

JSky

Building a repository of reusable Java Components for Astronomy

Miguel Albrecht
Allan Brighton
Francois Ochsenbein
Anuradha Koratkar
Pierre Fernique

JSky Goals

- **Open source repository**
- **Achieve standards for**
 - **documentation (JavaDoc)**
 - **Interfaces (beans, interface classes, etc.)**
 - **Data object definitions (XML)**
- **100% pure Java**
- **Foster re-usability by fostering component architecture**

Experience reports

- **The making of NGST's Expert Assistant
(Sandy Grosvenor, STScl)**
- **Using components in Observing Tool 2
(Kim Gillies, Gemini)**
- **Using components for Jipa and Starview II
(Markus Dolensky, ESO/ST-ECF)**
- **Using components for image/catalog access
and display (Allan Brighton, ESO)**

What makes a JSky component?

- **Class libraries (static instantiation)**
 - **Utility (e.g. xml parsing)**
 - **Algorithmic (e.g. WCS)**
 - **Basic packages (e.g. FITS-IO)**
- **Beans (dynamic instantiation)**
 - **Both kinds: GUI and non-GUI**
- **Apps or mega-widgets (runnables)**
- **Applets (`has init()`)**

Anatomy of a component

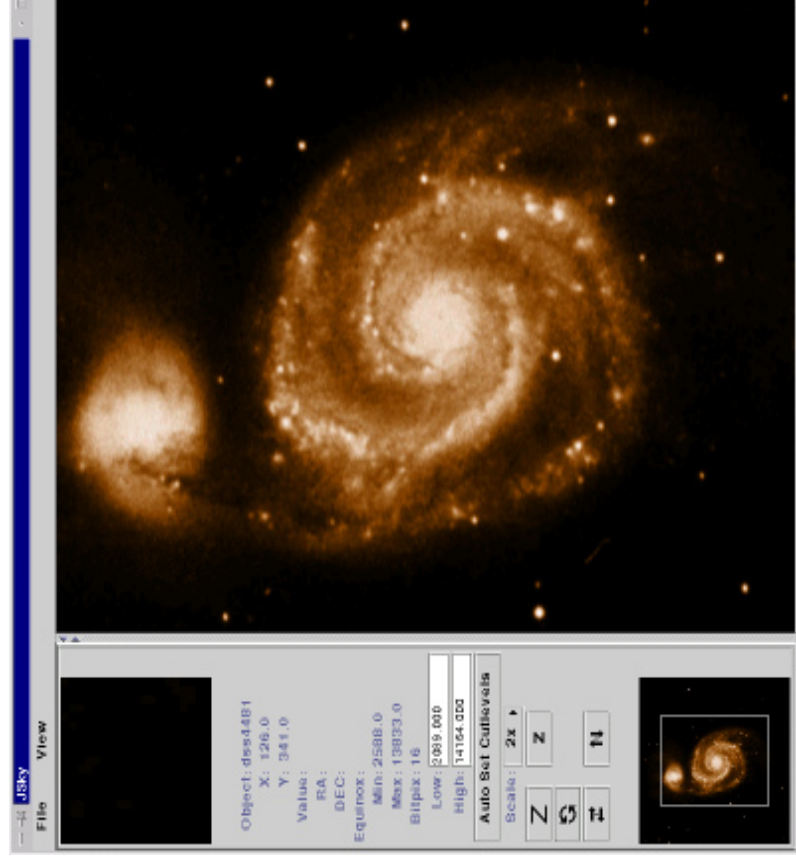
- Each component belongs to a 'package'
- Package names should point to source
- Each component has
 - index.html with description and links to /doc
 - <name>/src (sources)
 - /doc (JavaDoc)
 - /conf (properties, XML DTD's, etc.)
 - /examples if any

ESO components

- **Image display and manipulation**
(uses JAI, jdk1.2)
- **Catalog and data server support**
(uses astrores XML definitions)
- **Support for large FITS images and tables**
- **Next generation *Skycat* tool ...**

Image Widgets (First Prototype)

- Image display window
- Pan window
- Zoom window
- Cut levels widget
- Image info panel
- More planned...



First Prototype Catalog Widget

- Example shows query of an XML catalog file
- Still need to implement filtering for tabular data
- Should support URL columns with pointer to more info

J	GSC	RA(J2000)	DE(J2000)	PosErr	Pmag	e_Pmag	n_Pmag
1.3259	0543403469	123.12400	-12.37175	0.5	13.51	0.40	0
1.3261	0543403469	123.12408	-12.37174	0.4	13.54	0.40	0
1.4533	0543401417	123.10466	-12.36903	0.4	12.75	0.40	0
1.4590	0543401417	123.10458	-12.36909	0.5	12.75	0.40	0
1.7183	0543401693	123.13104	-12.32347	0.4	15.21	0.40	0
2.1673	0543402584	123.08513	-12.38202	0.4	14.75	0.40	0
2.2867	0543400959	123.15643	-12.33636	0.4	13.92	0.40	0
2.6784	0543402546	123.16501	-12.34229	0.4	14.71	0.40	0
2.7176	0543401329	123.09298	-12.38681	0.4	14.25	0.40	0
2.8238	0543401119	123.14048	-12.39260	0.5	12.17	0.40	0
2.8242	0543401119	123.14056	-12.39257	0.4	11.92	0.40	0
2.8274	0543403121	123.10760	-12.39554	0.5	12.46	0.40	0
2.8281	0543403121	123.10744	-12.39551	0.4	12.84	0.40	0
2.8326	0543403121	123.10737	-12.39557	0.5	13.00	0.40	0
3.1236	0543402059	123.13553	-12.30014	0.4	13.84	0.40	0
3.1992	0543400161	123.07045	-12.37237	0.4	12.71	0.40	0
3.2037	0543400161	123.07038	-12.37240	0.5	12.75	0.40	0

JSky repositories

At ESO

<http://archive.eso.org/Jsky>

In future also at

STScI, GSFC, NCSA, CDS, CADDC

Image Display with JAI

- **JAI - Java Advanced Imaging**
 - Large number of built-in *image processing ops*
 - Supports GIF, TIFF, JPEG, PPM *image formats*
 - *Tiling* support, for large images (no *memory mapping* in Java, but *tiles* should help)
 - *Extensible* (add your own *image formats* and *image processing operations*)
 - Version 1.0 released, but new *Image I/O package* is in the works

JAI is Format Independent

- Registry used to find code to read/write image formats
- Application need not know the image format
- FITS keywords accessed as image *properties*

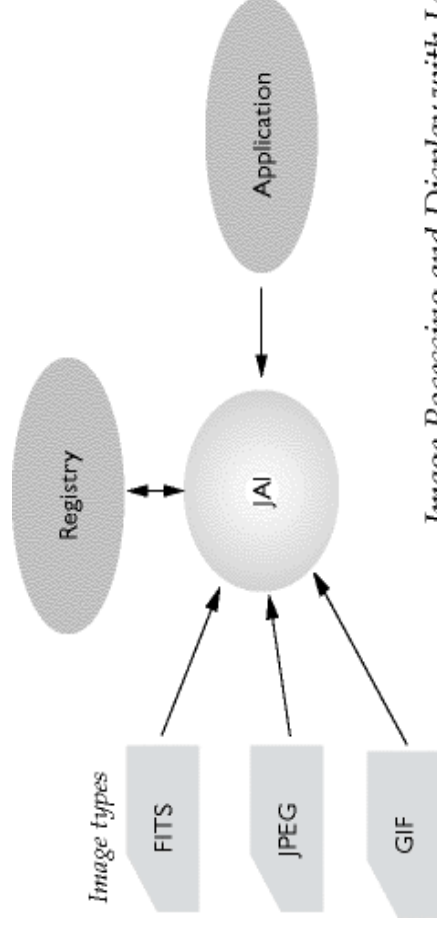


Image Processing and Display with JAI

Alternatives to JAI?

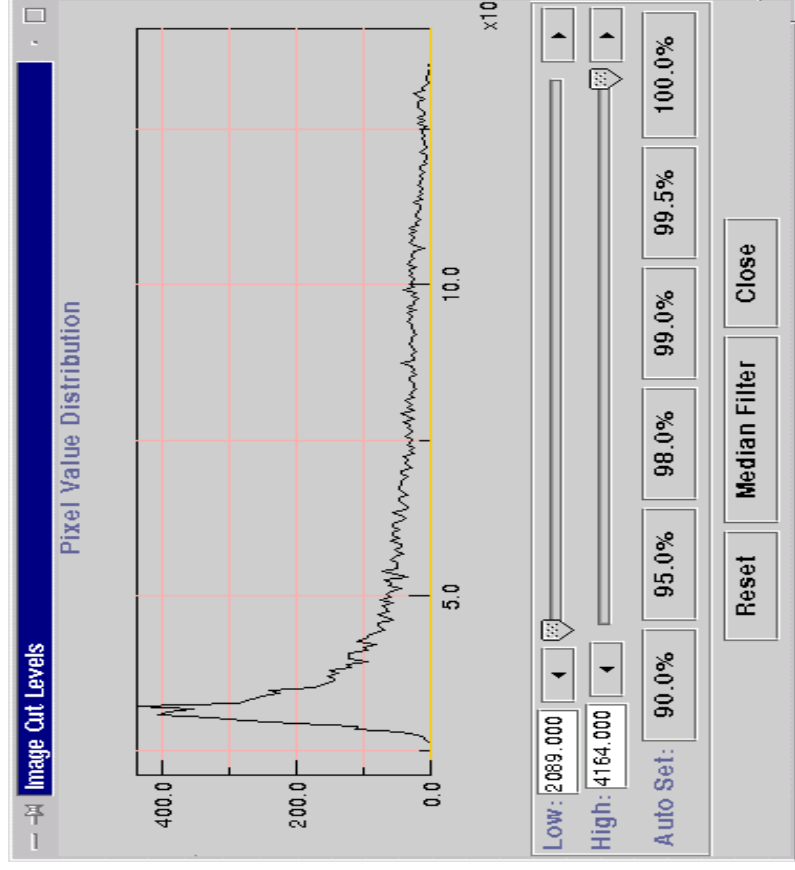
- **JIMI, Activated Intelligence (was purchased by Sun Microsystems)**
- **Java2D (entire image must be in memory)**
- **Native C/C++ code (not portable)**

Fits Support

- Based on Thomas McGlynn's FITS library, released with VisAd
<http://www.ssec.wisc.edu/~billh/visad.html>
- Supports image extensions (ASCII, binary tables)
- Used to implement FITS support for JAI
- Works well, but needs to be optimized for use with JAI to avoid unnecessary image copying

Jsky Cut Levels Widget

- Uses JAI operator to get *pixel* distribution
- Implemented median filtering (ported from SkyCat)
- Using public domain Graph package (need a better one)



Current Status

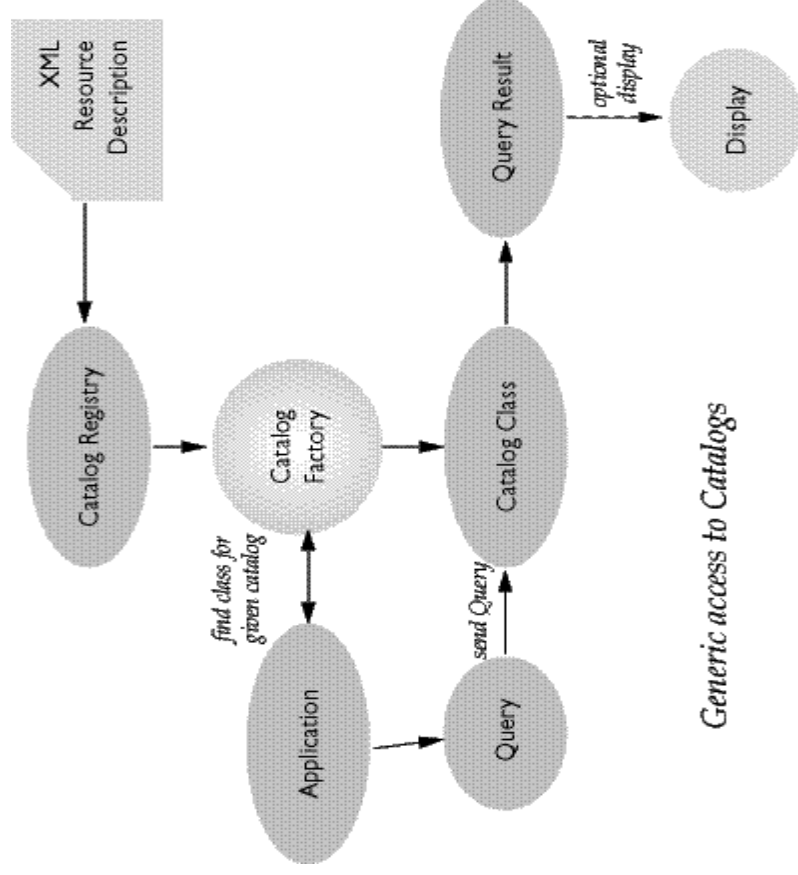
- **Prototype only (pre-pre-alpha...)**
- **FITS image handling needs to be optimized to improve speed and reduce memory usage for large images**
- **Some image data types not supported yet**
- **Image extensions not handled yet**
- **No world coordinate handling yet**

Catalog Support

- New XML catalog format being defined by CDS, ADC, ESO and others (astrores)
- Support for other catalog formats may be added
- Low level classes and interfaces for use by applications (*Catalog*, *QueryResult*, ...)
- Catalog Widgets (*CatalogQueryTool*, *CatalogQueryPanel*, ...)

Generic Catalog Access

- Applications use a Catalog Registry to access a catalog object
- No need to deal with XML or other format at application level



XML Catalog Format

- **AstroRes DTD (*everything is a resource*)**
<http://vizier.u-strasbg.fr/doc/astrores.htx>
- **Currently using Sun's XML parser (SAX)**
(Could also use higher level *DOM* parser)
- **One Java class for each XML element**
- **Classes implement the *Catalog* or *TableQueryResult* interfaces, so applications don't see actual format, only the interfaces**

Main features of AstroRes

- Proper resource information (parameter, column descriptions, units, ...)
- XML file may contain *multiple catalogs* (tables, resource descriptions)
- Supports meta-queries (e.g. cascading groups of catalogs or resources)
- Supports linking to web docs and actions

Status of Catalog Classes

- Still working on prototype...
- Table display works (*uses Swing Jtable*)
- Basic XML parser is working (*just need to settle on one XML format...*)
- Need to rethink the user interfaces to avoid Skycat's limitations, be more flexible, ...
- XML software is still *cutting edge* technology...