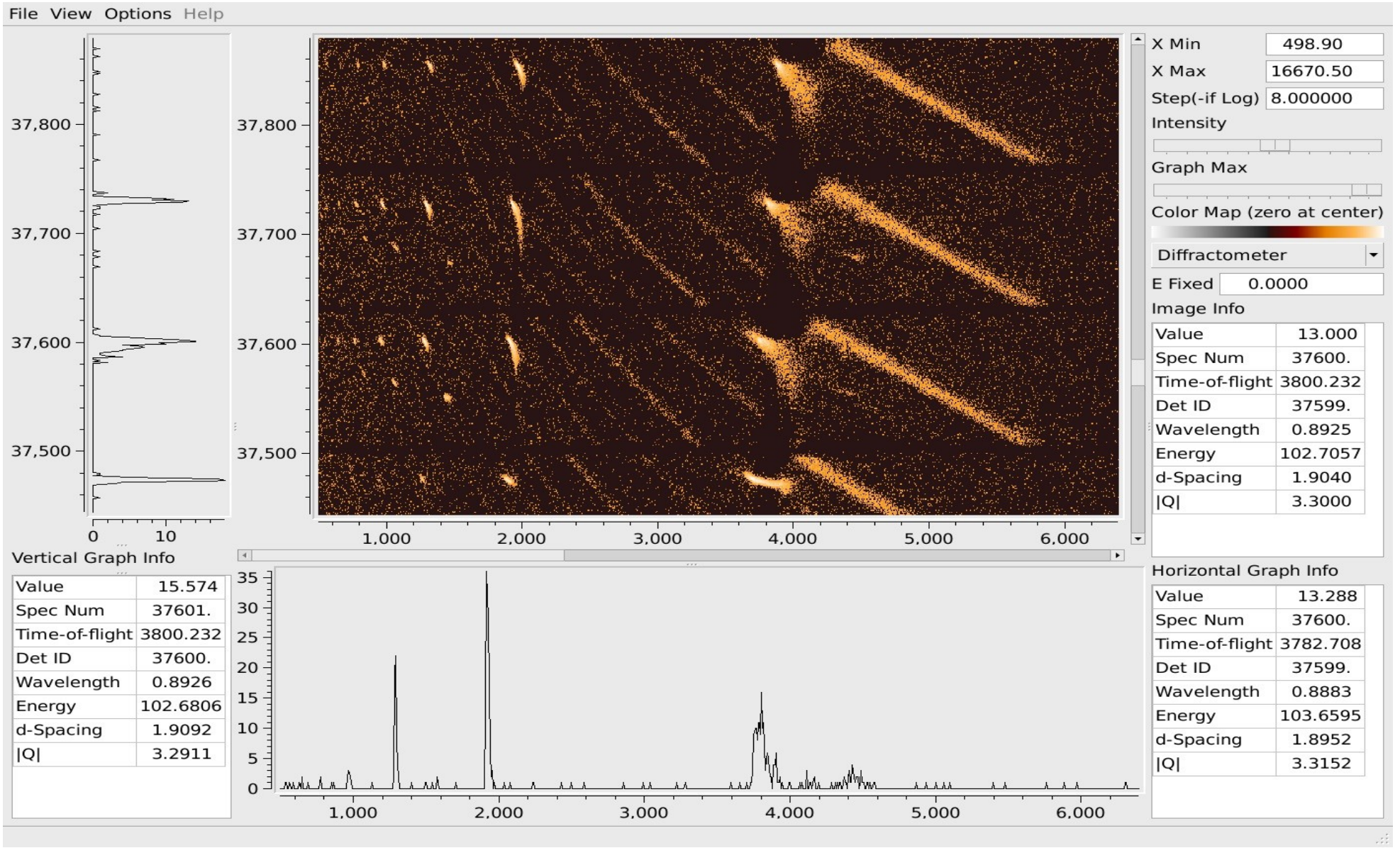
IMAGE VIEWER OVERVIEW



General Design Goals

- Provide for quick interactive examination of a list of spectra
- Display the collection of spectra as a scrollable image
- Accommodate a large collection of spectra with potentially a large number of bins
- Support selection and display of a specific spectrum, and of a vertical cut through the image display, as ordinary line graphs.

Design Goals (continued)

- Display information about the data at the cursor position
- Support quick "on-the-fly" rebinning
- Loosely coupled to Mantid workspaces (via an interface) to allow for reuse

Possible Uses

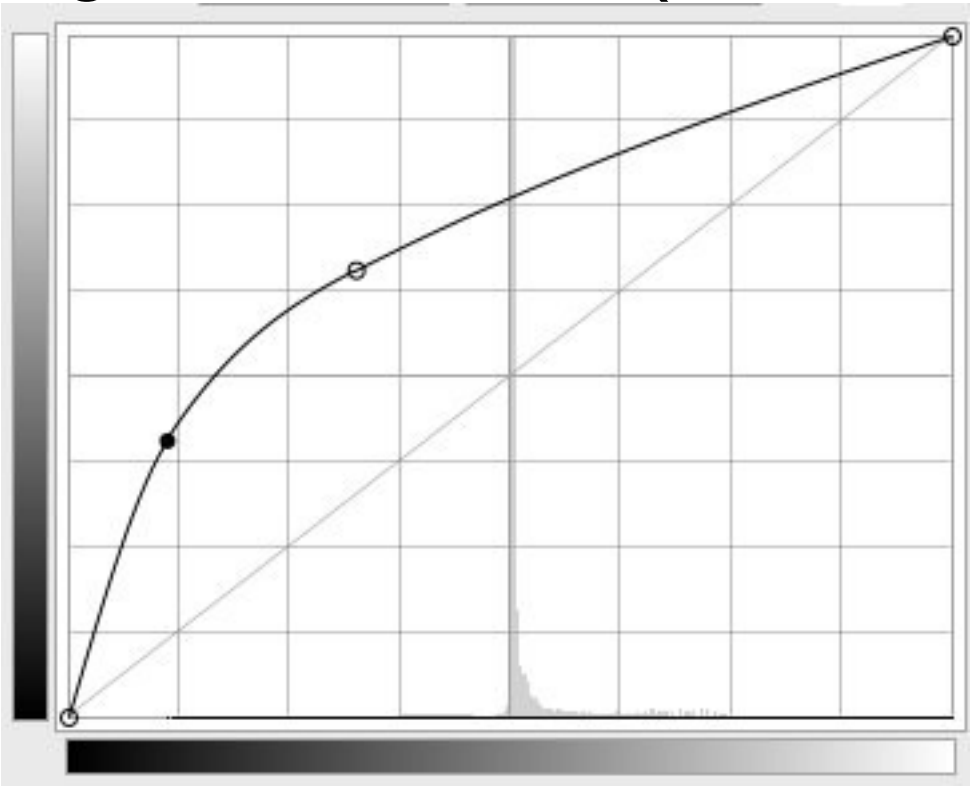
- Quick look at raw data, either event or histogrammed
- Find anomalies in data (detector problems or?)
- Display processed data such as collections of spectra corresponding to temperature scans or other parametric studies

Specific Features

- Supports "Instant rebinning"
- Scrollable image, with scroll bars depending on binning
- Scrolling image also scrolls graph(s)
- Readouts of data about pointed at locations on image and on each graph
- Provides a true cross-hair cursor extending the full width and height
- Works on MatrixWorkspaces (EventLists or Histograms)

Specific Features (continued)

- Quick selection of color scales, including all Mantid color scales
- "Pseudo-Log" color scales (like curves in GIMP)

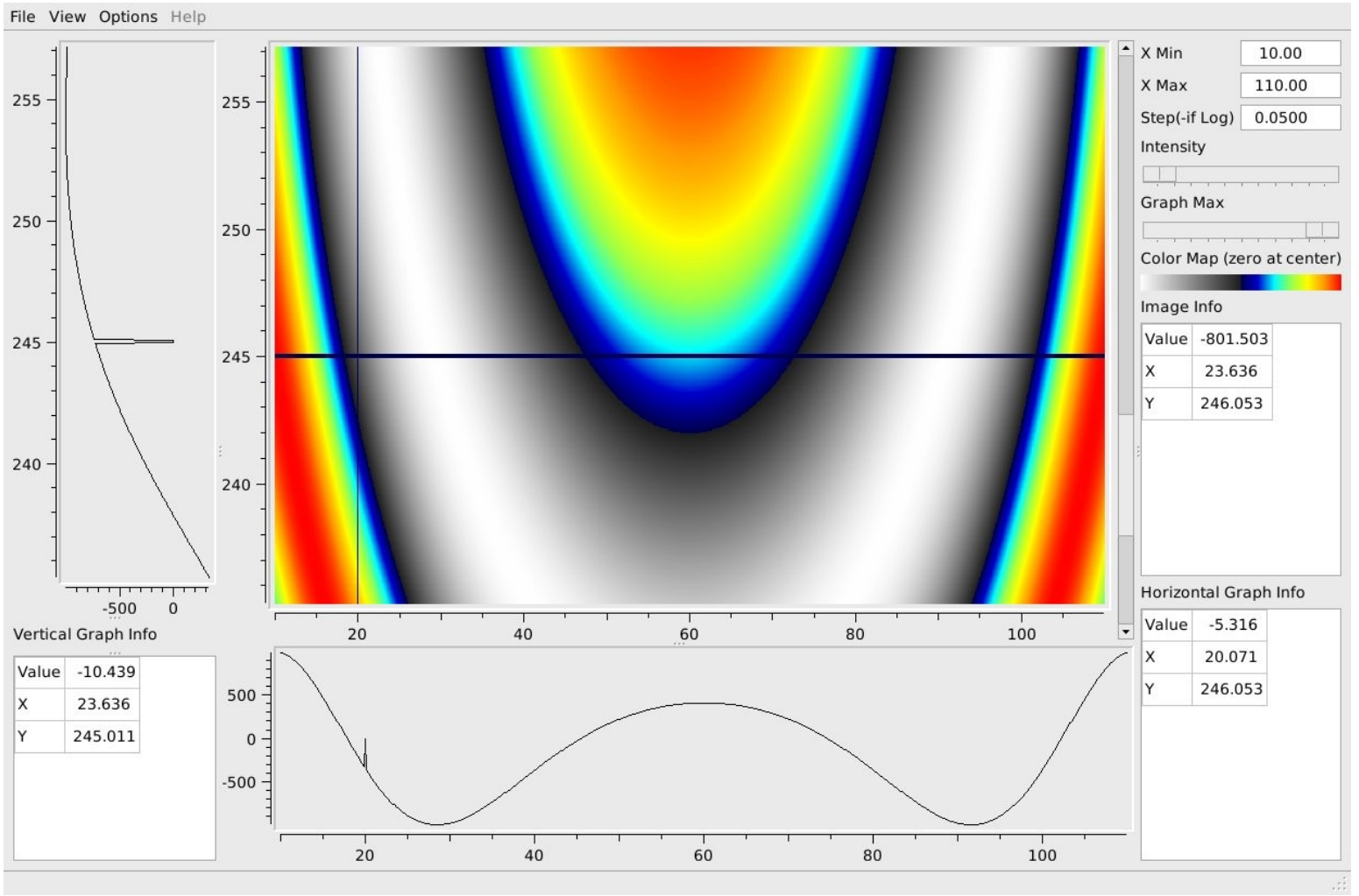


Specific Features (continued)

- "Two-sided" color scales to support negative intensities
- Graphs can be collapsed to zero size to provide larger image display

Examples

- ImageViewDemo (make ImageViewDemo)



Examples (continued)

- GEM38370_Focussed_Legacy (Log scale)
- CNCS_7860_event (noise in Det ID 38011?)
- ARCS_17856_event.nxs (dead time in Det ID 29400 ?)
- SXD23767 (Det ID 2724, thermal diffuse scattering, Det ID 15806, ??)
- WISH00017986 (pulse at 100,000 us, peak shape Det ID 59070)

“Oddities”

- Asymmetric treatment of rows and columns
- A row is obtained by selecting one spectrum from list of available spectra
- All displayed spectra are rebinned to a common x-scale and the binned values for one x-scale bin form a column of the image
- Graphs of slices are just cuts through the resampled data image hence, rebinned in X direction but subsampled in Y direction

“Oddities” (continued)

- With a large number of spectra, if vertical scrolling is turned off, the display can be mis-leading
- Data WILL always be rebinned according to the current setting on the binning controls
- You can't show arbitrary data binning, just log or linear
- Image requires columns of values =>problems with axes and problems with different binnings for different spectra

Reuse

- ImageView code is loosely coupled to Mantid workspaces, to allow for use in other ways in Mantid
- The basic interface concept is the ImageDataSource interface, which has two key methods
- GetDataArray(xmin, xmax, ymin, ymax, n_rows, n_cols, log_x_flag)
- GetInfoList(x, y)

Reuse (continued)

- Two ways to use this for other purposes:
- First, use or Extend the ArrayDataSource. This just applies the ImageViewer to view a 2-D array. Eg. See ImageViewDemo.cpp
- float * data = MakeTestData(...);
- ArrayDataSource* source = new ArrayDataSource(..., data);
- ImageView image_view(source);

Reuse (continued)

- Second, make a new class that extends ImageDataSource
- Eg. See MatrixWSDataSource class

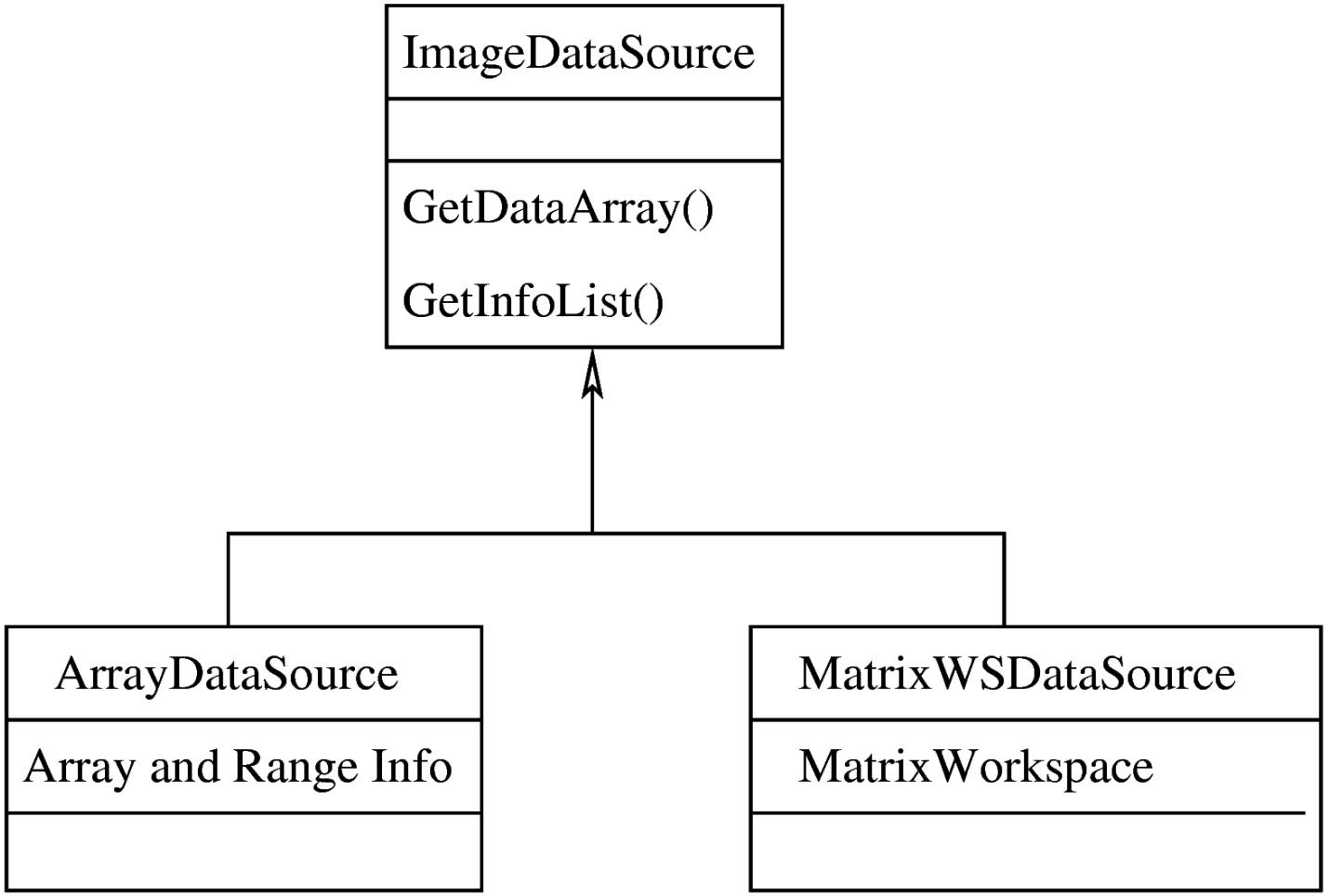
Overall Structure

- All code in Code/Mantid/MantidQt/ImageViewer
- Interface was made using Qt designer
- Three key classes:
- DataArray: Simple immutable class used to pass 2D array of values and information about the extent of the array from a data source to the image display

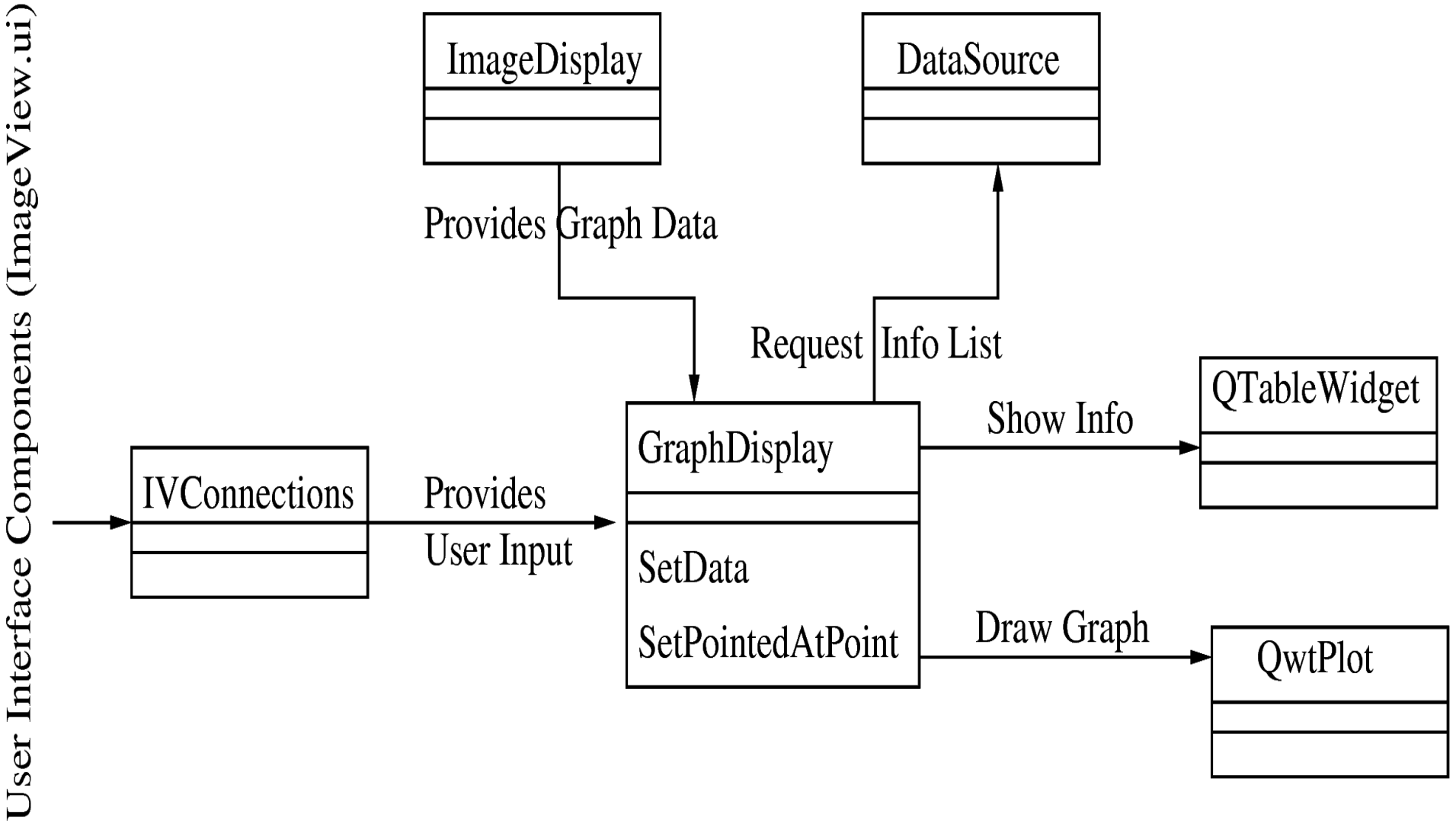
Overall Structure (continued)

- ImageDataSource: Provides the image DataArray corresponding to a requested data region, and a list of relevant information (name,value pairs) about a point in the data region
- ImageDisplay: Displays the data as an image and uses a GraphDisplay object to display the currently selected spectrum and a second GraphDisplay to display the currently selected vertical cut

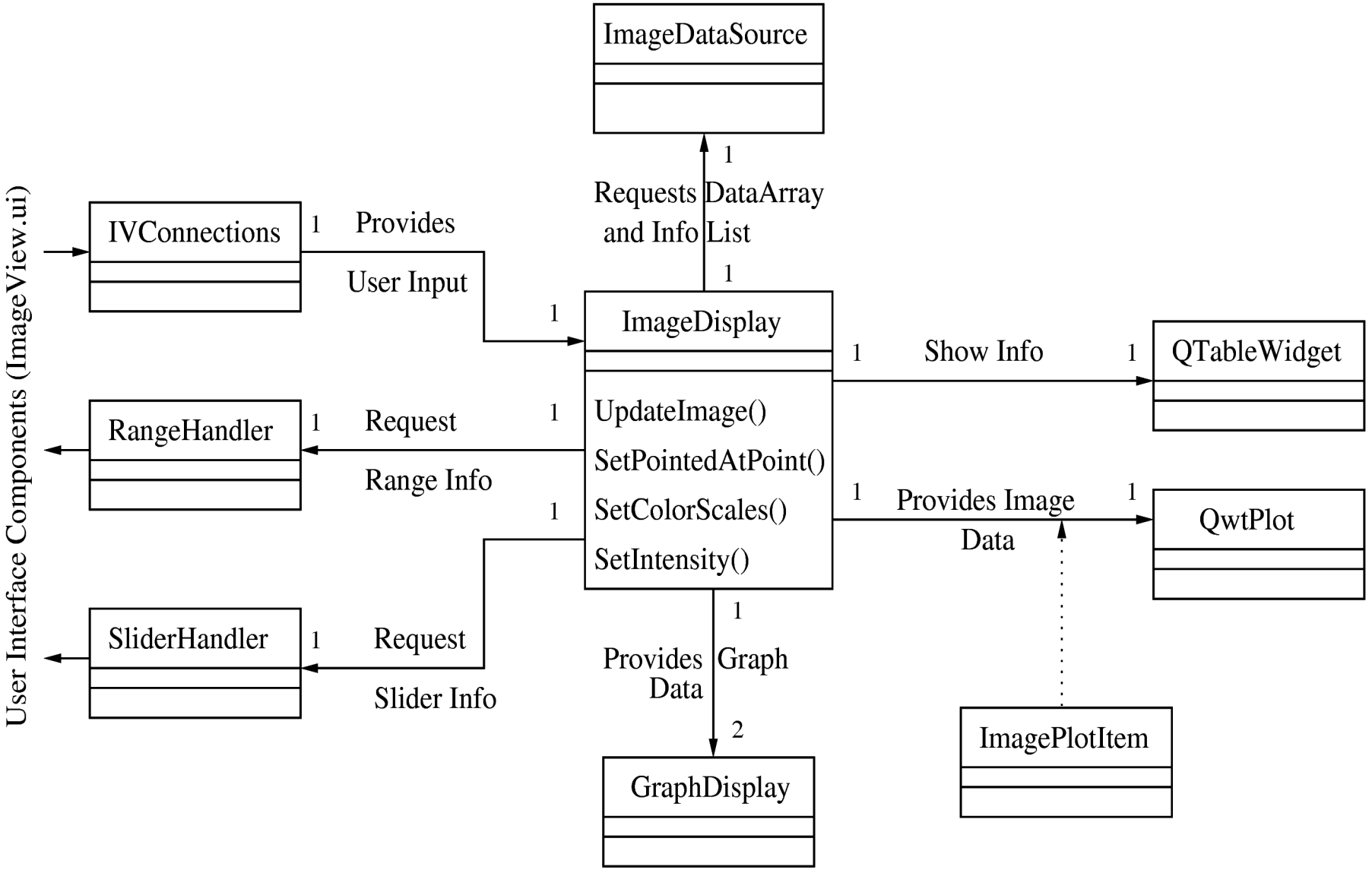
Image Data Sources



GraphDisplay



ImageDisplay



Summary of Classes

- MatrixWSImageView

  Top level class. This just constructs a MatrixWSDataSource and an ImageView object to display the data.

- ImageViewDemo

  Demo application. This just constructs a demo ArrayDataSource and an ImageView object to display the data.

  'make ImageViewDemo' to build it

Summary of Classes (2)

- ImageView

  High level class. Given an ImageDataSource this constructs and configures the various objects that form the Image View.

Summary of Classes (3)

- ImageDataSource

  Abstraction for domain-specific rebinned image and information. Provides an image array over a specified "real" region at a specified resolution:

- GetDataArray(xmin, xmax, ymin, ymax, n_rows, n_cols, is_log_x)
- GetInfoList(x, y, &list)

Summary of Classes (4)

- ArrayDataSource

  Simple case of an ImageDataSource, wrapped around an ordinary 2D array.

- MatrixWSDataSource

  An ImageDataSource that gets the data array and information list from a MatrixWorkspace.

- DataArray

  Simple array with range and resolution information that is passed from an ImageDataSource to the ImageDisplay.

Summary of Classes (5)

- ImageDisplay

  Key class that uses and coordinates many of the other classes.

  Uses an ImageDataSource, gets control values and passes specified subimage, cuts across image and information about the pointed at location to other objects for display.

- ImagePlotItem

  Displays current DataArray

  Extends QwtPlotItem to support two-sided color scale and pseudo-log intensity map.

Summary of Classes (6)

- GraphDisplay

  Gets (x,y) data for graph of a row or column slice from the ImageDisplay.

  Sends the graph data and information about the pointed at location to other objects for display.

Summary of Classes (7)

- IVConnections

  “Glue” code that passes changes in GUI controls to display objects. Has “slots” that are called when the controls are adjusted. These “slots” in turn call methods on ImageDisplay and GraphDisplay to carry out the appropriate changes.

Summary of Classes (7)

- RangeHandler and SliderHandler

  These classes provide information about the GUI state to the ImageDisplay and GraphDisplays, when requested. Eg. When the image is to be redrawn, the range and slider handlers proved information about the current sub-range of data, so the image DataArray can be generated and displayed.

- TrackingPicker

  Emits a mouseMoved signal when the mouse is moved over a specific canvas region. Extends QwtPlotPicker by calling SetPointedAtPoint() on the Graph or Image display.

Summary of Classes (8)

- EmodeHandler

  Emode and Efixed should come from the MatrixWorkspace, however, the required values may not always be present, or may not be correct. The EmodeHandler gets Emode and Efixed from the GUI and/or set values from the MatrixWorkspace into the GUI.

Summary of Classes (9)

- ColorMaps

  Utility class with methods to generate basic color maps by interpolating between specified key colors, and to form pseudo-log intensity map.

- QtUtils

  A few basic utilities for interfacing with Qt

- IVUtils

  A few convenience functions for ImageView