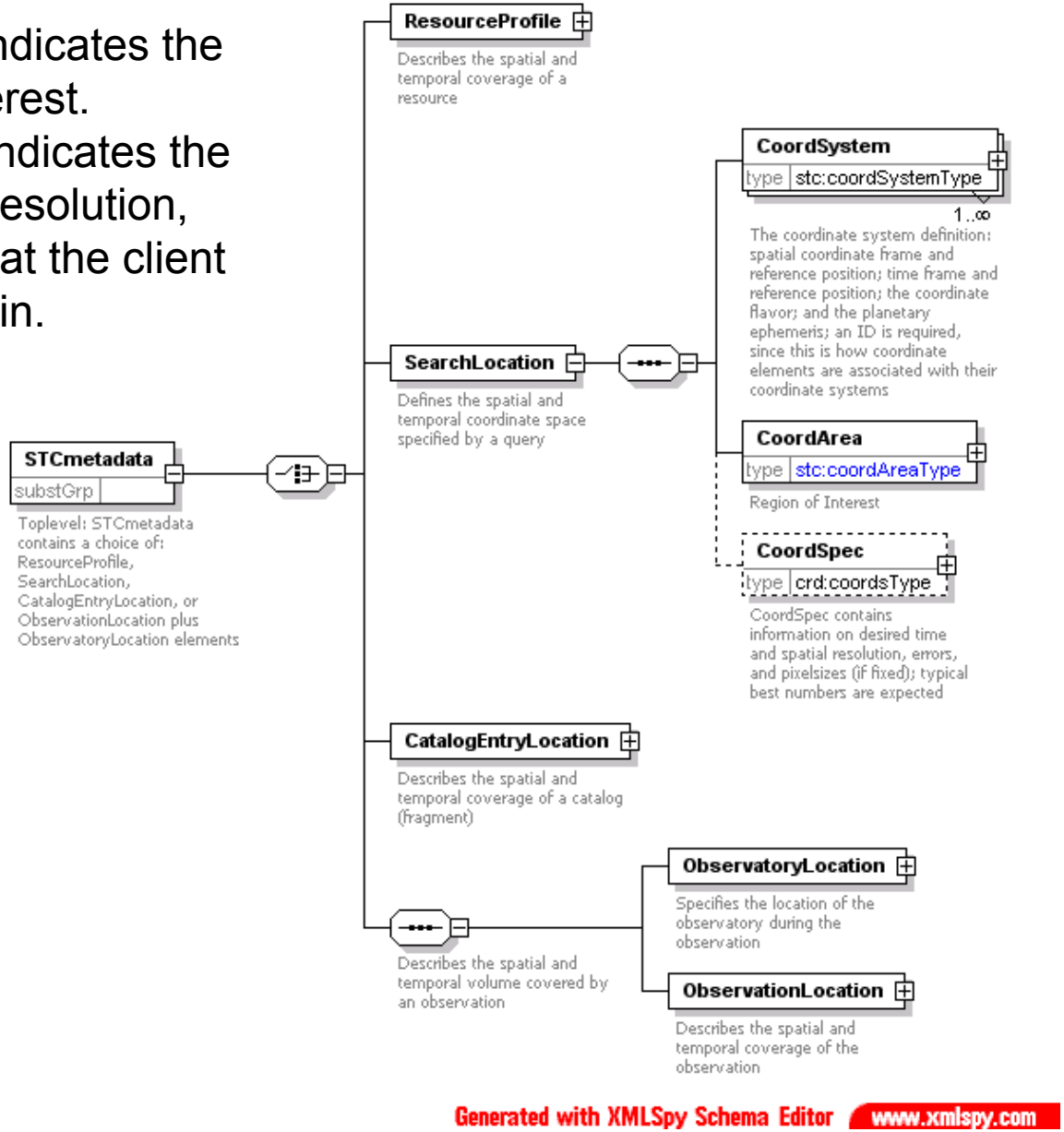
- ***STC*** contains three top-level element types:
  - *CoordSys* describes the coordinate frames
  - *Coords* for specifying a particular position
  - *CoordArea* specifies a volume
- The ***STC*** metadata element takes on one of four forms, depending on context:
  - Resource description
  - Query specification
  - Catalog dataset
  - Observation dataset
- The (spatial) ***region*** specification is used in *CoordArea*
- The schema definition is provided in three files, found at **http://hea-www.harvard.edu/~arots/nvometa**:
  - **stc.xsd**
  - **coords.xsd**
  - **region.xsd**


STCmetadata
substGrp
Toplevel: STCmetadata contains a choice of: ResourceProfile, SearchLocation, CatalogEntryLocation, or ObservationLocation plus ObservatoryLocation elements
ResourceProfile
Describes the spatial and temporal coverage of a resource
SearchLocation
Defines the spatial and temporal coordinate space specified by a query
CatalogEntryLocation
Describes the spatial and temporal coverage of a catalog (fragment)
Describes the spatial and temporal volume covered by an observation
ObservatoryLocation
Specifies the location of the observatory during the observation
ObservationLocation
Describes the spatial and temporal coverage of the observation
Generated with XMLSpy Schema Editor
www.xmlspy.com

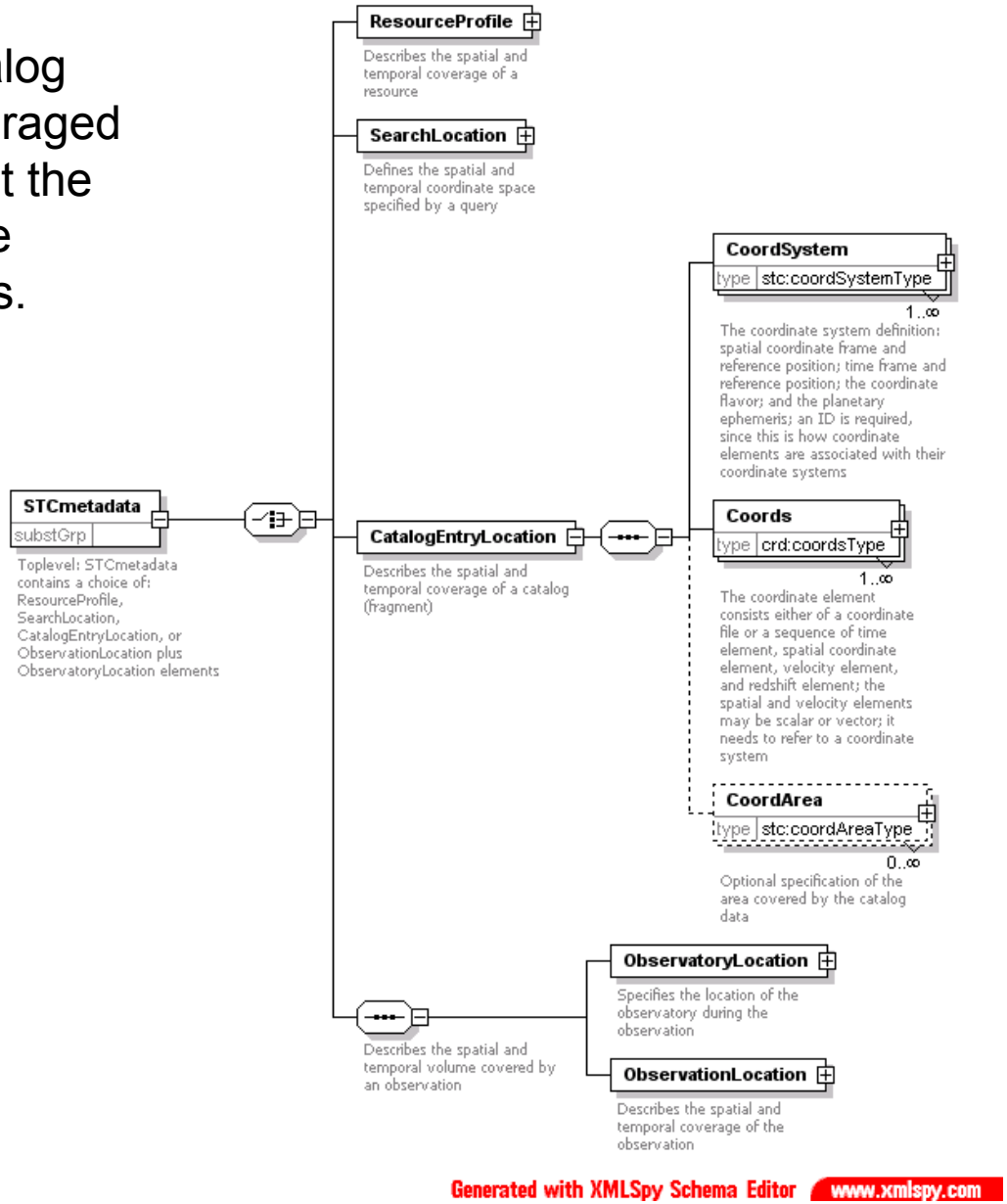
The *CoordArea* indicates the resource's coverage. *CoordSpec* provides information on typical errors, resolution, and fields of view.

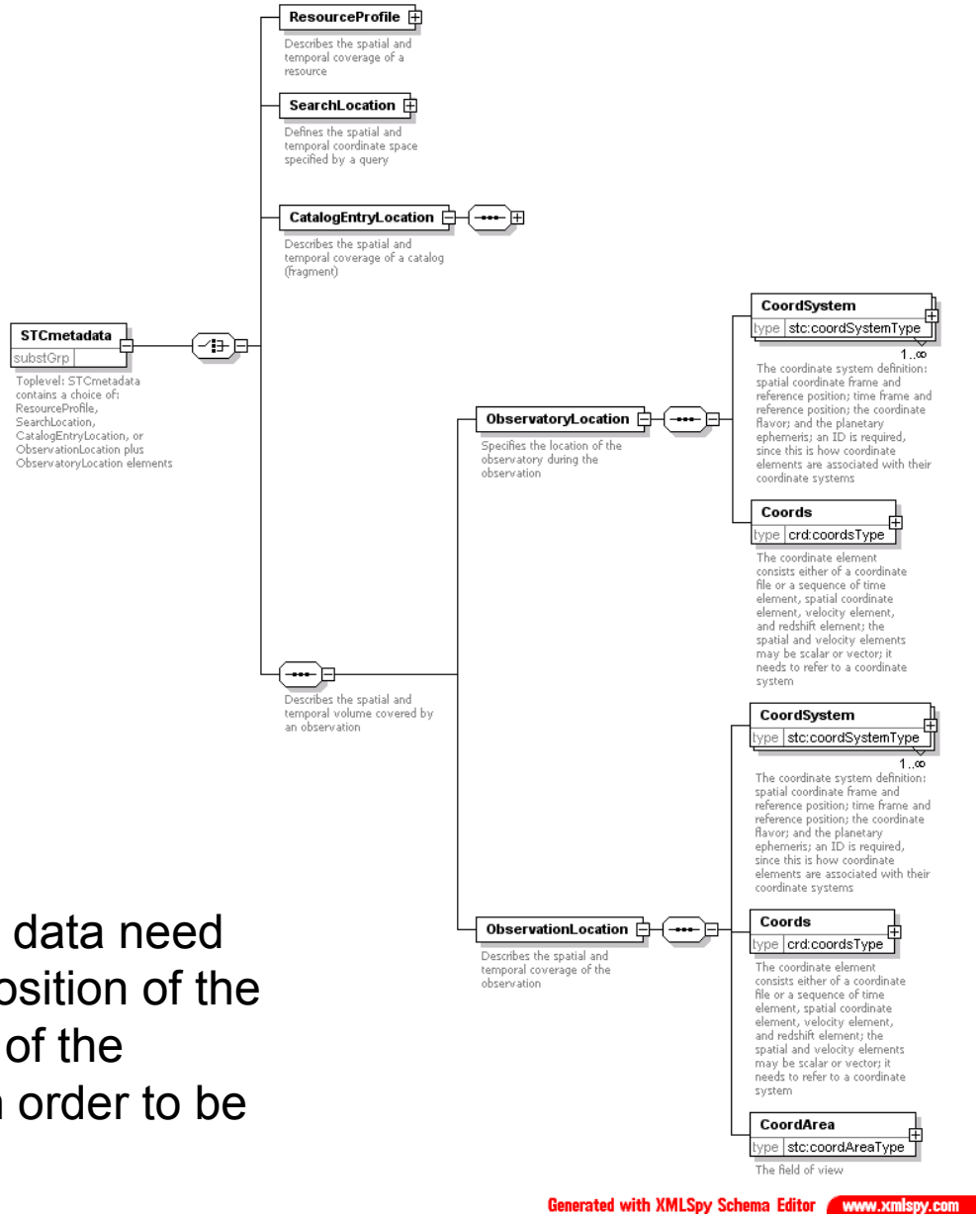


CoordArea indicates the region of interest. *CoordSpec* indicates the image size, resolution, and errors that the client is interested in.



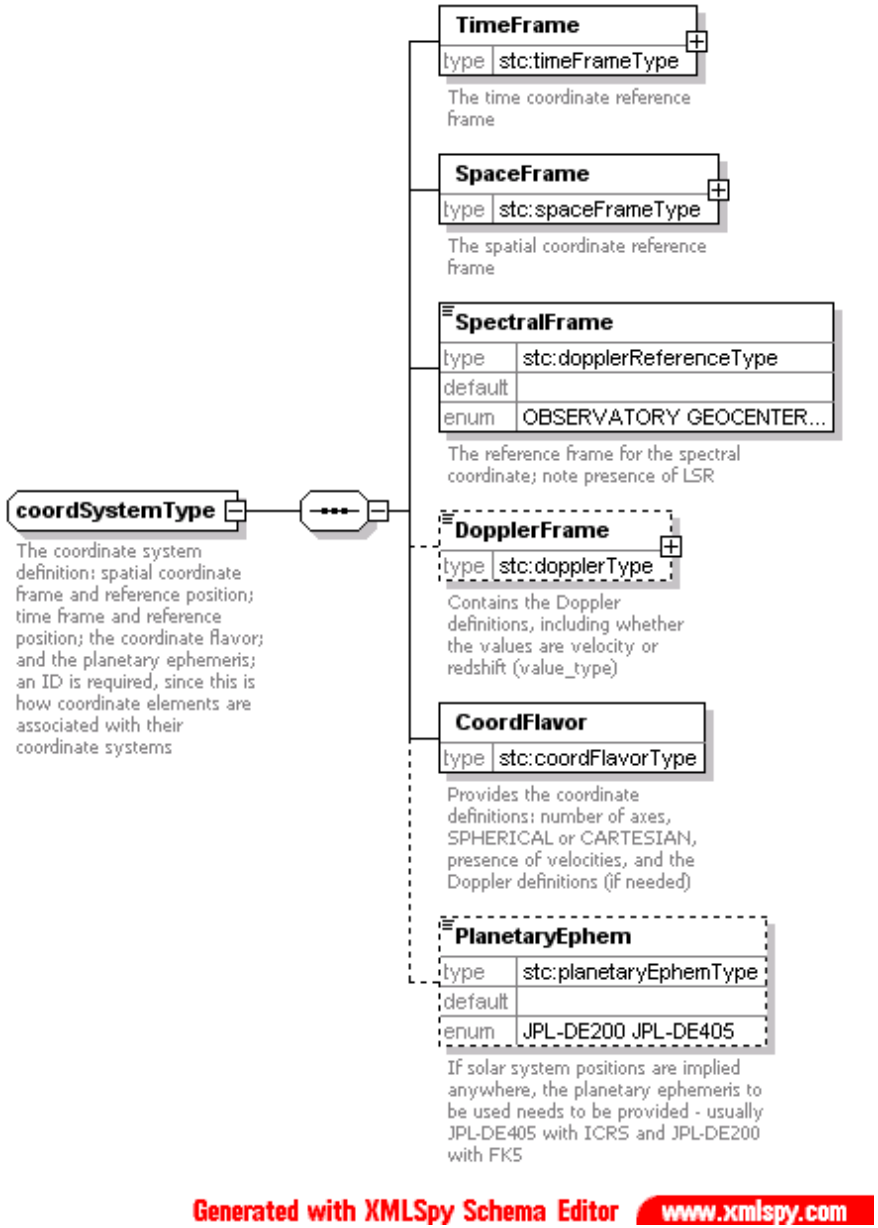
Servers of catalog data are encouraged to indicate what the coverage of the returned data is.

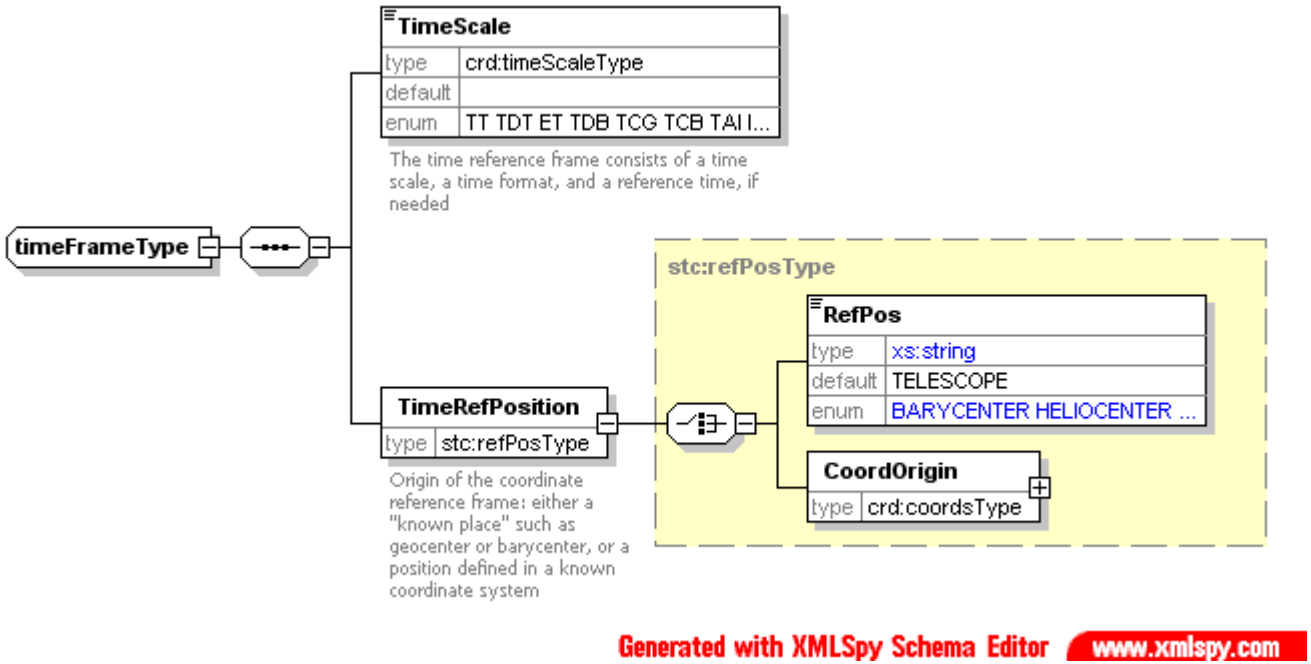




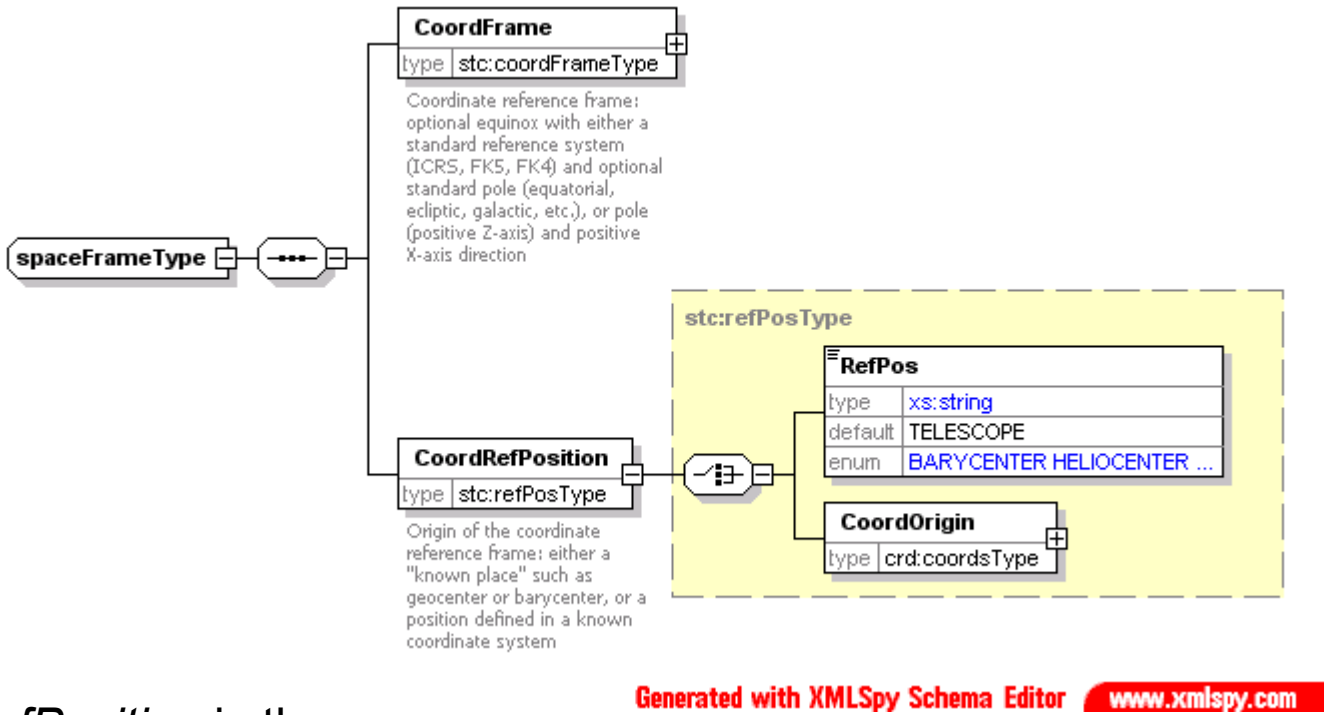
Observational data need not only the position of the data, but also of the observatory in order to be complete.

The *CoordSys* element contains the coordinate frames for each of the coordinates, a description of what is the coordinates axes are, and a reference to a planetary ephemeris (if needed).

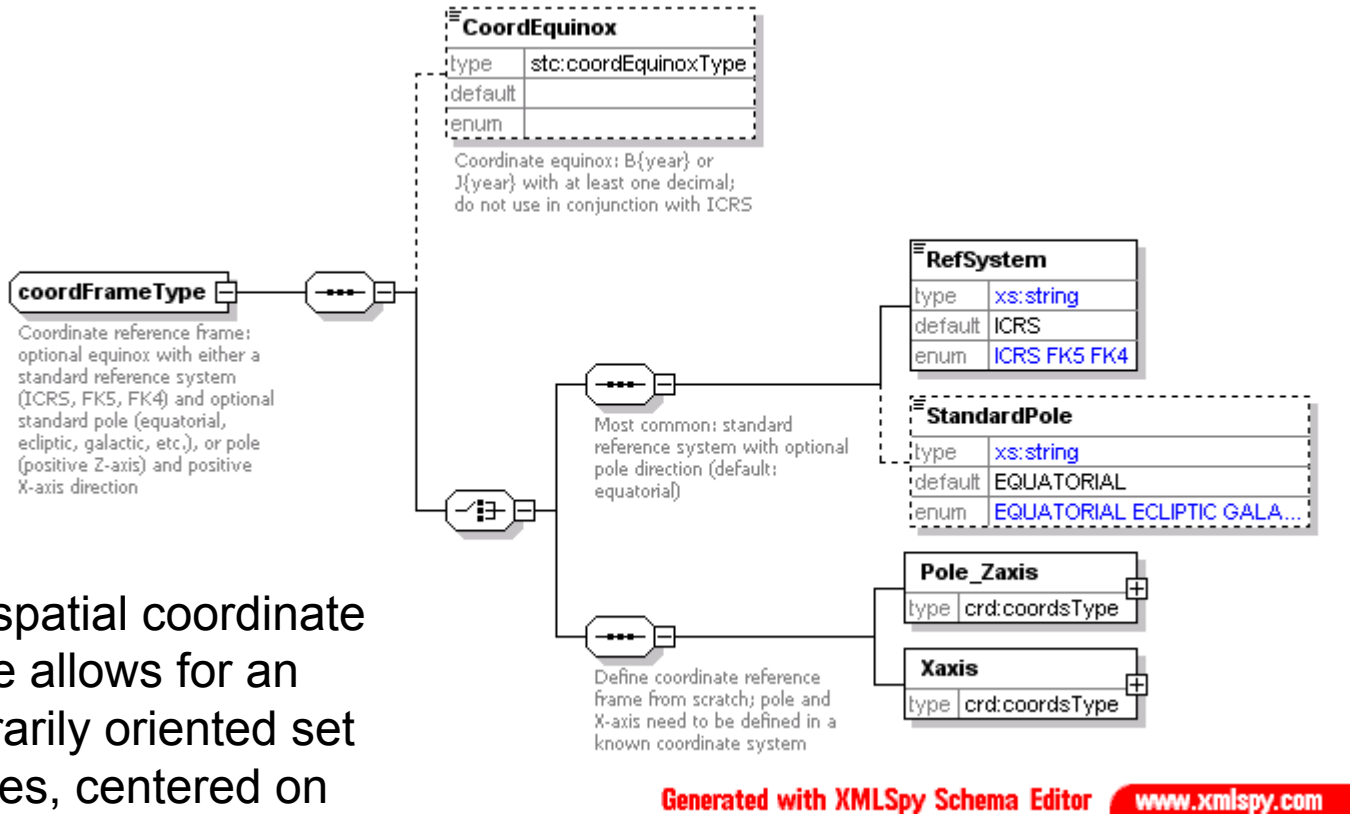




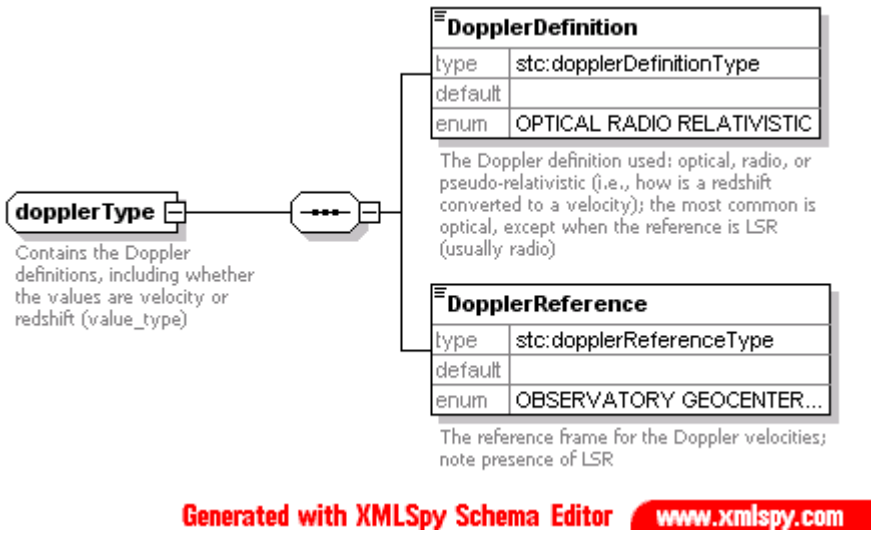
TimeRefPosition is the place where time is measured – either a name from an enumerated list, or a specified position in space.



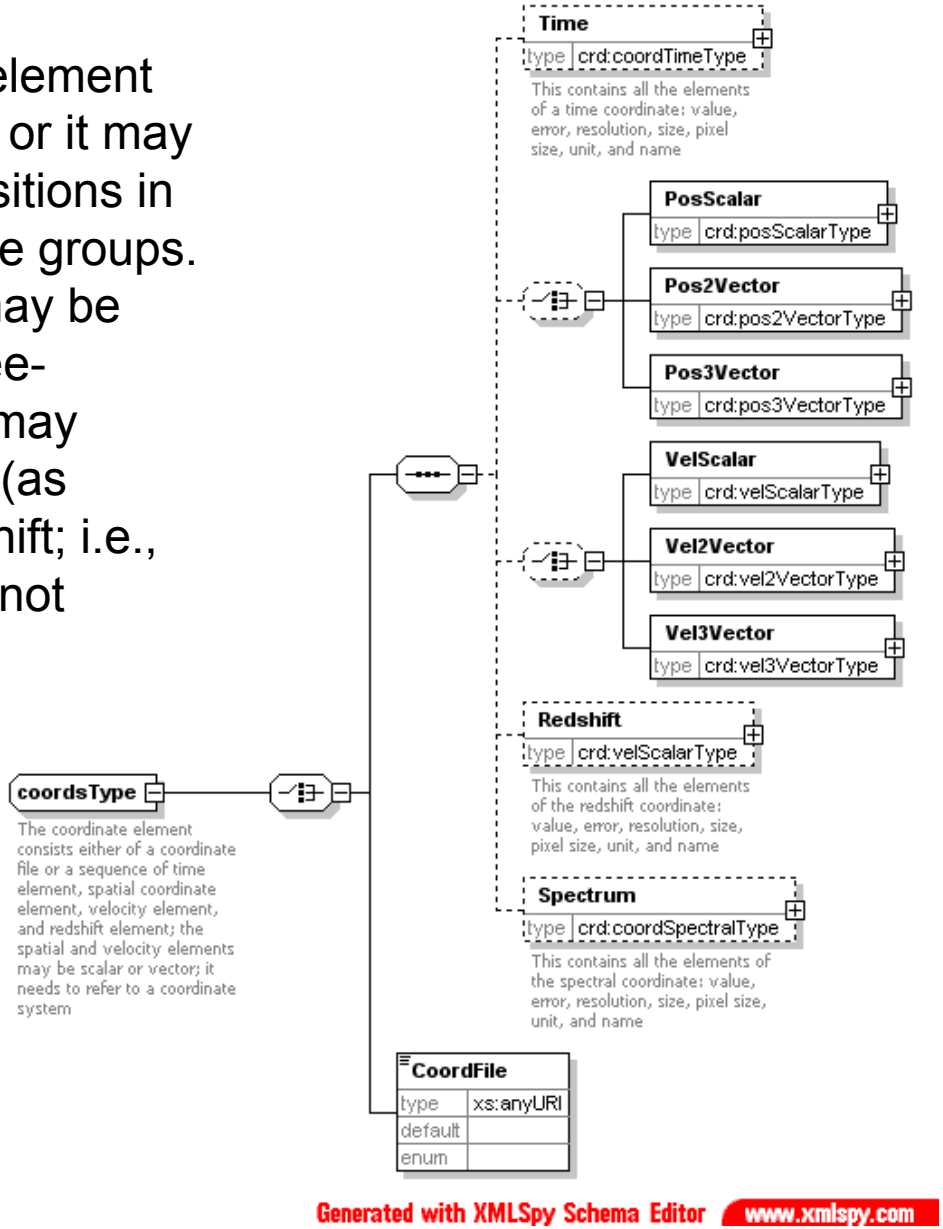
CoordRefPosition is the origin of the spatial coordinate system – either a name from an enumerated list, or a specified position in space. For the coordinate frame, see the next page.



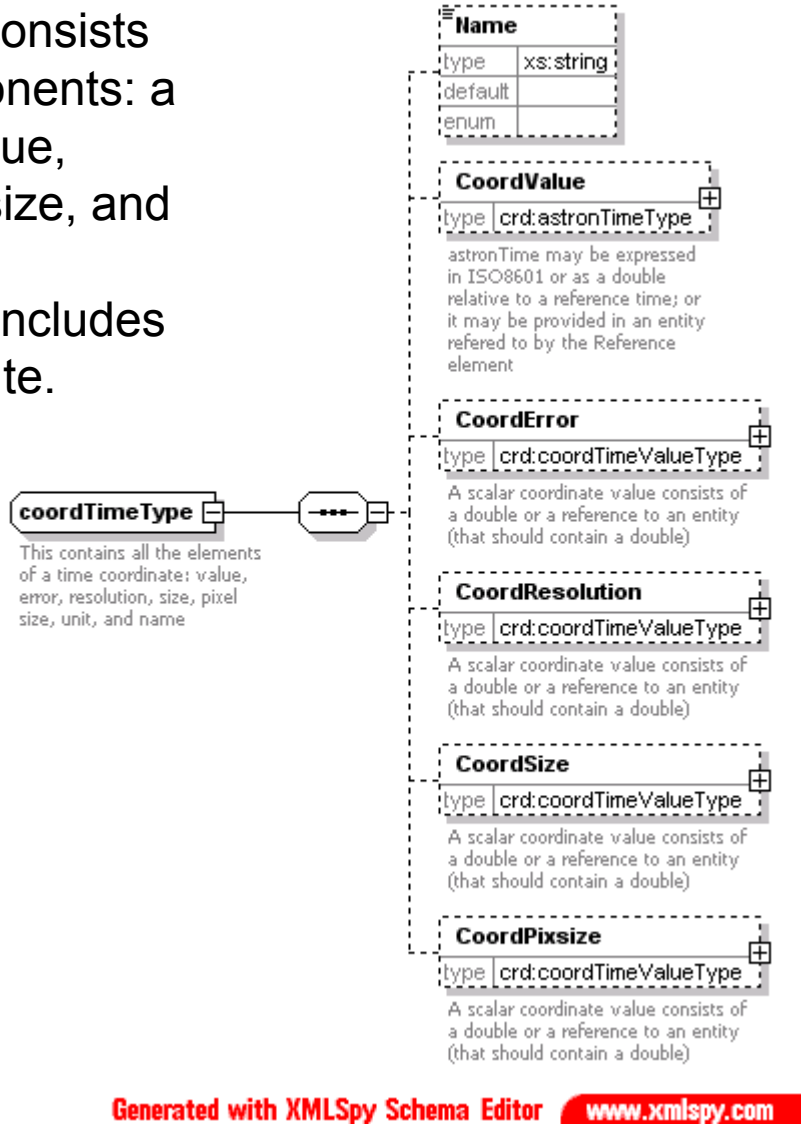
The spatial coordinate frame allows for an arbitrarily oriented set of axes, centered on an origin (see previous page).


DopplerDefinition
type
stc:dopplerDefinitionType
default
enum
OPTICAL RADIO RELATIVISTIC
The Doppler definition used: optical, radio, or pseudo-relativistic (i.e., how is a redshift converted to a velocity); the most common is optical, except when the reference is LSR (usually radio)
dopplerType
Contains the Doppler definitions, including whether the values are velocity or redshift (value_type)
DopplerReference
type
stc:dopplerReferenceType
default
enum
OBSERVATORY GEOCENTER...
The reference frame for the Doppler velocities; note presence of LSR
Generated with XMLSpy Schema Editor
www.xmlspy.com

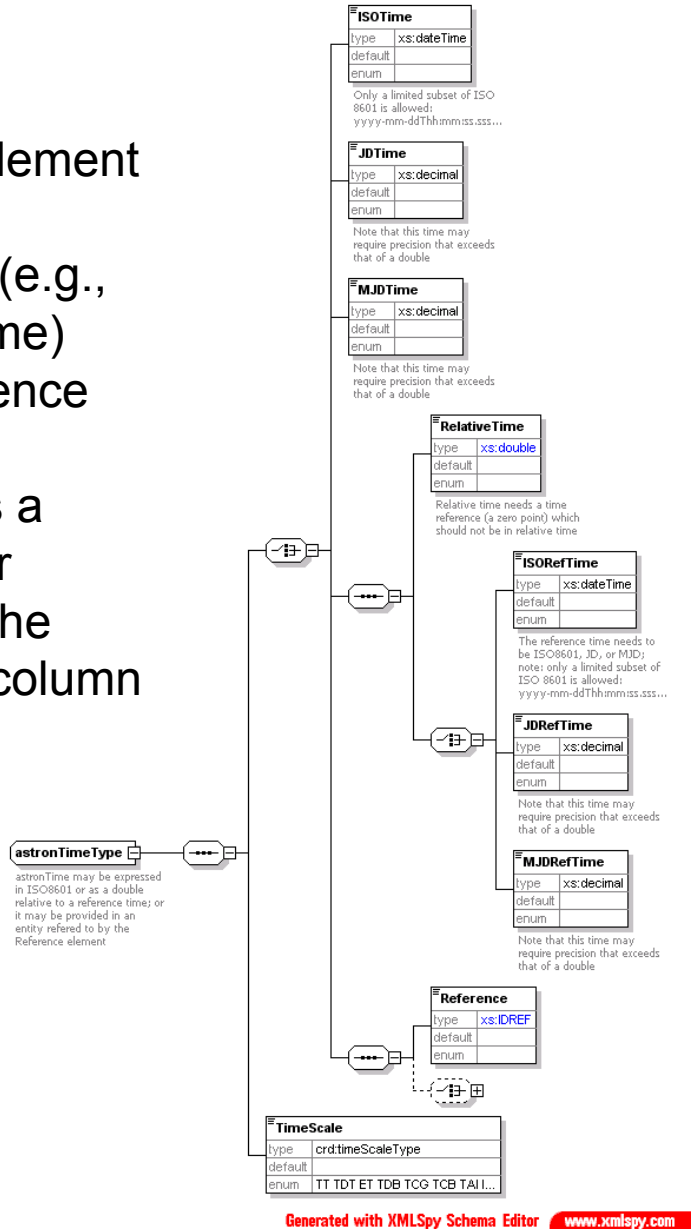
The coordinates element may refer to a file or it may contain actual positions in the four coordinate groups. Spatial position may be one-, two-, or three-dimensional and may include velocities (as distinct from redshift; i.e., physical velocity, not Doppler).



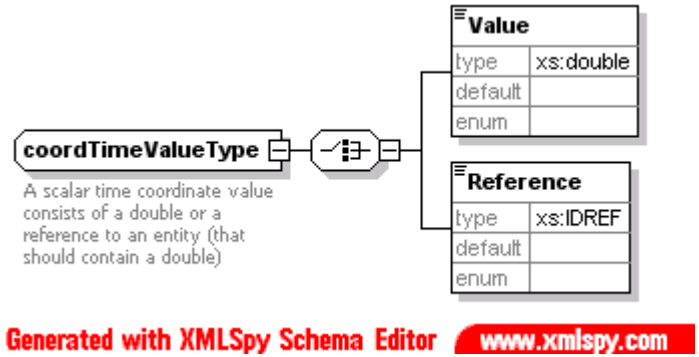
Each coordinate consists of up to six components: a name (*UCD*?), value, error, resolution, size, and pixel size.
Each component includes its own unit attribute.



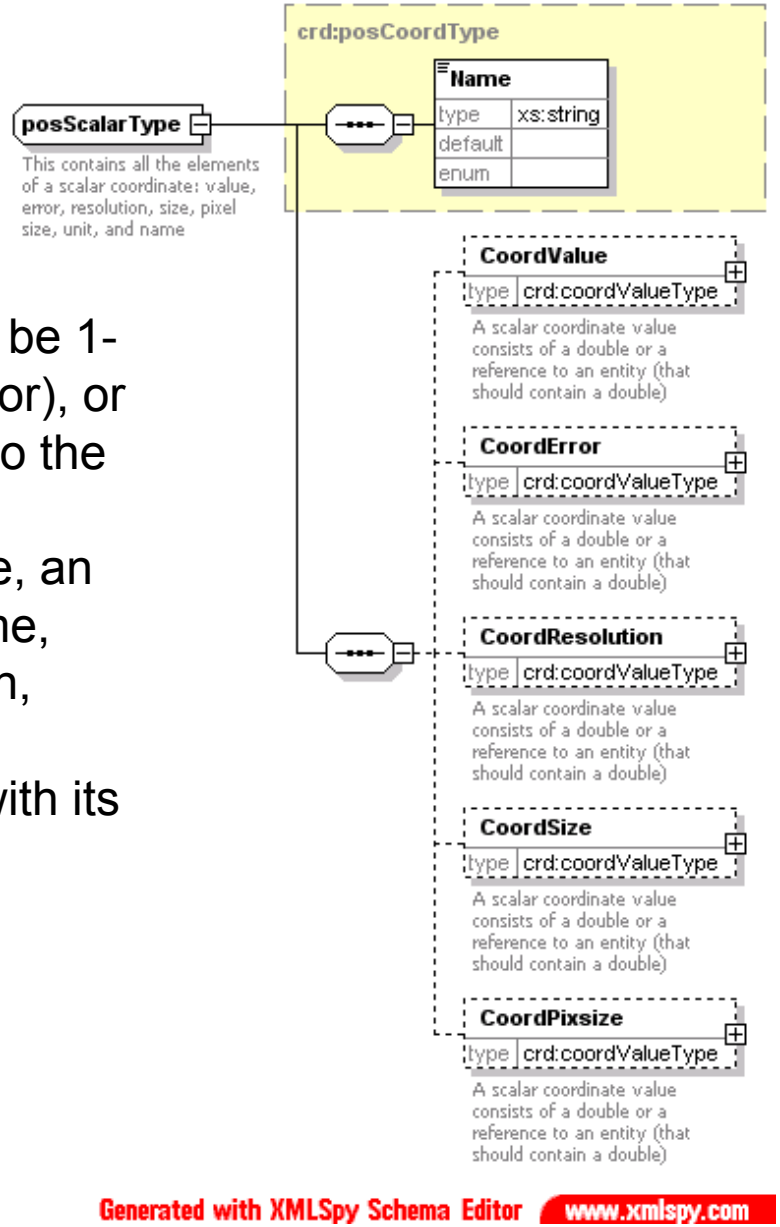
A general-purpose astronomical time element type.
Relative time types (e.g., Mission Elapsed Time) require a time reference that is not relative.
A *Reference* means a reference to another element that holds the value (e.g., a table column in the document).

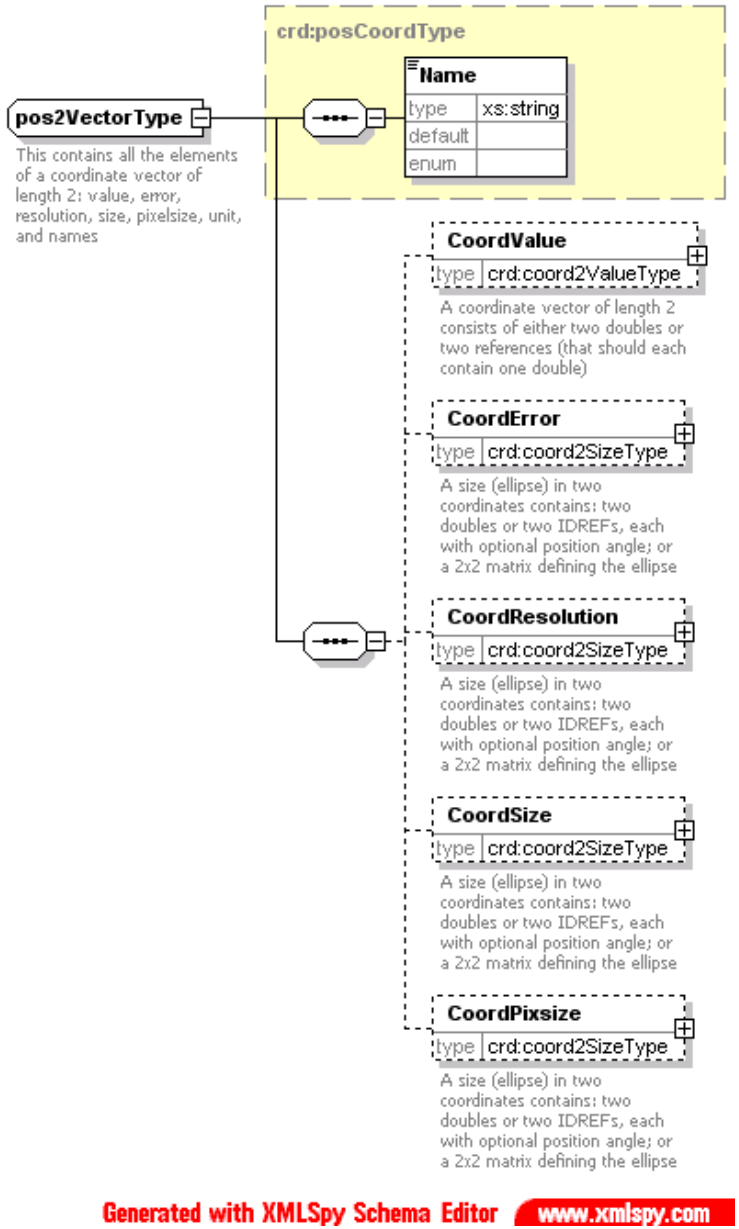


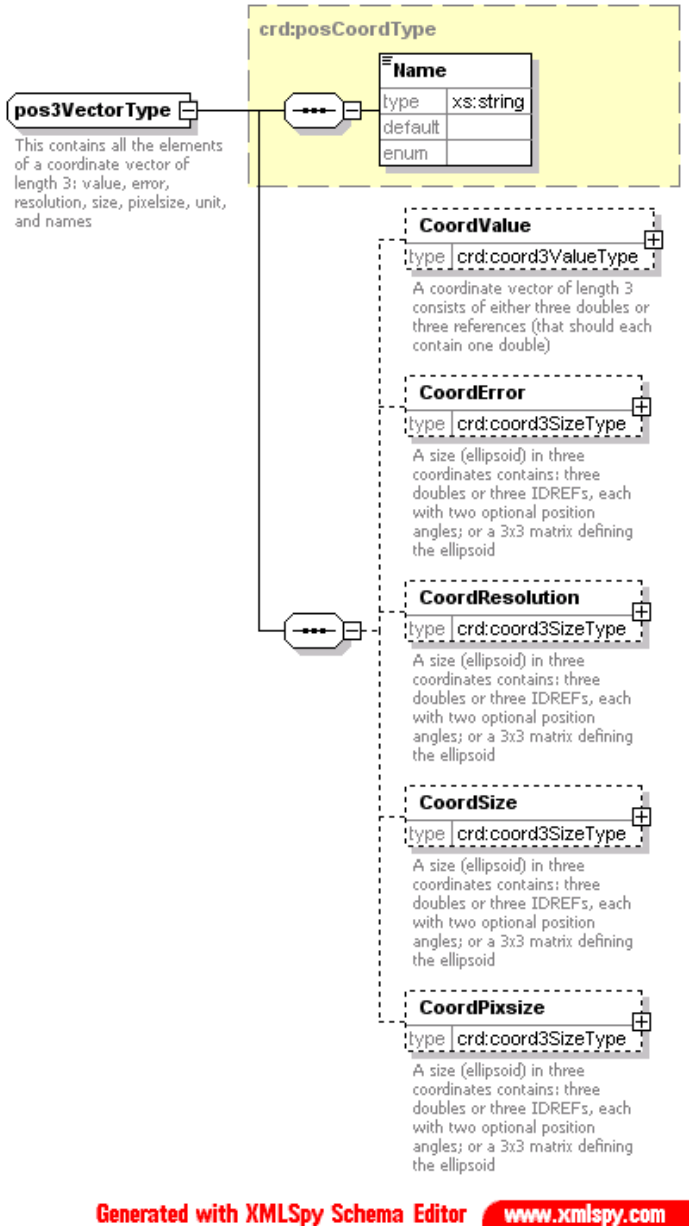
Used for error, resolution, size, pixel size; either an actual value or *Reference* (meaning a reference to another element that holds the value).



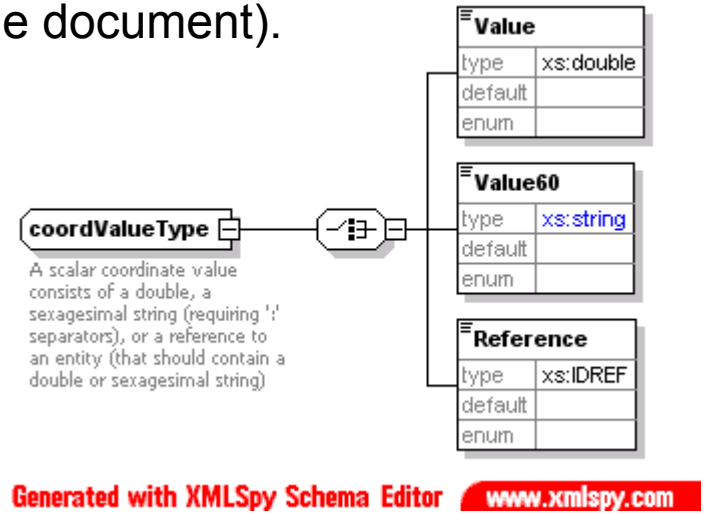
Spatial positions may be 1-D (scalar), 2-D (2vector), or 3-D (3vector); see also the next two pages.
As is the case for time, an element contains name, value, error, resolution, size, and pixel size components – each with its own unit attribute.



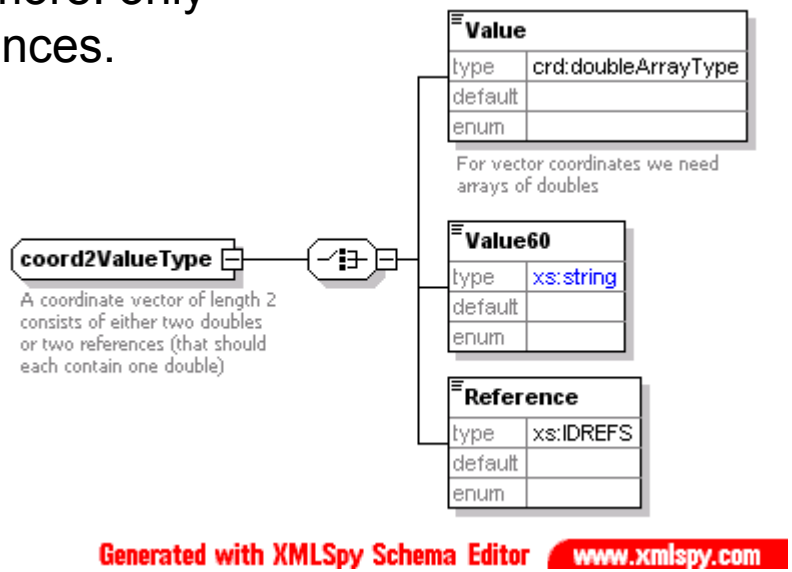

crd:posCoordType
Name
type xs:string
default
enum
pos2VectorType
This contains all the elements of a coordinate vector of length 2: value, error, resolution, size, pixelsize, unit, and names
CoordValue
type crd:coord2ValueType
A coordinate vector of length 2 consists of either two doubles or two references (that should each contain one double)
CoordError
type crd:coord2SizeType
A size (ellipse) in two coordinates contains: two doubles or two IDREFs, each with optional position angle; or a 2x2 matrix defining the ellipse
CoordResolution
type crd:coord2SizeType
A size (ellipse) in two coordinates contains: two doubles or two IDREFs, each with optional position angle; or a 2x2 matrix defining the ellipse
CoordSize
type crd:coord2SizeType
A size (ellipse) in two coordinates contains: two doubles or two IDREFs, each with optional position angle; or a 2x2 matrix defining the ellipse
CoordPixsize
type crd:coord2SizeType
A size (ellipse) in two coordinates contains: two doubles or two IDREFs, each with optional position angle; or a 2x2 matrix defining the ellipse
Generated with XMLSpy Schema Editor
www.xmlspy.com


crd:posCoordType
Name
type xs:string
default
enum
pos3VectorType
This contains all the elements of a coordinate vector of length 3: value, error, resolution, size, pixelsize, unit, and names
CoordValue
type crd:coord3ValueType
A coordinate vector of length 3 consists of either three doubles or three references (that should each contain one double)
CoordError
type crd:coord3SizeType
A size (ellipsoid) in three coordinates contains: three doubles or three IDREFs, each with two optional position angles; or a 3x3 matrix defining the ellipsoid
CoordResolution
type crd:coord3SizeType
A size (ellipsoid) in three coordinates contains: three doubles or three IDREFs, each with two optional position angles; or a 3x3 matrix defining the ellipsoid
CoordSize
type crd:coord3SizeType
A size (ellipsoid) in three coordinates contains: three doubles or three IDREFs, each with two optional position angles; or a 3x3 matrix defining the ellipsoid
CoordPixsize
type crd:coord3SizeType
A size (ellipsoid) in three coordinates contains: three doubles or three IDREFs, each with two optional position angles; or a 3x3 matrix defining the ellipsoid
Generated with XMLSpy Schema Editor
www.xmlspy.com

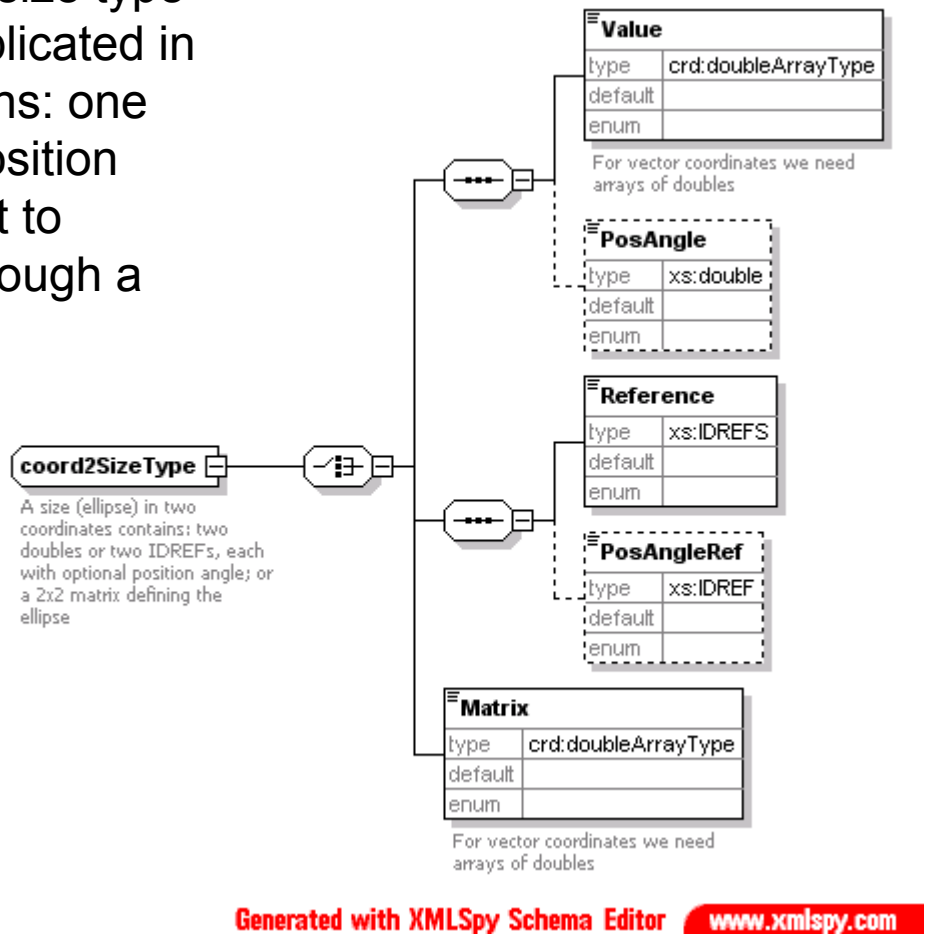
For spatial coordinates we allow decimal or sexagesimal values, as well as references. A *Reference* means a reference to another element that holds the value (e.g., a table column in the document).



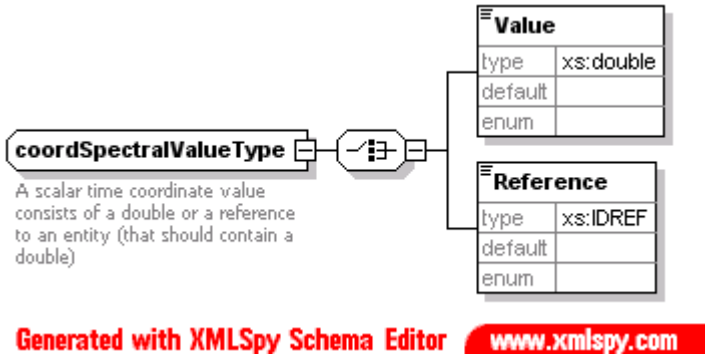
The same comments apply as on the previous page, except that *Value* will hold two doubles and *Value60* a string with two sexagesimal numbers.
The same applies to the 3-D case, except that sexagesimal is not allowed anymore: only doubles and references.



The coordinate size type gets more complicated in 2 or 3 dimensions: one needs to add position angles, or resort to specification through a matrix.

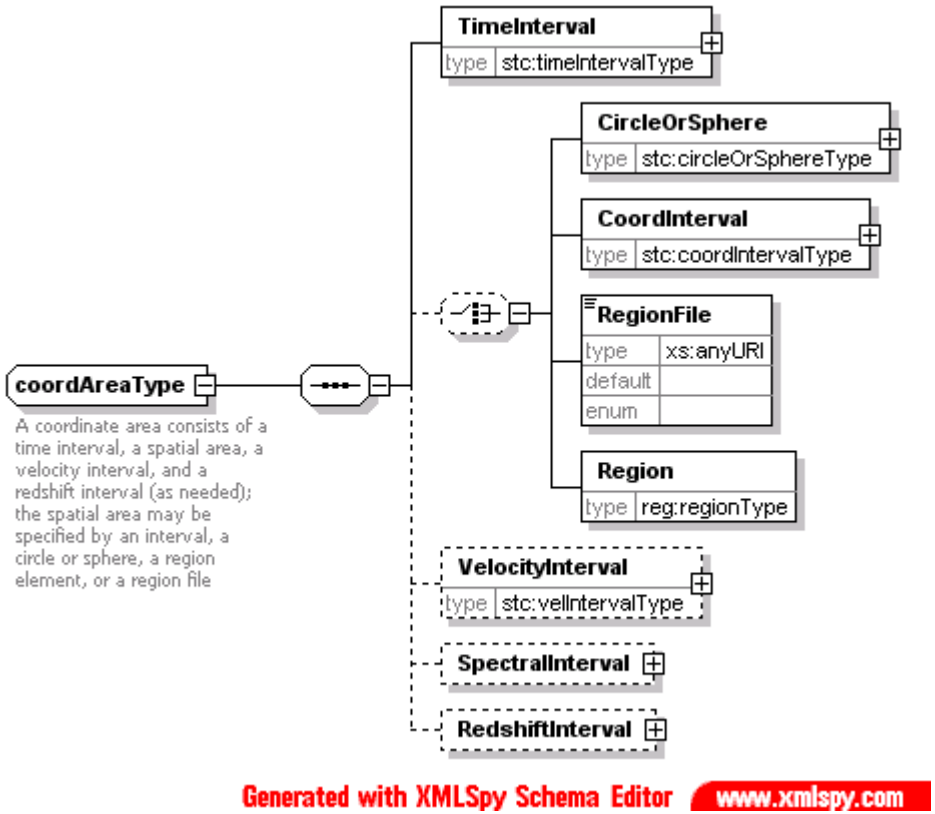


Used for spectral values; either an actual value or *Reference* (meaning a reference to another element that holds the value).

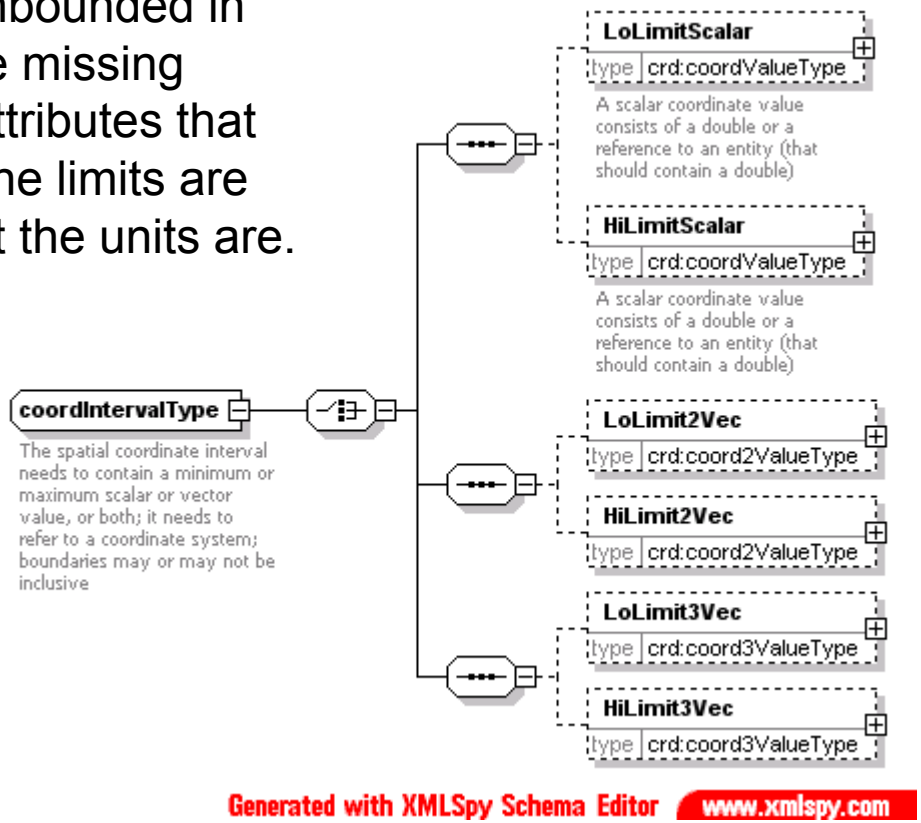


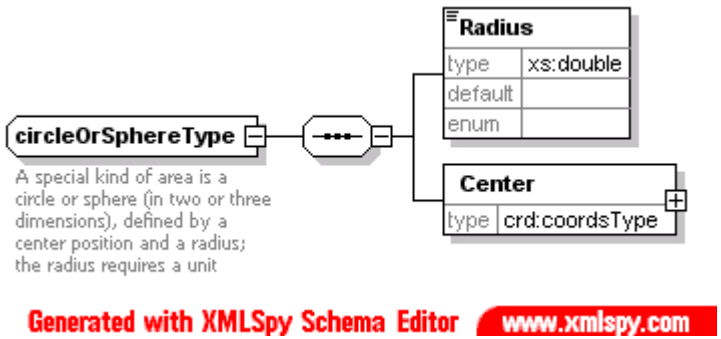
CoordArea

- The *CoordArea* consists of:
  - Time interval
  - Velocity interval (one- to three-dimensional)
  - Spectral interval
  - Redshift / Doppler velocity interval
  - Spatial area:
    - Straight interval (one- to three dimensional)
    - Circle (2-D) or sphere (3-D) specified by center and radius
    - *Region* (see region element specification from region.xsd)
    - FITS region file


TimeInterval
type stc:timeIntervalType
CircleOrSphere
type stc:circleOrSphereType
CoordInterval
type stc:coordIntervalType
RegionFile
type xs:anyURI
default
enum
Region
type reg:regionType
coordAreaType
A coordinate area consists of a time interval, a spatial area, a velocity interval, and a redshift interval (as needed); the spatial area may be specified by an interval, a circle or sphere, a region element, or a region file
VelocityInterval
type stc:velIntervalType
SpectralInterval
RedshiftInterval
Generated with XMLSpy Schema Editor
www.xmlspy.com

An interval is defined by a (1-D, 2-D, or 3-D) lower limit and/or an upper limit (i.e., unbounded in the direction of the missing limit), as well as attributes that indicate whether the limits are inclusive and what the units are.



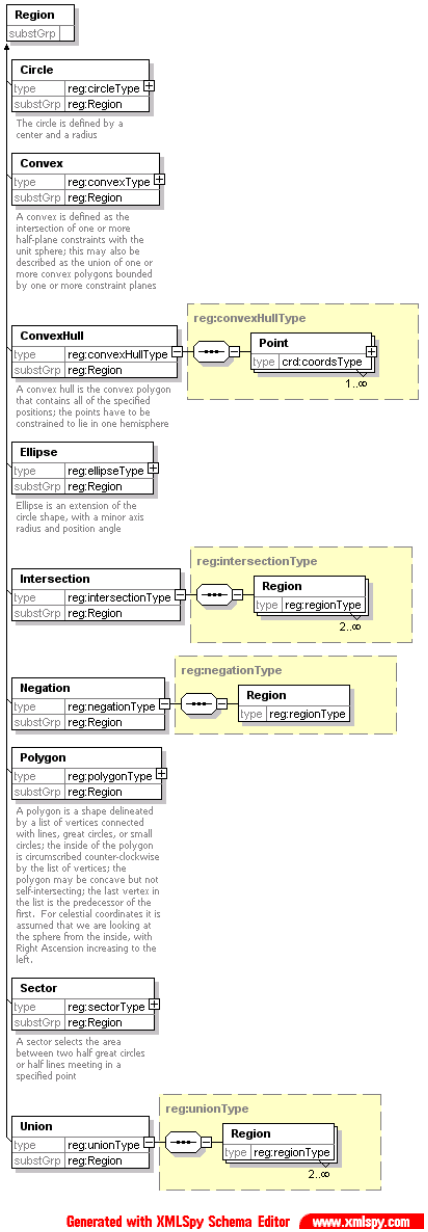

Radius
type
xs:double
default
enum
circleOrSphereType
A special kind of area is a circle or sphere (in two or three dimensions), defined by a center position and a radius; the radius requires a unit
Center
type
crd:coordsType
Generated with XMLSpy Schema Editor
www.xmlspy.com

Spatial Region Specification

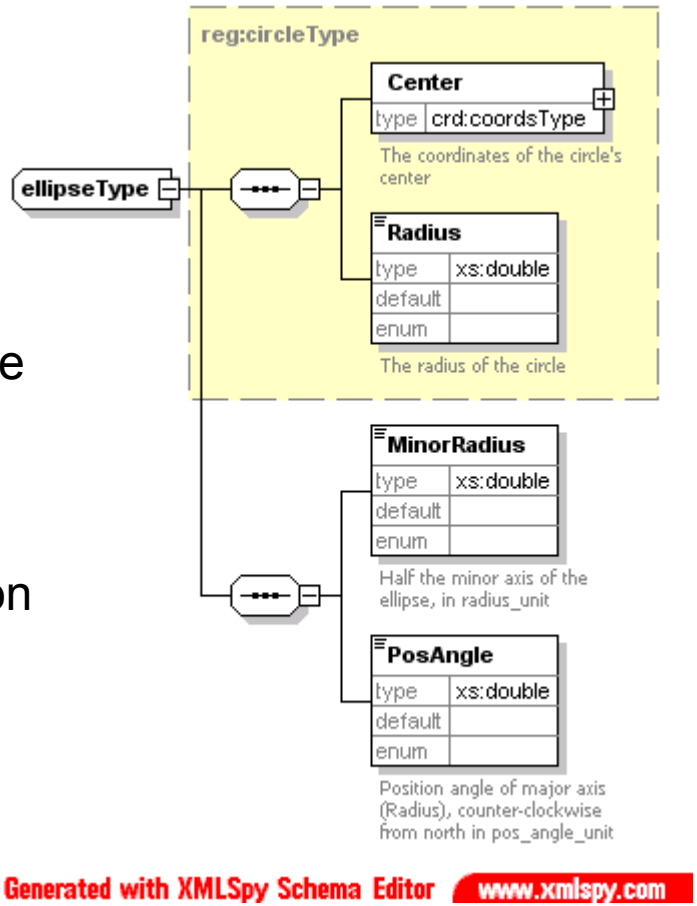
region.xsd

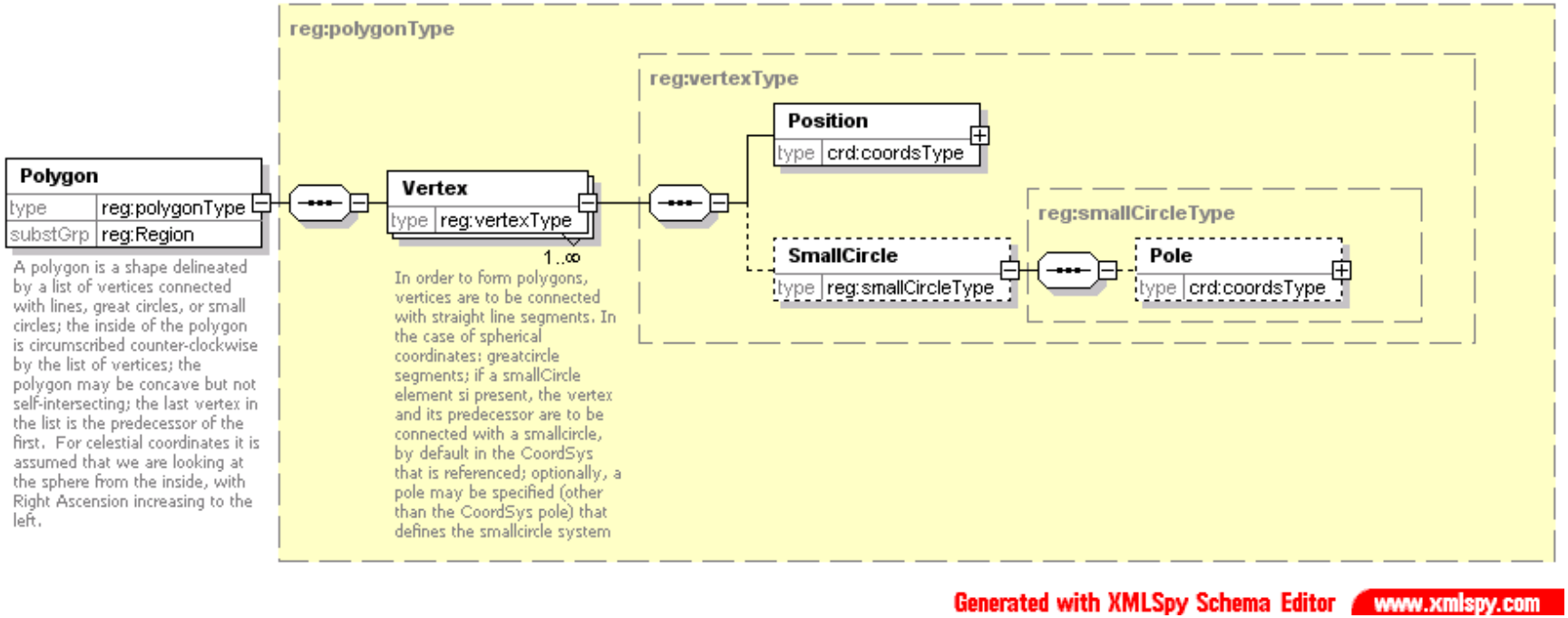
- Building blocks:
  - *Shapes*
    - Polygon
    - Circle
    - Ellipse
    - Sector
    - Constraint (on unit sphere only)
    - Convex (on unit sphere only)
    - Convex hull
  - *Region operators*
    - Negation (operand: one region): logical NOT
    - Union (operand: two or more regions): logical OR
    - Intersection (operand: two or more regions): logical AND
- A ***region*** is defined as either a *shape* or the result of a *region operator*

This diagram is explained on the previous page. *Region* is a *substitutionGroup* with nine derived types.



Strictly speaking, a *circle* is a special case of an *ellipse*, but for practical reasons we have defined the *ellipse* as an extension of the *circle*

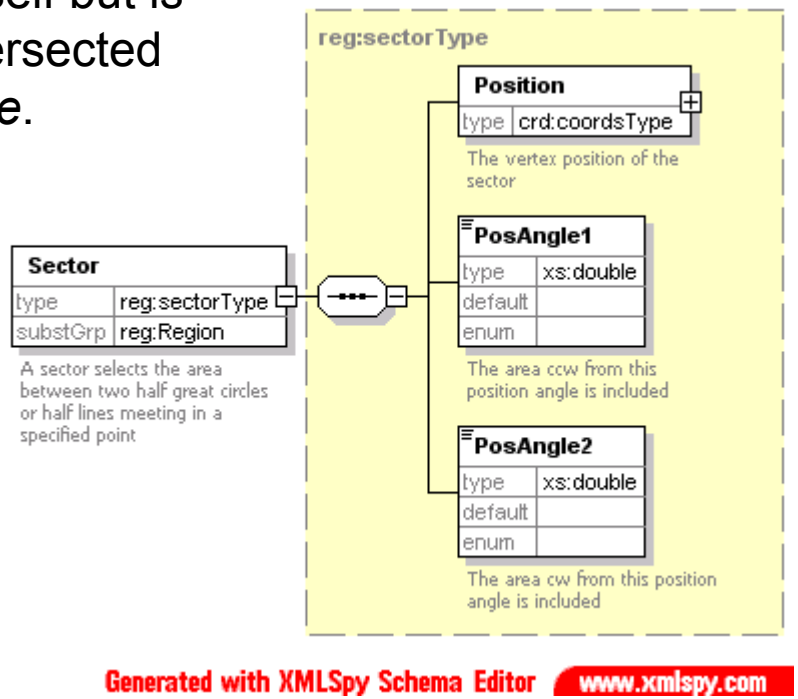


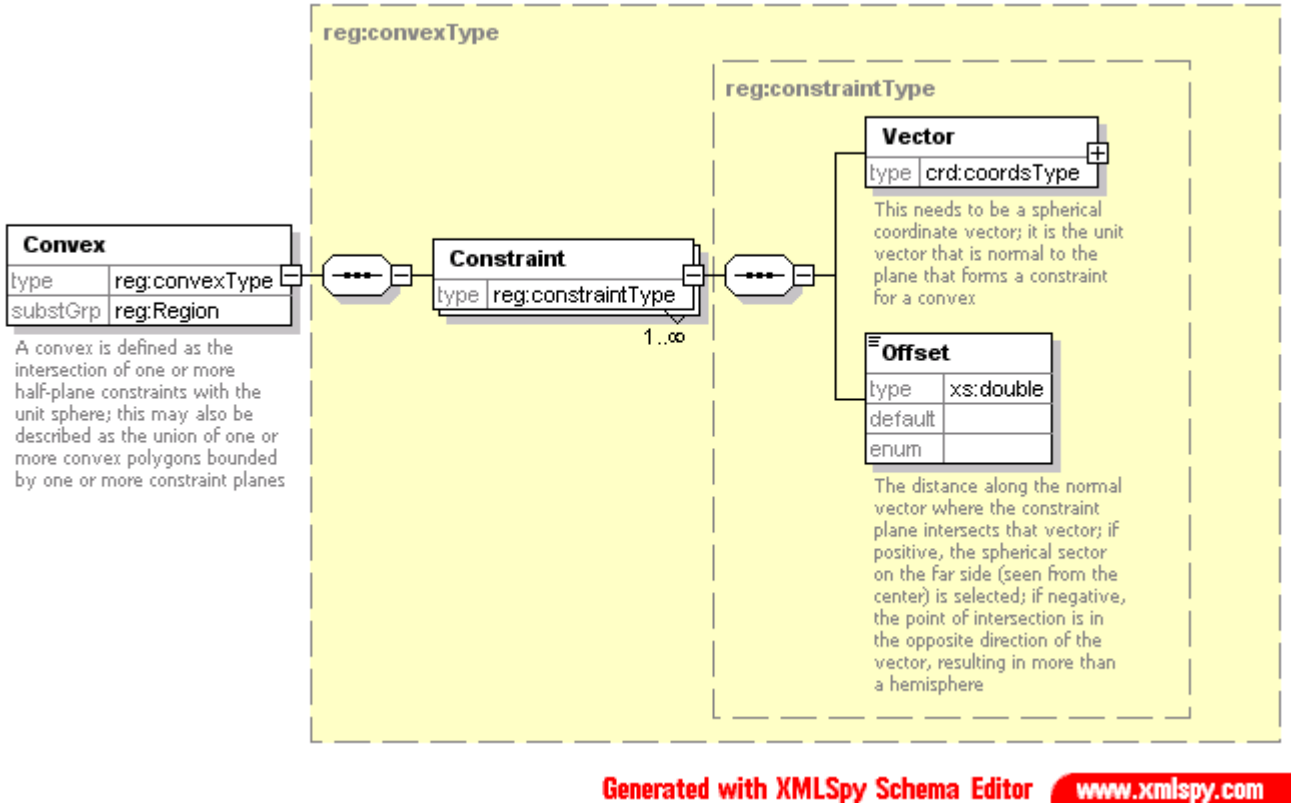


A *polygon* is specified by a list of vertices. It is defined as the shape enclosed by connecting successive vertices by straight lines (Cartesian coordinates) or by either great-circles or small-circles (spherical coordinates). The small-circle option is indicated by the presence of a *SmallCircle* element. The default pole of the small-circle system is the pole of the *CoordSys*; optionally, a different pole may be specified. It is the responsibility of the *region* constructor to make sure that successive vertices that are to be connected with a small-circle actually lie on the same small-circle or parallel.

The inside of the polygon is circumnavigated counter-clockwise by going through the list of vertices. Polygons may be concave but may not be self-intersecting.

A *sector* is the area between two half-lines emanating from the same point.
It may be used by itself but is more likely to be intersected with a *circle* or *ellipse*.





A *constraint* is the part of the unit sphere that is cut off by intersecting it with a plane that is normal to a unit *vector* and that intersects that normal *vector* at a certain *offset* distance from the center of the sphere. The selected spherical section is the one that contains the normal unit vector's point.
A *convex* is the intersection of one or more constraints. The result will be empty, a single point, the entire sphere, or one or more convex polygons.