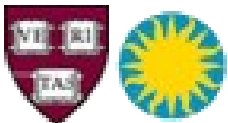


Spectral Energy Distribution Library

K. McCusker

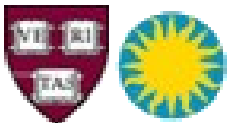
Harvard-Smithsonian Center for Astrophysics

IVOA Interop – British Columbia 5/2006



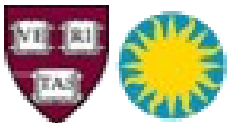
What is the SED Library?

- A set of Java classes to store
 - spectra
 - spectral energy distributions
 - time series data
 - their associated metadata
- The library implements the Spectrum Data Model Doc v0.96b and SED DM Document v0.96.



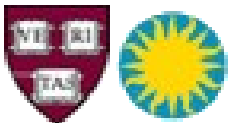
What can the SED Library do?

- Convert Spectra or SED files from and to:
 - VOTables
 - FITS files
 - XML files following the spectrum schema v0.96
- Display data to screen in table format
- Simple plots of Flux vs Spectral Coord



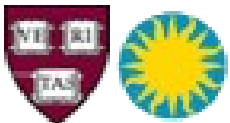
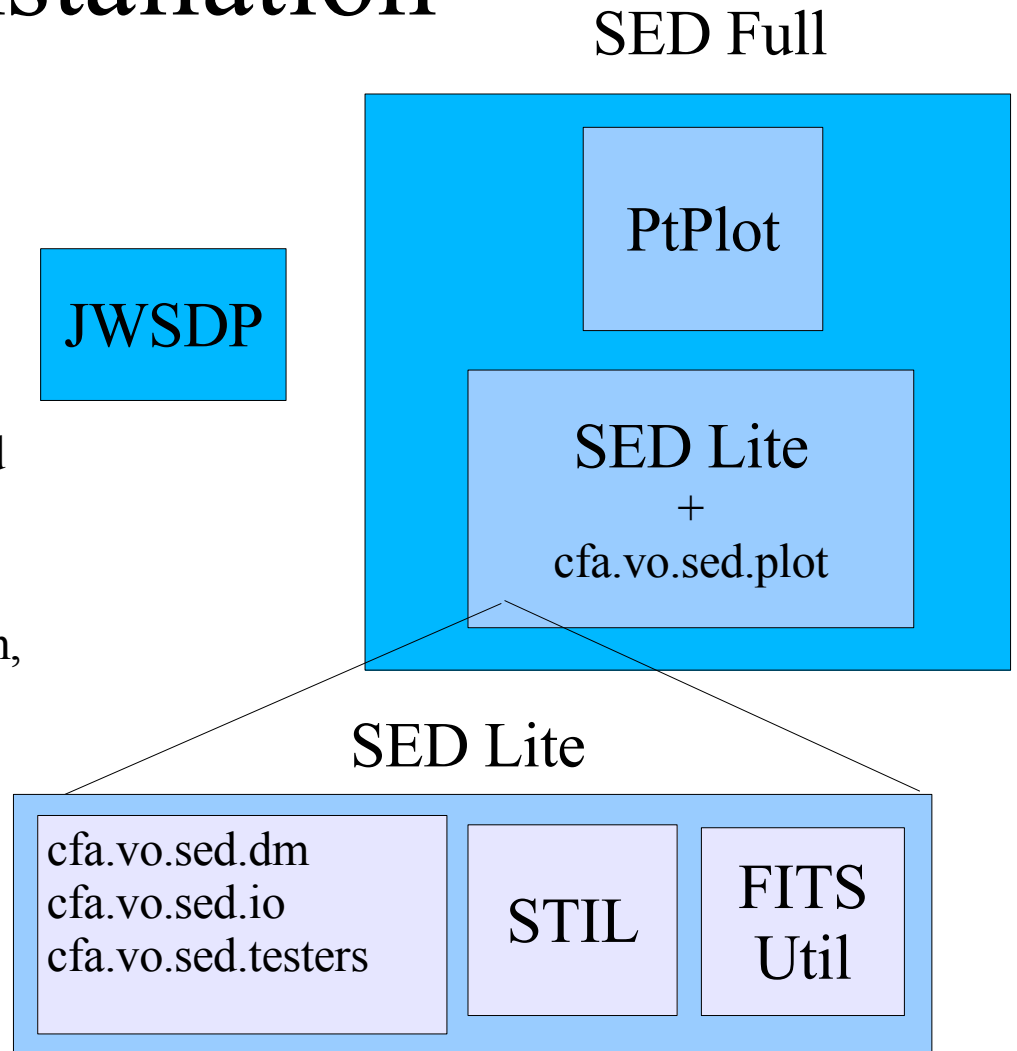
Why You Want the SED Lib...

- Suitable as middleware, on which to layer an SED application.
- Included test program easily converts file formats
- Lightweight
- Easy to install
- The library comes with detailed documentation making it easy to code to.



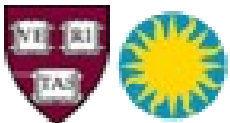
The Installation

- Visit <http://vo.cfa.harvard.edu/soft/> to download the package.
 - Choose between SED Lite and SED Full: Full includes functionality for basic plotting.
 - SED library is bundled with STIL and Java Fits Util Library.
 - If Java Web Services Developer pack is not already installed on your system, download JWSDP Bundle.
- Follow the INSTALL instructions to easily get up and running.



Intro to Using the SED Lib

- Online API Javadoc at <http://vo.cfa.harvard.edu/soft/SED/javadoc/index.html>
- Reference the Spectral Energy Distribution v0.4 Tech Spec included in the download for sample code.
- Basic steps to using the SED library:
 - Deserialize a file to SED data model objects
 - Display, Plot, Modify
 - Serialize SED to output file in given format



Sample Code

```
SED sed = null;

IWrapper wrapper = null;
IReader reader = new Reader();
ISerializer serializer = new Serializer();

wrapper = reader.read( new FileInputStream("/path/to/your/file"),
                      FileFormat.FITS );

sed = serializer.serializeToSED( wrapper );

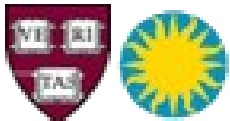
SEDSegment segment = sed.getSegment(0);

Extent extent = segment.getCoverage().getExtent();
IValue scExtentVal = extent.getSpectralCoord().getBandwidth();
Double bandwidth = null;
if(scExtentVal.isSet())
    bandwidth = (Double)scExtentVal.getDataValue(0);

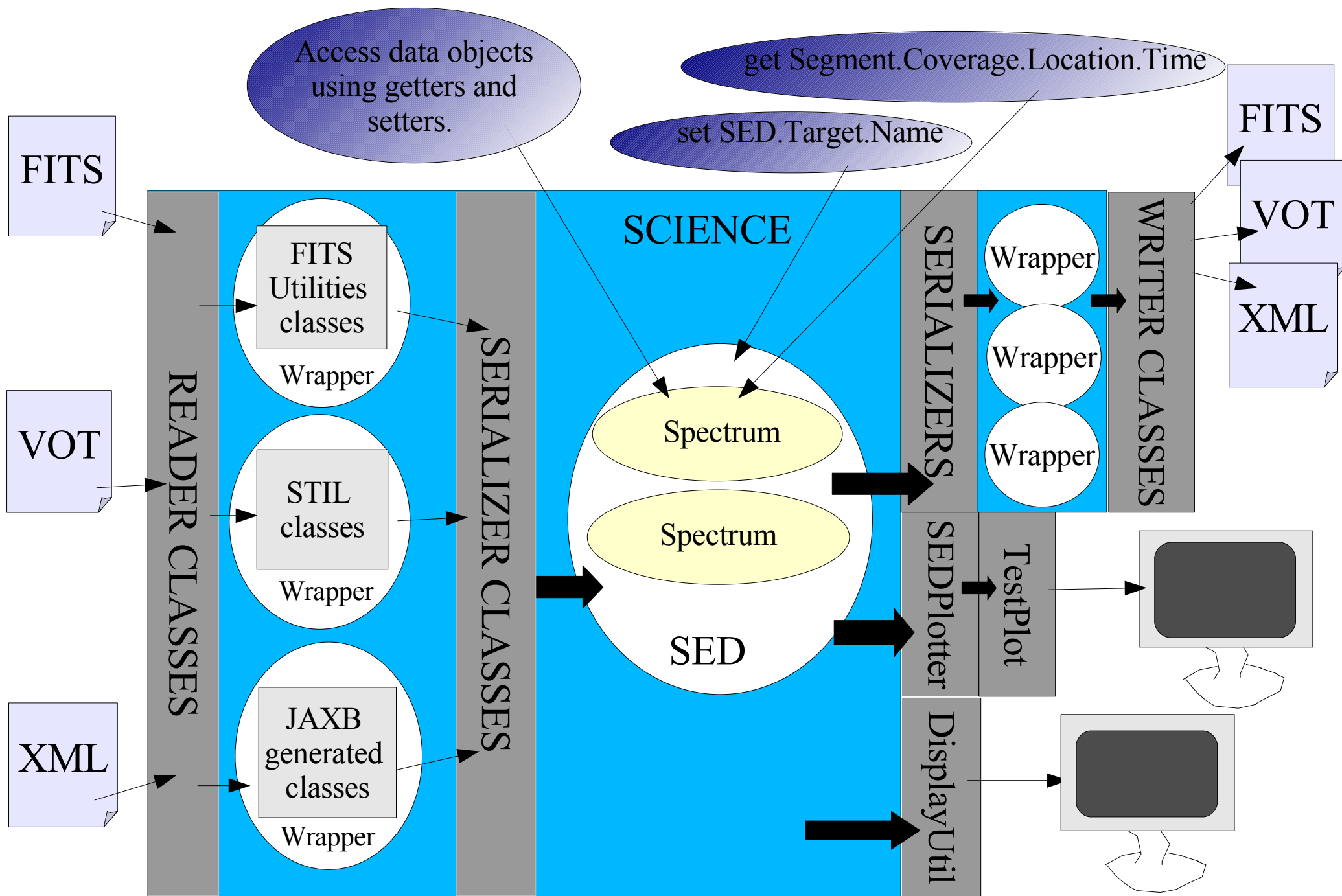
System.out.println("We have spectral extent. Bandwidth= " + bandwidth);

Double newValue = new Double(0.5);
scExtentVal.addDataValue(newValue, IDataTypes.DOUBLE);
scExtentVal.setUCD("Foo.UCD");

IWriter writer = new Writer();
String outFileName = "sed.out";
writer.write( outFileName,
             serializer.serializeFromSED( sed, FileFormat.VOTABLE ) );
```

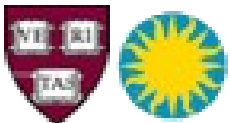


Data Flow Diagram



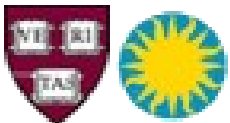
Third Party Libraries – Using the Wheel That Is Already Invented

- **STIL: Starlink Table Infrastructure Library – Mark Taylor**
 - Provides extensible API to handle any sort of table.
 - SED lib makes extensive use in its VOTable processing.
 - SEDStarTable implements StarTable interface. Methods in STIL are called on it to serialize data from and to VOTable TABLEDATA.
- **Java Fits Utilities Library – Tom McGlynn**
 - Provides API to read, modify, create, write FITS files.
 - SED lib uses it to read a file from stream/disk, then uses Fits methods to extract data from the Headers and Tables and transfer into SED classes.
 - Uses it to build a FITS file from SED dm objects and write to file.



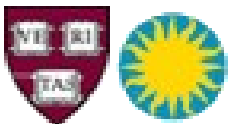
Other Tools Used...

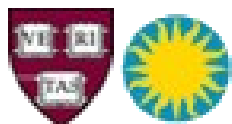
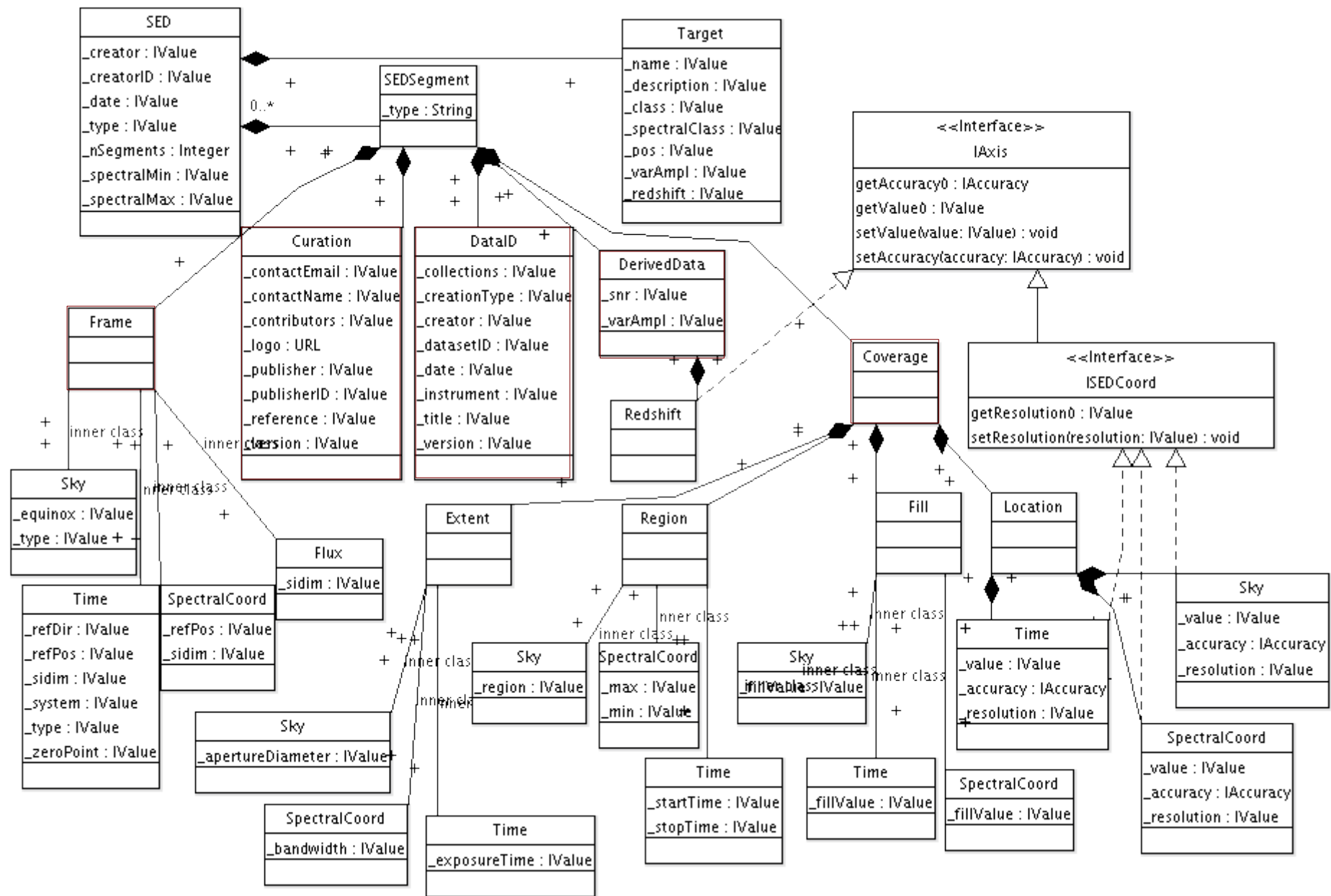
- Java Architecture for XML Binding (JAXB)
 - Java classes and interfaces are generated mirroring the elements and types in the XML schema, and marshal/unmarshal methods are provided.
 - Java code was generated from the SED schema v0.96
 - This code is used to read an XML instance file into the generated classes, whereby the library transfers the data into the SED dm classes.
 - The vice versa is also true: the generated code is used to write an XML instance file.
- Ptolemy PtPlot Library – UC Berkeley
 - Provides API for 2-D data plots and histograms
 - SED's TestPlot class implements PlotFrame to display a plot on screen
 - Not required for core library (SED Lite)



Description of Data Model Layout

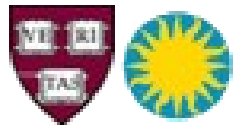
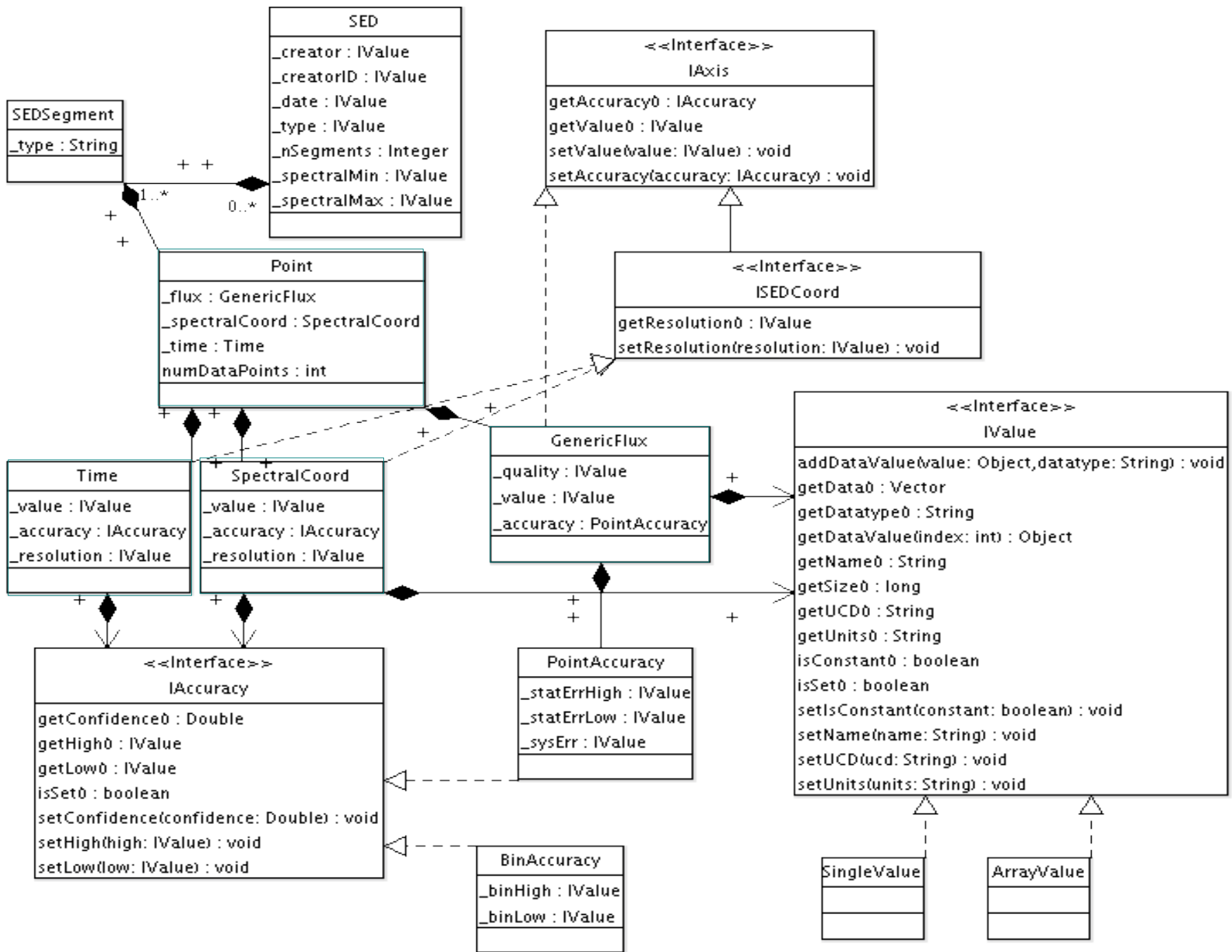
- The all-encompassing class for all data (data and metadata) is SED.
 - SED contains metadata about itself such as Target, SegmentType, and Creator.
 - Contains any number of SEDSegment objects representing spectra, time series, etc.
 - A single spectrum is equivalent to an SED with one segment.
- Each Spectrum/Segment contains a set of metadata objects and a Point object containing the data.
 - Metadata objects include Coverage, Frame, DataID, DerivedData, and Curation
 - Point contains objects representing the axes which then contain the data values and error values.





Metadata Class Diagram



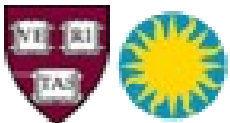


Data Class Diagram



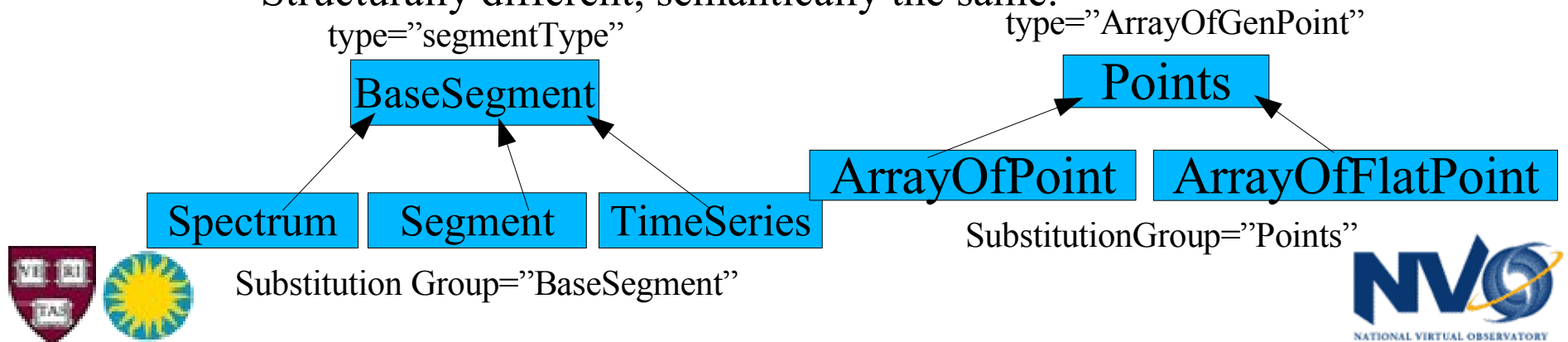
Code Design Features

- Many data elements and objects can be handled generically:
 - All axes (Sky, Time, SpectralCoord, Flux) are of type IAxis, containing accuracy and value data.
 - Also, IAccuracy, ISEDCoord
 - All discrete data values are contained in IValue instantiations; either SingleValue or ArrayValue
- Memory use is reduced by storing the data point's metadata once. eg. In ArrayValue, one set of metadata is stored along with many data values.
- Calling the library's API programmatically is simple and intuitive once the tree of data model classes is understood.



Schema Design Updates

- Allowed for base element to be either Spectrum or SED
- JAXB required customization bindings for code generation of the schema.
- Using substitution groups to provide polymorphism
 - BaseSegment group: Spectrum, Segment, and TimeSeries
 - Substitution group members have types which derive from segmentType but do not add anything..
 - Structurally the same, just adds semantic meaning.
 - Points: ArrayOfPoint and ArrayOfFlatPoint
 - Substitution group members have types deriving from arrayOfGenPoint and add elements of different types.
 - Structurally different, semantically the same.



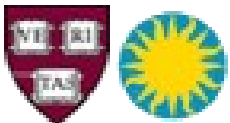
XML Instance Example

```
<?xml version="1.0" encoding="UTF-8"?>
<SED xmlns:xsd="http://www.w3.org/2001/XMLSchema" xsi:noNamespaceSchemaLocation="http://vo.cfa.harvard.edu/2
soft/SED/spec0.96.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<!-- xml instance example -->
  <Target>
    <Name>Arp 220 </Name>
    <redshift>
      <Value>0.0018</Value>
    </redshift>
    <pos>233.737917 23.503330</pos>
  </Target>
  <Spectrum>
    <SegmentType ucd="meta.code">Spectrum</SegmentType>
    <Coverage>
      <Location>
        <Sky>
          <Value ucd="pos.eq" unit="deg">132.4210 12.1232</Value>
        </Sky>
        <Time ucd="time.obs" unit="d">52148.3252</Time>
      </Location>
    </Coverage>
    <!-- MORE METADATA HERE -->
  <!-- Define table structure -->
  <Fields>
    <SpectralCoord>
      <Value ucd="em.wavelength" unit="Angstrom"/>
      <Accuracy>
        <BinLow ucd="stat.min;em.wavelength" unit="Angstrom"/>
        <BinHigh ucd="stat.max;em.wavelength" unit="Angstrom"/>
      </Accuracy>
    </SpectralCoord>
    <Flux>
      <Value ucd="phot.flux;em.wavelength" unit="erg cm**(-2) s**(-1) Angstrom**(-1)"/>
      <Accuracy>
        <StatErrLow unit="erg cm**(-2) s**(-1) Angstrom**(-1)"/>
        <StatErrHigh unit="erg cm**(-2) s**(-1) Angstrom**(-1)"/>
        <SysErr>0.05</SysErr>
      </Accuracy>
      <Quality/>
    </Flux>
  </Fields>
  <!-- Use table structure -->
  <ArrayOfPoint>
    <Point>
      <SpectralCoord>
        <Value>3200.0</Value>
        <Accuracy><BinLow>3195.0</BinLow><BinHigh>3205.0</BinHigh></Accuracy>
      </SpectralCoord>
      <Flux>
        <Value>1.38E-12</Value>
        <Accuracy><StatErrLow>5.2E-14</StatErrLow><StatErrHigh>6.2E-14</StatErrHigh></Accuracy>
        <Quality>0</Quality>
      </Flux>
    </Point>
  </ArrayOfPoint>
  <!-- OR -->
  <ArrayOfFlatPoint>
    <Point SP="3200.0" SP_BinL="3195.0" SP_BinH="3205.0" F="1.48E-12" F_ErrL="2.0E-14" F_ErrH="2.0E-14"/>
  </ArrayOfFlatPoint>
</Spectrum>
</SED>
```


Unimplemented Items

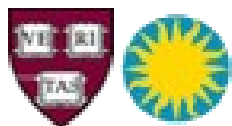
Library not done yet!

- Some metadata is ignored in some formats
- There are some issues with UTYPE handling in VOTable
- The current release (v0.4) is for evaluation only
- SAO Team will maintain the library in sync with the document upgrades.



Unimplemented Items - Details

- There is no BackgroundModel class at this time. There is no distinction in the GenericFlux class between total and next flux yet.
- Auxiliary data is not preserved across serializations.
 - When serializing a VOTable to SED, auxiliary data is noted in the log but not stored in the SED objects.
 - When serializing an XML document to SED, CustomParams are not serialized.
 - When serializing a Fits file to SED, unexpected keywords or columns (ie. keywords or columns not explicitly listed in the IVOA SED Data Model document v0.93) are not serialized.
- Upon serializing data into an SED, datatypes of some data are converted to a different type. The original datatype is not yet remembered.
- The Reader class read() methods require the caller to specify the file format of the input file. Eventually this requirement will be unnecessary.
- Serialization from SED to Fits is not complete. Certain aspects of Segment metadata are not implemented yet. They are Frame, Coverage, and DerivedData.



Unimplemented Items – Details cont...

- Serialization from Fits to SED handles items listed as keyword-only in the IVOA SED DM document v0.93 only if they appear as file keywords, or at the SED level. In actuality, these items could also appear as Columns in the Fits file (or as TTYPE). If they do, they will not be serialized.
- Serialization from XML to SED only handles ArrayOfFlatPoint format for storing data. Handling of ArrayOfPoint is coming shortly.
- The handling of utypes in a VOTable when serializing to SED is case-sensitive.
- "Inherited" utypes when serializing from VOTable to SED is not yet implemented. Utypes in an input file should be fully qualified. For example, a PARAM with utype "Segment.Coverage.Location.Sky.Value" is necessary to serialize this data. A PARAM with utype "Location.Sky.Value" appearing within a "Segment.Coverage" GROUP would not be serialized.
- When a VOTable input file contains one Spectrum table (rather than an SED containing Segments/Spectra), some metadata items are not serialized, such as Target.
- When serializing from a single Spectrum, the library will treat it as an SED containing one Segment. This means when converting a single Spectrum, it will be written out as contained within an SED, thus not preserving the initial structure. This will be updated

