Elements of The Chandra Data Analysis System

Michael S. Noble

Harvard-Smithsonian Center For Astrophysics, Cambridge, MA 02138

Abstract. The Chandra X-Ray Observatory was launched on July 23, 1999, with the first public release of the data analysis system (CIAO: Chandra Interactive Analysis of Observations) made availabile on October 4, 1999. This paper presents several of the more novel components of CIAO, including FirstLook, FilterWindow, ChIPS, Prism, and ToolAgent. The functionality is discussed in the context of two issues plaguing modern software – increasing complexity and insufficient reuse – and what approaches were taken with CIAO to mitigate the situation. Chief among these are intuitive GUIs, a thoughtful commingling of both GUI and command line tools, and the development of homogeneous interfaces to libraries and applications already in wide use.

1. Introduction

The emergence of the electronic computer has inextricably linked the progress of science with the march of technology, a phenomenon from which astronomy is hardly exempt. For legions of astronomers, the "scientist's labcoat" was long ago exchanged for a sophisticated workstation loaded with powerful analysis software.

In this arrangement, then, it is imperative that technology provide not only raw function, but do so in as convenient a fashion as possible, because by reducing the thought required to use software we enable the astronomer to spend more time thinking about science. It is in this spirit that much of CIAO was developed, with concerted efforts having been made throughout the system to shield users from the inexorable effects of software entropy, as well as leverage their knowledge of existing software by reusing popular components wherever possible.

For example, rather than reinvent yet another visualization system, we designed ChIPS (Chandra Imaging and Plotting System) to homogenize access to potentially any plotting or imaging package. Likewise, the DataModel was developed so that tools and applications would be free to concentrate on more abstract and algorithmic issues, released from concerns specific to underlying file formats. Where appropriate and sensible we developed new GUIs (like firstlook and the filter window) to simplify and formalize common analysis operations, and have achieved a new level of integration between both the GUIs themselves and the command line tools that form the core analysis functionality.

Since space is limited snapshots of the GUIs are omitted here, in favor of referring the interested reader to the CIAO documentation available online at chandra.harvard.edu.

2. FirstLook

FirstLook is envisioned as the starting point for observational analysis, offering an intuitive and efficient means of broadly assessing the data products of Chandra observations.

When launched firstlook scans the user environment for observation index files (OIFs). These OIFs are created during archive extraction and describe the structure and content of the dataset in conformance with the draft FITS grouping standard (Jennings, et al 1999). For each OIF firstlook initially displays the observation ID, instrument and target names, exposure time, and target coordinates, but the window can be configured to omit or display as much detail as desired.

A toolbar provides the means to quickly generate the primary image, spectrum, or lightcurve for each OIF, with configurable popup help attached to each button to indicate its function. Alternatively, clicking upon either the magnifying glass or any OIF entry displays the complete observation fileset in hierarchical form, at which point prism may be launched to browse file internals via another selection-click. During any of these steps the user has the option of automatically launching the filter window and continuing a bona fide analysis session.

The speed and ease with which one can thus become acquainted with one's data contrasts very favorably with missions past, where observers were given media containing scores of files through which they could spend considerable time fumbling for the relevant handful of files with which to begin their work.

3. Filter Window

The filtwin application is envisioned as one of the focal points of GUI analysis sessions. As data are massaged via filtering, binning, region selection, plotting, and fitting, filtwin iteratively notes the various applied filters, in essence recording "the state of your data". Control flow is bi-directional, in that filtwin can effect changes to the plotter and imager, for example, or the plotter and imager can effect changes to filtwin.

A typical thread might begin by generating a primary image from within firstlook, which in turn launches filtwin to record that the imager is displaying a binned event list with a given blocking factor. After isolating the image source by laying region markers, the filtwin display would be updated to reflect the marked regions, and said information could then be passed back to firstlook to generate more focused spectra or lightcurves.

Recorded filters and data may also be saved to disk for later use with command line tools, and early in Y2K we envision adding a powerful "bookmarking" mechansim through which one will able to save/restore entire analysis sessions by simply invoking filtwin on saved state files.

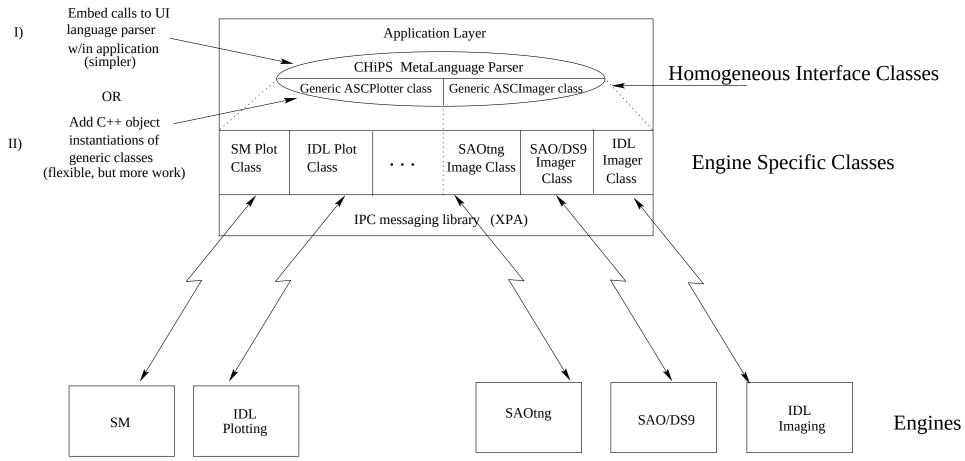

Figure 1. ChIPS C++ class architecture

4. ChIPS

As mentioned earlier, within CIAO the approach taken for data visualization was to avoid inventing yet another plotting and imaging package, in favor of developing generic interfaces that permit the reuse of existing software like SAOtng and SM. Not only does this save development time and trim costs, but it also diminishes the need for user retraining by leveraging their familiarity with popular interfaces.

From a developer perspective there are two means of accessing ChIPS, either by directly instantiating generic imager or plotter objects and calling their public methods, or by embedding within the application calls to the ChIPS parser and letting it instantiate and manage the C++ objects internally. The latter method is simple to utilize, while the former provides far greater flexibility and power.

Note that in both cases explicit knowledge of which imaging or plotting engine is actually satisfying the visualization requests is hidden from the high-level application code, ensuring that virtually zero rewrite cost would be incurred should, for example, one supported imager be switched out in favor of another. End users may customize which engines are utilized by editing .CXCdefaults, an X resources file created at CIAO installation time.

From a user perspective, ChIPS offers a "metalanguage" interface which homogenizes operations commonly found within most visualization software. For example, while IDL and SM are undeniably powerful packages, both suffer from arcane and often cryptic syntax. In contrast, the ChIPS language utilizes a consistent and modern "object-like" syntax, which simplifies many operations and provides several useful extensions like multi-level undo and redo, and sophisticated display area manipulations. Should the user prefer the original command language, however, a "native" mode is supported wherein command input is passed directly through to the underlying engine, sans interpretation by the ChIPS parser.

The ChIPS design has been utilized successfully within sherpa (the CIAO fitting and modeling application), the prism file browser, the filter window, and the chips parser itself. Engine-specific interfaces currently exist for the SM plotting package, and both the SAOtng and SAO/DS9 imagers (all three of which are bundled within CIAO), with an IDL interface planned for Y2K.

5. Prism

Prism is a format-independent file browser, which in terms of performance, footprint, ease of use, breadth of supported filetypes, and tool integration provides an attractive alternative to existing file viewers.

Rather than requiring the user to navigate intricate collections of windows, prism presents the most crucial information within one main window, subdivided into scrolling regions that display extensions, header keywords, and data, with hooks provided to generate quicklook ChIPS plots and images by simply selecting the desired column(s) and/or row(s) in the data matrix.

Utilizing the CXC DataModel for I/O, prism currently reads FITS images and bintables, as well as IRAF QPOE and IMH files, with support for ASCII files and potentially STSDAS tables planned for some time in Y2K. An XbaeMatrix-based table provides a scrolling "view" onto the entire dataset. This, plus the fact that data are read in their native format, gives consistent read performance across the range of supported filetypes, even as files scale to huge sizes of 300 Mb or more.

Complex column structures are supported naturally, and in most cases the contents of arbitrarily large arrays can be expanded for viewing in only seconds. Plans for future development include adding full editing capability, statistical and arithmetic functions, traversal of multiple file lists, and image sweetspot detection.

6. ToolAgent

The toolagent is the primary means by which the CIAO GUIs interact with command line analysis tools. As a C++ class toolagent reuses the same XbaeMatrix widget as prism, and by virtue of the CIAO application framework is embeddeded within the Analysis menu of each X/Motif GUI.

ToolAgent currently provides a parameter file editor and the capability to launch and/or terminate potentially any CIAO tool, while early Y2K development plans include feeding toolagent output back into filtwin, and providing a means for users to customize the tool list to taste.

Acknowledgments. The work described herein was supported by the Chandra X-Ray Center under NASA contract NAS8-39073, with design and development contributions from Antonella Fruscione, James Balson, Thomas Deckert, Wai Min Lau, Warren McLaughlin, and James Overly.

References

D. Jennings, W. Pence, M. Folk, B. Schlesinger, 1997, http://adfwww.gsfc.nasa.gov/other/convert/group.html