Space-Time (and other) Coordinate Metadata: WCS is supported

Arnold Rots

SAO

Context

- Earlier this week there were statements that STC does not support WCS mappings
- This is not correct
- Since that session was never finished and I did not have a chance to correct the impression, I decided just to post this set of slides
- The basic problem is that STC was designed to provide data characterization as far as coordinates are concerned, in a self-consistent manner – and that included WCS
- But that characterization role was dropped by some implementers and as a result we ended up with a broken system

STC is Self-contained and Complete

- STC was intended to provide a complete, self-contained, and self-consistent metadata description of the coordinate systems, including a characterization
- In that sense, it is the complete equivalent of FITS WCS – with some additional information and features
- The relevant characterization object in the STC schema is the ObsDataLocation element

ObsDataLocation Contains:

- Specification the coordinate systems
- The observatory location
- The WCS location of the observation (the volume occupied by the observation in WCS space)
- Additional information: resolution, errors, etc.
- The pixel coordinate space
- The projection of the pixel coordinate space onto WCS

The Problem

- I argued from the beginning that one needed a complete model for space-time and related coordinates (independent of the question whether all of it needed to be implemented immediately)
- People didn't think that was necessary and preferred to cherry-pick: here a coordinate system, there a coordinate, somewhere else a region – dropping the characterization

The Problem (cont'd)

- That destroyed the completeness and self-consistency – the integrity – of the data model
- The complaint now is that STC does not provide WCS support
- That is simply not true; to be blunt: those who say that have not taken the trouble to read the standard document and try to understand it

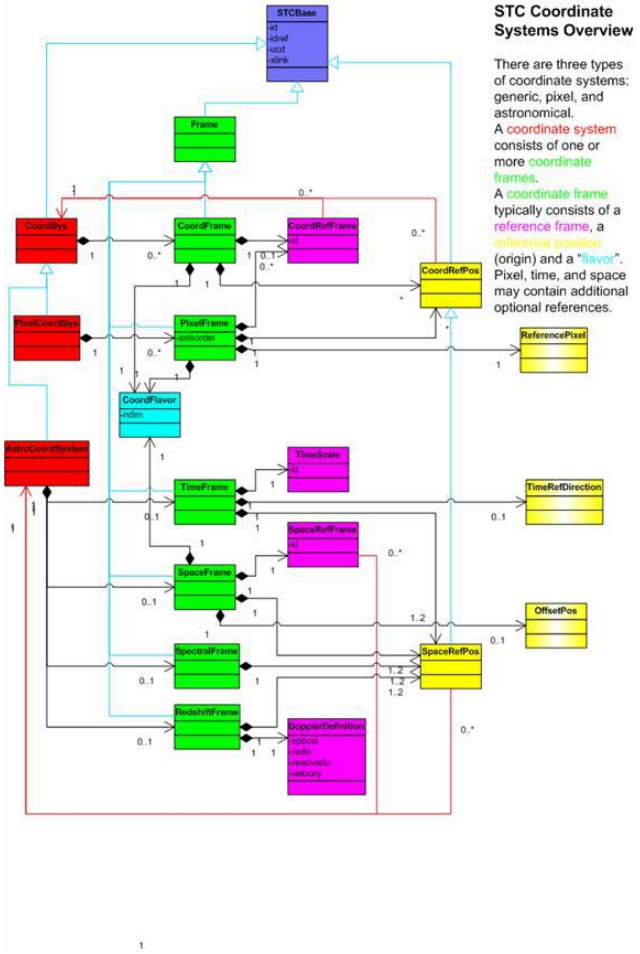
The Solution

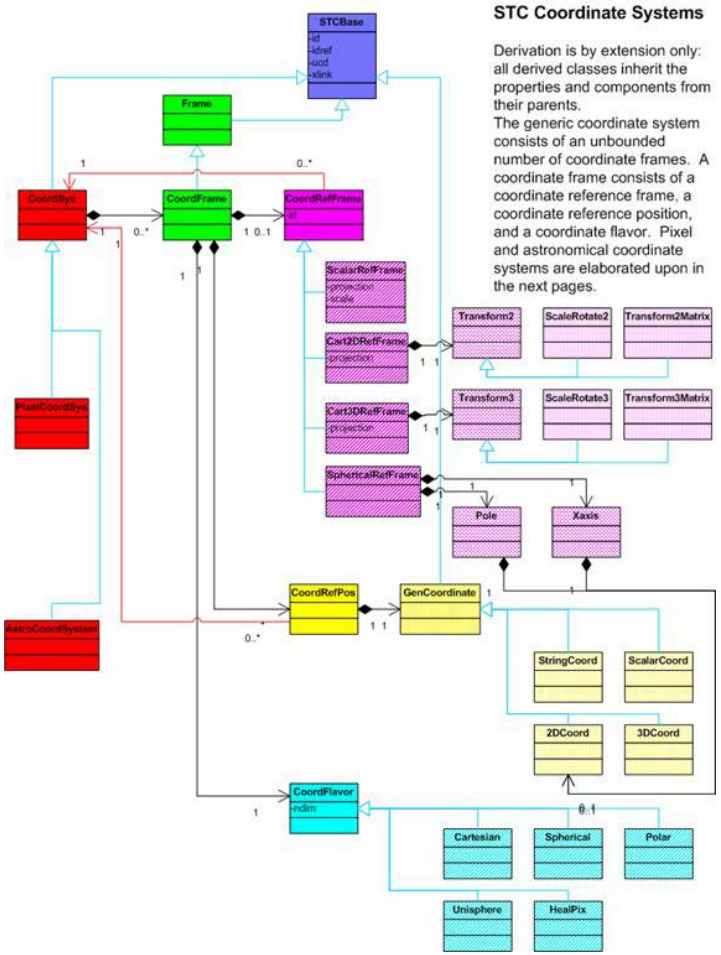
- At this point it is hard to fit the WCS support back in, because the integrity of the data model has been lost by cherry-picking a few scattered elements and abandoning the characterization component; utypes won't help, either
- The problem could be solved by returning to the STC standard and use the ObsDataLocation element as it was intended
- I'd be happy to assist with that

From the STC Document

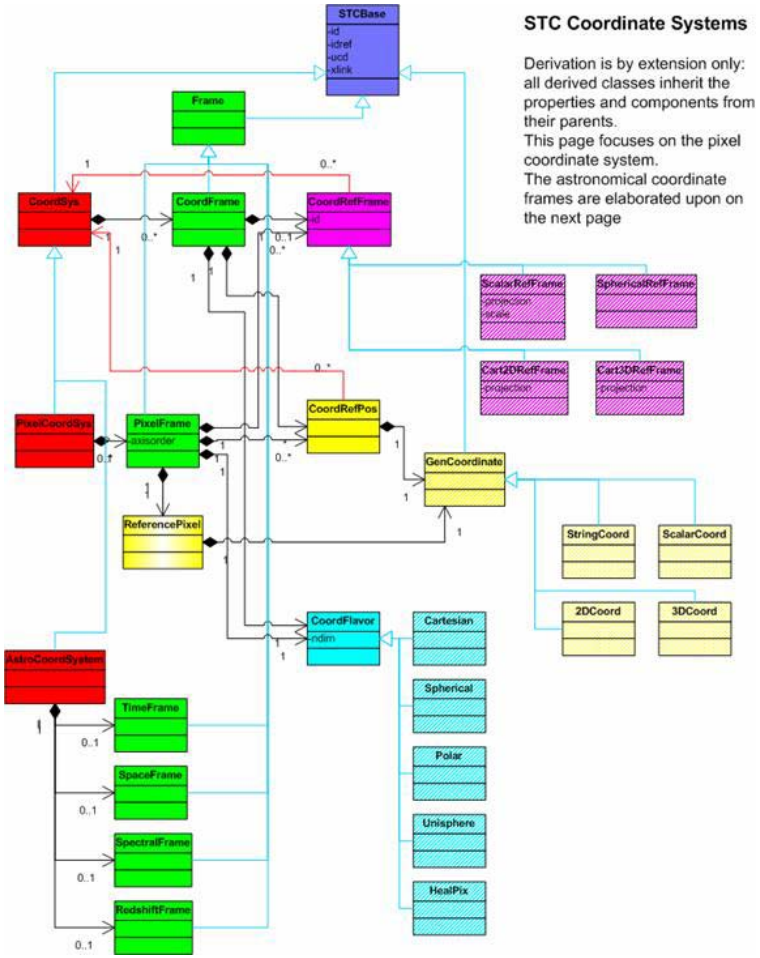
- Following are some pages from the STC document:
  - The model, from Appendix A
  - An example of full WCS specification, from Appendix B.2

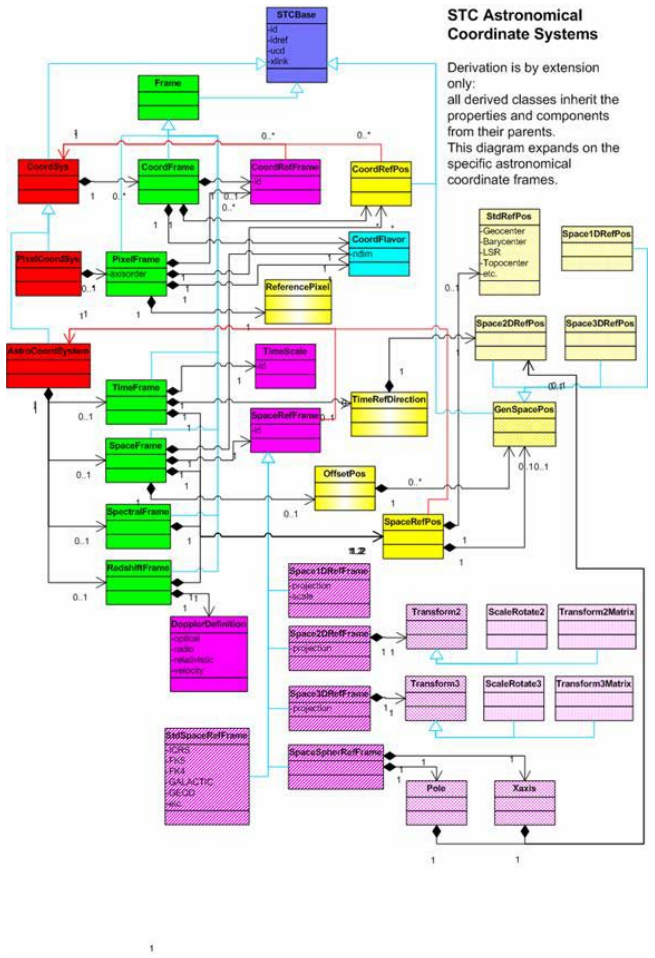
STC Overview / Coordinate Systems





Pixel Coord Systems / Astonomical C S





Example B.2: Observatory Location & Observation Location

```
<?xml version="1.0" encoding="UTF-8"?>
 <!--
   The ObsDataLocation element is the container for the complete
   metadata description of an observational dataset, in this case
   an image.
   It specifies the various required namespaces (for STC specifically:
   STC and Xlink, and the location of the STC schema).
   ObsDataLocation contains two main components: ObservatoryLocation
   and ObservationLocation.
 -->
<ObsDataLocation xmlns="http://www.ivoa.net/xml/STC/stc-v1.30.xsd"
            xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
            instance" xsi:schemaLocation="http://www.ivoa.net/xml/STC/stc-v1.30.xsd
            http://www.ivoa.net/xml/STC/stc-v1.30.xsd">
  <!--
    The location of the observatory is specified by a set of
    Spatial coordinates and an associated coordinate system
    - geodetic, in this case.
  -->
 <ObservatoryLocation id="KPNO">
  <AstroCoordSystem id="TT-GEOD-TOPO">
   <TimeFrame>
    <TimeScale>TT</TimeScale>
    <TOPOCENTER/>
   </TimeFrame>
   <SpaceFrame>
    <GEO_D/>
    <TOPOCENTER/>
    <SPHERICAL coord_naxes="3"/>
   </SpaceFrame>
  </AstroCoordSystem>
  <AstroCoords coord_system_id="TT-GEOD-TOPO">
   <Position3D>
    <Value3>
     <C1 pos_unit="deg">248.4056</C1>
     <C2 pos_unit="deg">31.9586</C2>
     <C3 pos_unit="m">2158</C3>
    </Value3>
   </Position3D>
  </AstroCoords>
 </ObservatoryLocation>
<!--
   The observation's location and the volume it occupies in
   coordinate space is specified by a set of coordinates and their
   properties (errors, resolutions, etc.) and the area or interval
   covered in each coordinate
   - with the associated coordinate system.
   Note that a generic coordinate frame has been defined to handle
   the brightness axis.
 -->
 <ObservationLocation id="M81">
  <AstroCoordSystem id="TT-ICRS-WAVELENGTH-TOPO">
   <CoordFrame ucd="em..." id="brightness">
    <Name>Brightness</Name>
    <ScalarRefFrame>
     <Scale>1.0</Scale>
    </ScalarRefFrame>
    <CARTESIAN coord_naxes="1"/>
   </CoordFrame>
   <TimeFrame>
    <TimeScale>TT</TimeScale>
    <TOPOCENTER/>
   </TimeFrame>
   <SpaceFrame id="spaceFrame">
    <ICRS/>
    <TOPOCENTER/>
    <SPHERICAL coord_naxes="2"/>
   </SpaceFrame>
   <SpectralFrame>
    <TOPOCENTER/>
   </SpectralFrame>
  </AstroCoordSystem>
  <AstroCoords coord_system_id="TT-ICRS-WAVELENGTH-TOPO">
   <ScalarCoordinate unit="mJy/arcsec**2" frame_id="brightness">
    <Error>0.001</Error>
   </ScalarCoordinate>
   <Time unit="s">
    <TimeInstant>
     <ISOTime>2004-07-15T08:23:56</ISOTime>
    </TimeInstant>
    <Resolution>1000</Resolution>
    <PixSize>1000</PixSize>
   </Time>
```

Example B.2: Pixel Coordinates and WCS Transformation

```
<!--
    The image is represented as a pixel array.
    The Pixel Space specifies the properties of this pixel array
    and the way it is related to the world coordinates specified
    above.
    It consists of a pixel coordinate system that defines the pixel
    Axes and their transformation on the sky (WCS) and a pixel
    Coordinate area that defines the array's dimensions.
    Note the use of frame_id to tie pixel interval to the correct
    frame; in this case it is not strictly required since there is
    only one frame, but in the general case this is the mechanism
    used to avoid ambiguity.
  -->
 <PixelSpace>
  <PixelCoordSystem id="M81Pix">
   <PixelCoordFrame id="spacepix" axis1_order="1" axis2_order="2"
   ref_frame_id="spaceFrame">
    <Cart2DRefFrame projection="TAN">
     <Transform2 unit="deg">
      <C1>0.0001</C1>
      <C2>0.0001</C2>
     </Transform2>
    </Cart2DRefFrame>
    <CoordRefPos>
     <Vector2DCoordinate unit="deg">
      <Value2 idref="Center" xsi:nil="true"/>
     </Vector2DCoordinate>
    </CoordRefPos>
    <CARTESIAN coord_naxes="2"/>
    <ReferencePixel>
     <Pixel2D>
      <Name1>RA</Name1>
      <Name2>Dec</Name2>
      <Value2>
       <C1>512</C1>
       <C2>512</C2>
      </Value2>
     </Pixel2D>
    </ReferencePixel>
   </PixelCoordFrame>
  </PixelCoordSystem>
  <PixelCoordArea coord_system_id="M81Pix" id="M81PixImage">
   <PixelCoord2VecInterval frame_id="spacepix">
    <LoLimit2Vec>
     <C1>1</C1>
     <C2>1</C2>
    </LoLimit2Vec>
    <HiLimit2Vec>
     <C1>1024</C1>
     <C2>1024</C2>
    </HiLimit2Vec>
   </PixelCoord2VecInterval>
  </PixelCoordArea>
 </PixelSpace>
</ObsDataLocation>
```