CIAO: Chandra's data analysis system

Antonella Fruscione[a], Jonathan C. McDowell[a], Glenn E. Allen[b], Nancy S. Brickhouse[a], Douglas J. Burke[a], John E. Davis[b], Nick Durham[a], Martin Elvis[a], Elizabeth C. Galle[a], Daniel E. Harris[a], David P. Huenemoerder[b], John C. Houck[b], Bish Ishibashi[b], Margarita Karovska[a], Fabrizio Nicastro[a], Michael S. Noble[b], Michael A. Nowak[b], Frank A. Primini[a], Aneta Siemiginowska[a], Randall K. Smith[c] and Michael Wise[d]

[a]Smithsonian Astrophysical Observatory, 60 Garden Street, Cambridge, MA 02138, USA
[b]MIT Kavli Institute for Astrophysics and Space Research, 77 Massachusetts Avenue, 37-287 Cambridge, MA 02139, USA
[c]NASA's GSFC, LHEA Code 662, Greenbelt, MD 20771, USA
[d]ASTRON, Oude Hoogeveensedijk 4, Dwingeloo, 7991 PD, Netherlands

ABSTRACT

The `CIAO` (*Chandra* Interactive Analysis of Observations) software package was first released in 1999 following the launch of the *Chandra* X-ray Observatory and is used by astronomers across the world to analyze *Chandra* data as well as data from other telescopes. From the earliest design discussions, `CIAO` was planned as a general-purpose scientific data analysis system optimized for X-ray astronomy, and consists mainly of command line tools (allowing easy pipelining and scripting) with a parameter-based interface layered on a flexible data manipulation I/O library. The same code is used for the standard *Chandra* archive pipeline, allowing users to recalibrate their data in a consistent way.

We will discuss the lessons learned from the first six years of the software's evolution. Our initial approach to documentation evolved to concentrate on recipe-based "threads" which have proved very successful. A multi-dimensional abstract approach to data analysis has allowed new capabilities to be added while retaining existing interfaces. A key requirement for our community was interoperability with other data analysis systems, leading us to adopt standard file formats and an architecture which was as robust as possible to the input of foreign data files, as well as re-using a number of external libraries. We support users who are comfortable with coding themselves via a flexible user scripting paradigm, while the availability of tightly constrained pipeline programs are of benefit to less computationally-advanced users. As with other analysis systems, we have found that infrastructure maintenance and re-engineering is a necessary and significant ongoing effort and needs to be planned in to any long-lived astronomy software.

Keywords: data analysis, software, X-ray astronomy, modeling and fitting

1. INTRODUCTION

`CIAO`* (*Chandra* Interactive Analysis of Observations) is the name for the suite of tools and applications developed at the *Chandra* X-Ray Center (CXC) for the analysis of data from the *Chandra* X-Ray Observatory and other missions. `CIAO` 1.0 was released for the first time publicly in October 1999 and since then the software has evolved through several releases until the most current 3.3 version released in December 2005.

`CIAO` is a data analysis system which was originally written for the needs of users of the *Chandra* X-ray Observatory which was the first mission with high resolution 4-dimensional data (2 spatial, time, energy): in *Chandra* data each of these dimensions has many independent resolution elements. Therefore `CIAO` was built to handle N-dimensional data without concern about which particular axes were being analyzed. Apart from a few *Chandra*-specific instrument tools, `CIAO` is mission independent by design and has proven itself useful for the analysis of data from other X-ray and non-X-ray missions.

Further author information: AF: E-mail: afruscione@cfa.harvard.edu; JCM: jcm@cfa.harvard.edu

*For the linguists among us, the Italian word "ciao" comes from an old expression in the Venetian language, "s'ciavo", that means "I am your servant"

While the legacy from previous data analysis system (e.g. IRAF[1], FTOOLS[2], MIDAS[3]) is visible in several facets of the CIAO design, CIAO was architectured around a few innovative aspects:

- all axes are treated equally in the analysis, with no built-in assumptions toward space, time, spectrum or any other variable
- the user has the choice of a command line or a graphic-oriented approach in some tools and applications
- the syntax is uniform across the entire CIAO system
- the use of a common flexible data manipulation input/output layer called the "CXC Data Model" (DM).[4] results in:
  - file-format independence
  - row and column filtering and binning can be applied to the input file for **all** CIAO tools and applications, avoiding the need to generate subsidiary data files
  - all filters are recorded in the "data subspace" of output files which is then recognized in the subsequent analysis

The CIAO design allows close interconnection of tools. For example, the output of many tools can become the input of a subsequent tool or application. Also, the same CIAO tools are used in the *Chandra* Standard Data Processing (SDP) pipeline.[5] Users can easily and efficiently recalibrate their data if needed and they have an understandable access to the processing steps.

The CIAO software is available for a number of different platforms. The supported platforms have evolved during the years: currently it is offered for Solaris, Mac OS X and several Linux flavors. The source code for the entire software system is distributed along with the binary files, through a GNU General Public License (GPL).

CIAO can be downloaded from `http://cxc.harvard.edu/ciao/download/`.

2. TOOLS, FILES, AND ANALYSIS APPLICATIONS

2.1. Data Analysis Tools

CIAO is a collection of programs both "atomic", the so-called tools, or more complex "applications". The main CIAO applications are described in paragraphs below. CIAO offers users a wide and ever growing variety of tools for *Chandra* instrument-specific analysis, but also tools which are useful for generic data analysis tasks. There are more than 100 tools currently available in CIAO which can be divided into six broad categories:

1. **Data Manipulation**: tools needed for generic data analysis tasks such as filtering or binning, smoothing, image calculation, histogram extraction, data statistics, etc. Example of tools in this category include `dmcopy, dmextract, dmlist, dmstat`. As the names suggest, these tools rely heavily on the common DM library.
2. **Data Preparation**: tools needed to "reprocess" (recalibrate) the data processed by the *Chandra* pipeline to account for improvement in instrument calibration or special processing needs. Tools in this category include most notably the complex tools `acis_process_events` and `hrc_process_events` which play a crucial role in the pipeline and compute or re-compute, for example, coordinates and energy levels from raw instrumental data.
3. **Imaging**: tools optimized for the needs of imaging data analysis; they include tools for source detection (three different source detection methods are available for users: `celldetect, wavedetect, vtpdetect`), image smoothing (`aconvolve, csmooth`), for creating exposure maps and normalized images, 2D PSF models, and for calculating image statistics. They also include tools to create color images for publication as well as merging data from multiple observations.

4. **Imaging Spectroscopy**: tools to extract source and background spectra from imaging observations (`dmextract, specextract`).

5. **Grating Spectroscopy**: tools to detect and extract source and background grating spectra from *Chandra* grating observations (e.g. `tgdetect, tg_resolve_events, tgextract`)

6. **Timing Analysis**: tools to extract lightcurves (`dmextract`) and perform timing corrections (`axbary`)

7. **Response Tools:** tools to compute effective areas, exposure maps, energy redistribution, and line profiles for all instruments or any sub-component, including effects of dither, detector geometry, software masks and modes, and long-term effects, tailored to individual observational configurations so that the counts spectra may be modeled and fit (`mkarf, mkrmf, mkacisrmf, mkgrmf, mkgarf, mkinstmap`). The responses have also been unified through a formal underpinning on a theoretical framework.[6]

All the tools share a common infrastructure so that both the "look-and-feel" and the behavior of the tools is uniform and predictable across the system.

All parameters needed by the tools to run can be set on the command line or, as with `IRAF` and `FTOOLS`, using parameter files.[1] These are ASCII files stored in a users directory and can be easily accessible using a number of tools provided to read and write to them (e.g. `plist, pset, punlearn`) or through the GUI parameter editor `peg` (see below). Parameter files provide a simple history mechanism and allow the pre-setting of variables that rarely change.

The following are examples of setting parameters for the `dmlist` tool (1) when running the tool at the command line (2) through a `pset` statement or (3) allowing the tool to prompt for them:

```
unix% dmlist infile=pcad_asol1.fits opt=cols

unix% pset dmlist infile=pcad_asol1.fits opt=cols
unix% dmlist

unix% dmlist
Input dataset/block specification (): pcad_asol1.fits
Option (): cols
```

The Resulting `dmlist` parameter file is the same in all cases:

```
unix% plist dmlist

Parameters for /home/username/cxcds_param/dmlist.par

        infile = pcad_asol1.fits Input dataset/block specification
           opt = cols             Option
      (outfile = )                Output file (optional)
         (rows = )                Range of table rows to print (min:max)
        (cells = )                Range of array indices to print (min:max)
      (verbose = 0)               Debug Level(0-5)
         (mode = ql)
```

All tools make use of the DM library which provides users with powerful built-in data filtering and binning capability through the use of "virtual files". Virtual files (data files which are filtered and binned only in computer memory and not saved onto disk) are accepted as input by all the `CIAO` tools and applications which use the DM library making for a very powerful and uniform syntax.

The following is an example of the "virtual file" syntax. In this case only a subset of the data from the file evt2.fits is listed: the subset is defined by filters in time and coordinates. In addition, only a few columns from the original file are selected.

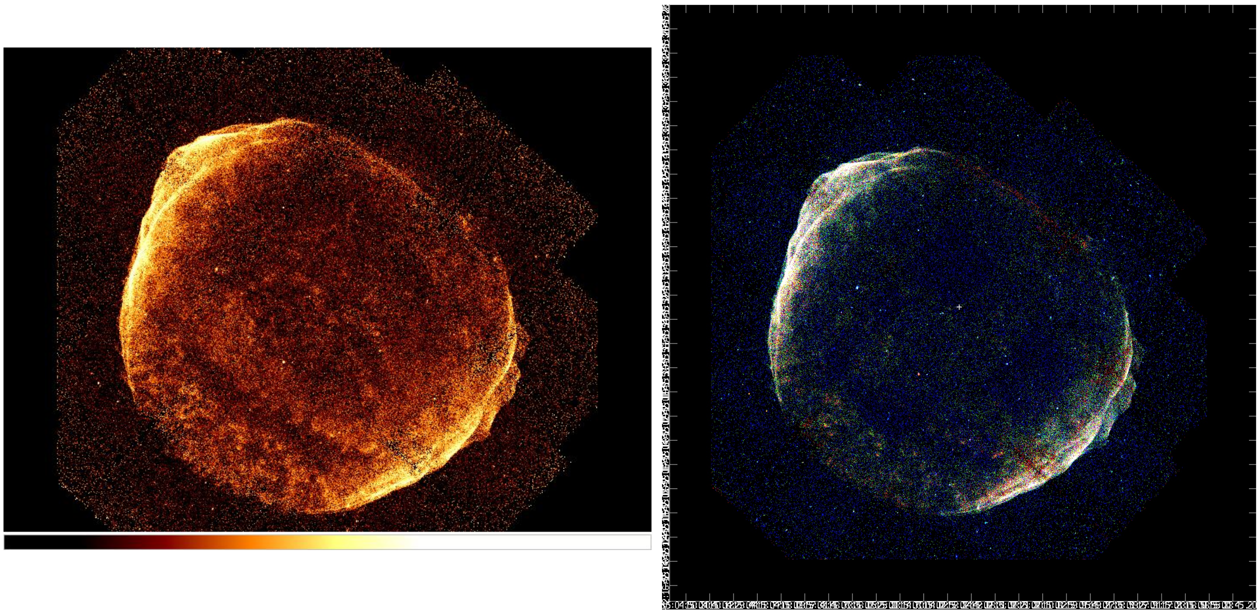

Figure 1. An exposure-corrected composite mosaic image from 11 separate *Chandra* ACIS observations of the supernova remnant SN1006. In the right panel the image is displayed in "true" colors (where color correlates with energy) obtained from the ACIS energy bands. Blue represents non-thermal emission above 2.5 keV while red represents thermal emission below 1.5 keV. Several `CIAO` tools were used to create these images and in particular the tools `reproject_image_grid` and `reproject_image` which allow a user to project images to common coordinate systems and `aconvolve` and `dmimg2jpg` used to create the smoothed three-color image.

```
unix% dmlist "evt2.fits[time=107:208][sky=ellipse(40,40,20,10,2)][cols time,det,pha]" data
```

One of the most difficult non-algorithmic problems to solve in complex data analysis is how to efficiently and easily propagate the information which is not the data itself, but which is related to the data or to the steps performed to obtain the data. For example it is very important to be able to record and maintain, along with the data, the extraction region for a source, the time filtering performed on a dataset, the version of the software used to analyze the data, the calibration files used, etc. Given the complexity of the *Chandra* data and the ample variety of tasks that can be performed on them, a complete metadata propagation has revealed itself as a very resource- and time-consuming undertaking but crucial for ease of use and traceability.

The `CIAO` software tackles this problem through the use of file "headers" and data "subspace". The "header" of a file is a collection of metadata in the form of keywords which is attached to the data and carried along during data analysis. All tools populate the header of the data they operate on, updating header keywords if changes are made. In particular, every run of a tool is recorded together with the tool parameters used in a HISTORY keyword, which makes it quite easy to trace how a dataset was generated.

The data "subspace" is a history of all filtering applied to a file (e.g. selection of a source region, energy interval, etc.). Since this information is attached to the data file, subsequent tools can use it to extract calibration information only appropriate for the selected filters. The subspace also allows users to see how the data file was altered during the analysis. All tools that use the DM library propagate the data subspace.

2.2. Files, New and Old

We have attempted to maintain back-compatibility with established FITS formats as much as possible. Given the new window *Chandra* opened on high-resolution spectroscopy over a broad band with well-calibrated instruments, we had to develop some new techniques and file formats for representing and manipulating these data. Based on experience with $1D$ and $2D$ UV and optical spectra, we extended formats and response files within the general

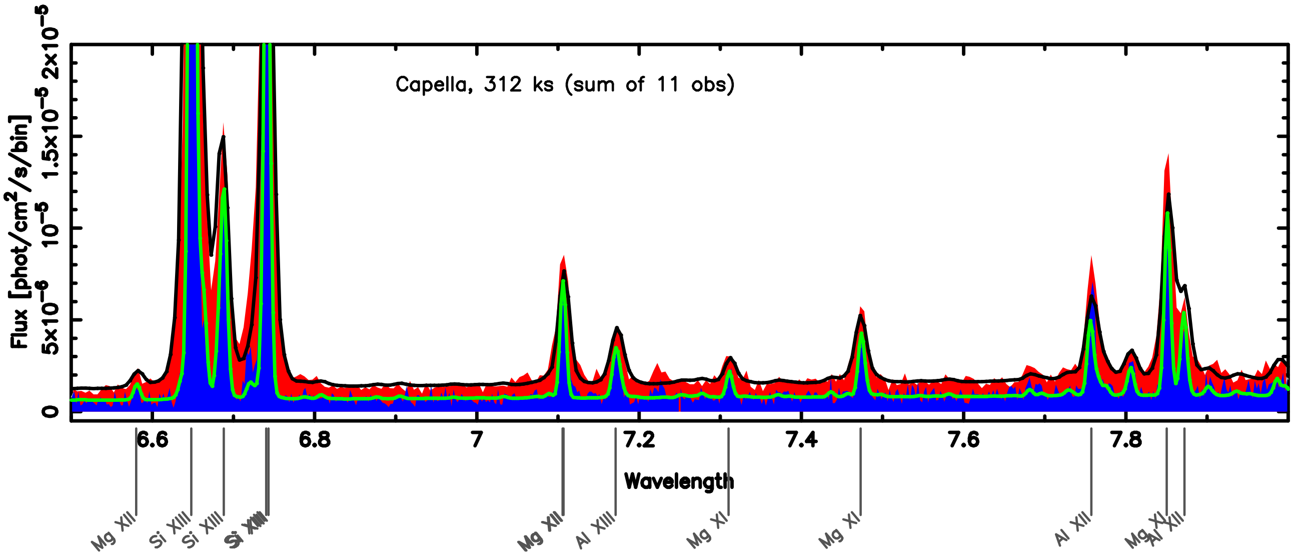

Figure 2. This shows a small portion of the High Energy Transmission Grating Spectrometer (`HETGS`) flux-calibrated spectrum of the star Capella, from the sum of 11 observations. The upper filled region (red) is from the Medium Energy Grating (MEG), and the lower (blue) is from the High Energy Grating (HEG) (HEG and MEG have different spectral resolutions and sensitivities). The solid lines outlining each region (black and green) are models derived from the atomic database (APED) convolved with the instrument responses. Line identifications are shown below the spectrum. Each individual observation and calibration were computed with standard `CIAO` tools, velocities and wavelength shifts computed with custom `S-Lang` programs, and spectra were combined and displayed with `ISIS`.

specification of X-ray event, spectral, and response formats to support new capability, but also to be largely compatible with existing software such as `XSPEC` and `FTOOLS`. For example, grating event files are augmented with diffraction coordinate and order sorting columns which are used when binning to $1D$ spectra or light curves; effective areas and instrumental line profiles are computed as a function of energy and position. However, the final products are still packaged in standard format "PHA" (counts spectrum), "ARF" (effective area), and "RMF" (spectral redistribution) files. All the the input calibration information was packaged into FITS files and stored in the Calibration Database (CalDB[7]), using established conventions.

High-resolution X-ray spectroscopy also generated a new need for access to atomic data, not just the final result as a binned spectrum. Since individual lines, or blends of a few lines, can be measured, new types of analysis can be done given access to individual emission line emissivities vs. temperature. The Astrophysical Plasma Emission Database (APED)[8] decouples atomic data from code. We have taken advantage of this to develop utilities for browsing the database and for use in modeling. Since the database is in FITS format, we have developed multiple interfaces, meeting different needs, such as with `ISIS` (see Section 3), `Sherpa`/GUIDE (see Section 2.4), or WebGUIDE[†]. Figure 2 shows a small section of a flux calibrated emission line spectrum with some line identifications and a model.

2.3. Graphical User Interfaces

A few graphical user interfaces (GUI) have been developed in `CIAO` as an alternative to command-line tools with the purpose of giving users an intuitive, visual interface for interacting with their data. While data analysis GUIs are appealing at the initial stages of the software learning process or for more "visual" tasks, we found that they are not always very easily extensible or very general, and are dropped in favor of command line tools by more experienced users or as soon as repetitive or complex tasks are performed.

However, one GUI application that has encountered a great success among `CIAO` users is `Prism`.

[†]`http://cxc.harvard.edu/atomdb/WebGUIDE/index.html`

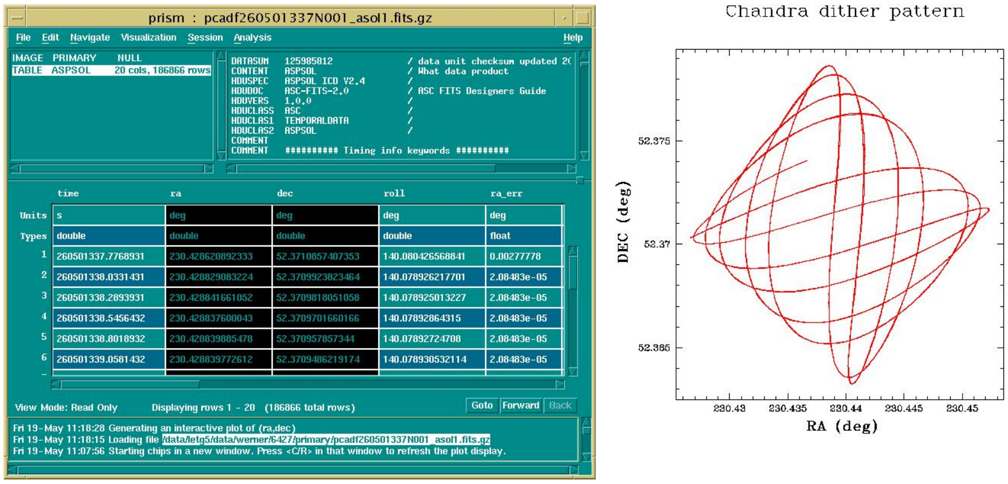

Figure 3. A Prism window browsing a *Chandra* aspect solution data file. A list of "extensions" in the file (in this case PRIMARY and ASPSOL) is listed in the top left panel, the header of the highlighted extension is visible on the top right panel, while the data are listed in the central panel. The bottom panel shows the commands that are launched from the prism window. In this example two columns (RA and DEC) are highlighted and are plotted through the "Visualization" menu via `ChIPS`. The result is shown in the plot on the right hand side. The figure represents the *Chandra* dither pattern around the aimpoint for this particular observation. The *Chandra* spacecraft has a built-in dither on its pointing position to average across calibration uncertainties: the effects of this complicated dither pattern must be taken into account by every instrument tool which is coordinate-dependent.

`Prism` is a format-independent (i.e. it can display any file format through the Data Model library) file browsing and manipulating tool. It lists the parts ("extensions") of a file and allows the user to view the header and data of each one. `Prism` offers many other analysis features including interactive plots of column data (including plotting errors and histograms), viewing of simple ASCII file, table editing, on-the-fly column selection, image display, header editing, and integration with other `CIAO` analysis tools.

Other GUI applications currently include `Peg` which allows users to easily view and modify parameter files and then run the associated tool with these values; and `Firstlook`, which provides users with a simple means of examining a *Chandra* dataset and it is intended to allow a quick examination of the data rather than perform detailed scientific analysis.

2.4. Sherpa and ChIPS: modeling, fitting and plotting

`Sherpa` is the `CIAO` modeling and fitting application which enables the analysis of multi-dimensional, multi-wavelength data (see elsewhere[9] for a detailed description of this application). `Sherpa` fits parameterized models (both parameterized algebraic expressions or parameterized code) to input data and is a fitting application which is "domain" and mission independent in the sense that it does not require particular axes to be fit (e.g. radius, energy, time) and it has no specific ties to the *Chandra* mission. It has been successfully used for the analysis of data from other observatories.

`Sherpa`'s main features include:

- the support of a variety of input and output data formats
- access to internal data and variables

- a model language which allows the construction of arbitrarily complex model expressions from simple definitions
- a library of pre-defined models, from simple equations to sophisticated algorithms
- ability to easily add user-defined models
- ability to cross reference and link model parameters
- an accessible library of fit statistics and optimization methods
- a selection of model comparison tests and parameter estimation methods
- 1-D and 2-D visualization capability
- interactive and scripting capability provided by the embedded `S-Lang`[‡] interpreted language

`Sherpa`'s capabilities are enhanced by the use of other programs and libraries included in `CIAO`. For example, the data input and output is provided by the `CIAO` Data Model library which allows a high-level abstract view of astronomical data files and gives access to several common astronomical file formats in addition to providing powerful filtering and binning capabilities that permit the extraction of specific portions of a data file.

During a typical `Sherpa` analysis session, users first input data and set filters and then define and build models that describe the best knowledge of either the data or the telescope and detector combination (e.g. instrument spectral response models or point spread function models) or both. `Sherpa` currently provides a library of 42 one- and 13 two-dimensional models plus all the one-dimensional models from the XSPEC[10] library v.11.3, all of which can be arbitrarily combined to build complex composite models. User-defined models can also be easily included. Multi-dimensional model expressions are also supported by `Sherpa` and therefore, for example, combination of 2-D surface and 1-D spectral fitting can be performed.

A choice of statistics and optimization methods (i.e. the act of minimizing the chosen statistics to obtain the "best-fit") allow users to tailor their analysis to their particular datasets or needs. For example the χ^2-based statistics is appropriate for Gaussian distributed data while Cash statistics is appropriate for low-count Poisson distributed data. Currently 8 different statistics and 13 different optimization methods are available in `Sherpa`. If several fits are performed (e.g. by changing model expression, data filters, etc.), the various "best" fit can be evaluated through model comparison tests and again `Sherpa` allows a choice of them. The last step in any modeling and fitting session is parameter estimation, i.e. the evaluation of confidence intervals (or errors) associated with each model parameter: currently three parameter estimation methods are available to users.

The most recent expansion to the `Sherpa` application[11,12] has been the inclusion of the `S-Lang` interpreted language which greatly expands not only `Sherpa`'s scripting capabilities but also gives users the ability to access and manipulate internal `Sherpa` variables (e.g. fit parameters, flux, errors, etc.) through a number of "set" and "get" functions.

`S-Lang` "modules" (libraries of `S-Lang` functions allowing access to several `CIAO` libraries) are directly accessible from `Sherpa` via `S-Lang`. They give users the ability to perform numerous operations within the `Sherpa` application. For example dynamic grouping of the data gives the flexibility to adjust the grouping (or binning) of a file within a `Sherpa` session rather than being required to run a separate tool.

Visualization in `Sherpa` is obtained through another `CIAO` application, `ChIPS`, the *Chandra* Plotting and Imaging System and the off-the-shelf visualization package `DS9`.[13] `ChIPS` is the plotting package of `CIAO` and can be used interactively or be driven by a script to create plots on the screen and in hardcopy form. In addition to being embedded within `Sherpa`, `ChIPS` is used by other `CIAO` applications (e.g. the file viewer `Prism`) for displaying data. ChIPS is designed so that plots can be built up interactively in small steps and can easily be refined, saved, printed, and restored. It is particularly flexible at creating plots from tables, but also allows users to generate plots of selected columns from an image or event file. The `S-Lang` programming language is also embedded in `ChIPS` and can be used to manipulate - or create - data for plotting.

[‡]http://www.s-lang.org/index.html

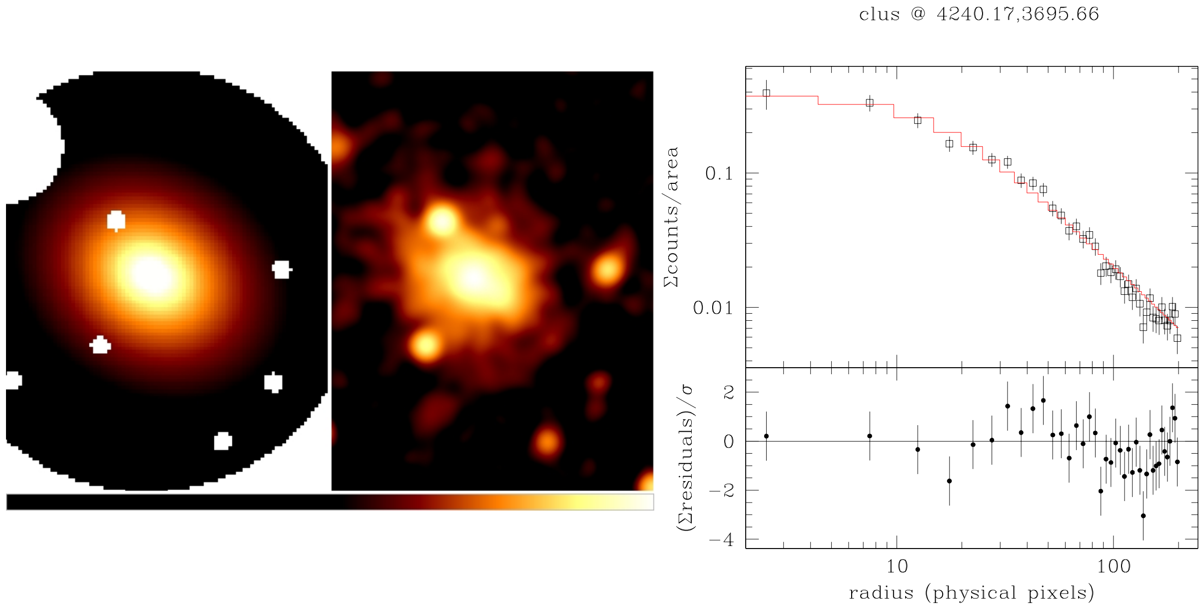

Figure 4.
An example of a 2D modeling and fitting and a radial profile extraction an fitting performed on a X-ray cluster using the **Sherpa** application. The left panel shows the best-fit 2D model (a two-dimensional version of $\mathrm{f(r)} = \mathrm{A}\times[1+(r/r_c)^2]^{(3\beta+0.5)}$ which includes ellipticity). The "null" values are regions, corresponding to point sources, excluded from the fit. The data (smoothed using the **CIAO** tool *aconvolve* for illustration purposes) is shown on the middle panel. Note however that the fitting has been performed to the un-smoothed data using the Cash statistics to account for the low count rate. The right panel represents a radial profile of best-fit 2D model (solid line) superimposed on the radial profile of the raw data. The profiles were generated within the **Sherpa** application with ellipticity and position angle taken from the best-fit values, rather than circular annuli.

Figure 4 shows an example of a **Sherpa**/**ChIPS** output where 2D modeling and fitting was performed on an X-ray dataset of a cluster of galaxies. A radial profile was then extracted within the **Sherpa** application and plotted using **ChIPS**. Extensive documentation on **Sherpa** can be found on the **CIAO** web pages and in particular at **http://cxc.harvard.edu/sherpa**

Both the **Sherpa** and **ChIPS** applications are currently being redesigned[14,15] with the following goals: (1) more robust, fast and programmable spatial and spectral fitting (2) more modular and easily maintainable code (3) ability to make plots of better publication quality (4) a plotting package usable both by users and by the *Chandra* validation and verification pipeline.

3. SCRIPTING LANGUAGE AND EXTENSIBILITY

An important lesson we learned which initially led to incremental changes, then to major re-writes of large components of **CIAO** (**Sherpa** and **ChIPS**; see above Section 2.4), is that to support *ad hoc* and arbitrary needs of forefront research, scientists need the ability to interactively script analysis, modeling, and visualization tasks, including the defininition of variables and evaluation of mathematical expressions which directly access data and models. An early prototype in this area was the Interactive Spectral Interpretation System (**ISIS**),[16] originally written for browsing the atomic database to support high resolution spectroscopy, but which evolved into a fitting and modeling system. **ISIS** introduced the scripting language, **S-Lang**, as an interpreter for interactive data analysis.

Based on this experience, the inclusion of a scripting language within **CIAO** appeared as an obvious benefit for users and software developers alike. After in-depth analysis and discussion within the **CIAO** development team and the larger community (see for example the "**S-Lang** as the **CIAO** Scripting Language" topic at http://cxc.harvard.edu/ciao/why/slang.html and the "Choosing a Scripting Language for CIAO" presentation at http://cxc.harvard.edu/ciao/presentations.html#slang), the S-Lang programming and scripting language (http://www.s-lang.org/) was chosen and integrated within CIAO since version 2.2. In addition to the benefits usually found in most scripting languages (e.g. extensibility, rapid prototyping, no compilation) S-Lang also satisfied our specific additional requirements:

- powerful mathematical capability
- concise syntax relatively natural for scientists to adopt
- embeddability into existing applications
- as small a footprint as possible

Starting from CIAO 3.0 the full S-Lang functionality has become easily accessible both within CIAO applications and at the unix command line through the use of slsh (the S-Lang shell), which is now distributed with CIAO. As further discussed below, all CIAO software capable of running S-Lang scripts (Sherpa, Chips, slsh, ISIS) can import a number of different libraries that have been wrapped in S-Lang, such as the GNU Scientific Library (GSL) Special functions, the DS9 image viewer, GTK, which includes the graphical filtering function vwhere,[17,18] and the Parallel Virtual Machine[19] (PVM; http://arxiv.org/abs/astro-ph/0510688).[§].

As mentioned in the previous section, one of the CIAO applications which has benefited greatly by the introduction of S-Lang is Sherpa. Sherpa -like XSPEC- was originally a classic 'black box': data were input from a file and analyzed with results possibly written out to disk; however, the data themselves could not be accessed once they were read in. By embedding S-Lang into Sherpa, the old 'black box' paradigm was moved to a new one in which numeric data can be accessed and freely manipulated through the functionality provided by the Sherpa/S-Lang library of functions.

The inclusion of a programming language in CIAO has empowered both advanced and average users. Advanced users have the the ability to automate their own complex tasks and also to import personal C++, C or Fortran code in scripts. Average users can benefit from additional scripts provided by the CIAO team in response to specific data analysis needs on timescale shorter than the official CIAO releases. Several of those scripts have already been released to the CIAO users' community and are used seamlessly together with other CIAO data analysis tools.

The S-Lang Interface Package (SLIRP)[¶] makes generation of importable libraries into almost a simple turnkey process, where a few simple commands can automatically create a custom user-module, or take a publicly available library and import it into CIAO via S-Lang. (See, for example, a description of using a S-Lang imported version of the Parallel Virtual Machine (PVM) library to perform complex, computer intensive analysis of the Chandra 1 Msec observation of the Cas A supernova remnant[20]).

4. SOFTWARE TESTING

A crucial step in the release process of the CIAO software is scientific and functional testing. A detailed description of the process can be found elsewhere.[21] Each release involves four stages of testing:

- **Stage 1: Unit Tests** At this stage software developers and scientists run unit tests on new and modified tools and applications and scientists run tools within science threads; the scientists report the results via test worksheets with pointers to example data; the tests are then incorporated into automated test scripts used for portability and later regression testing. The test lead scientist coordinates all the inputs.

[§]For a list of modules, see http://cxc.harvard.edu/ciao/slang/functions/modules.html and http://space.mit.edu/CXC/software/slang/modules/

[¶]http://space.mit.edu/cxc/software/slang/modules/slirp/

- **Stage 2: Mini-Test** Specialized regression test are run on key `CIAO` tools and on all new or modified tools; the tools are selected to test mainly new library functionality and to test out high risk areas, ensuring stability during preparations for a release

- **Stage 3: Full test** A regression test is run for all the tools in system and includes all the new tests added for each release via the input from the test worksheets. The test is run on all portability platforms and it is considered as passed only after the approval of the test lead scientist.

- **Stage 4: Package testing** Scientists verify the proper download of the software on each platform; in addition a brief 'smoke test' confirms that the as-packaged system does run. Web links, install instructions, and tar files are validated and, if they are correct, the release is approved. The official release document is then reviewed by CXC senior staff

The software testing stages are very intense and time consuming and could always benefit from extra time or testers: there is a very large number of tools/applications to test each with numerous parameters and therefore it is obviously not practical to test all possible paths through the code. The test suites developed for `CIAO` attempt to sample likely user cases and parameter values, but the resources do not always allow us to test all the cases that users may encounter.

5. DOCUMENTATION: MANUALS, THREADS AND HELP

The `CIAO` website (`http://cxc.harvard.edu/ciao`) is the primary documentation resource for users of the `CIAO` software package and contains links to the various kinds of documentation which are offered: mainly data analysis threads, analysis guides, "why" topics, help pages, manuals, bugs, "watch out" and FAQ lists. The obvious benefit of such on-line documentation is that it can be updated and expanded in very short time scales and permits cross-linking of information. The most challenging aspect however is that it becomes very easy to get "lost" in the multitude of web pages (several hundreds to date) so that documentation on a specific subject is not readily found.

Our experience early in the *Chandra* mission showed us that *thick manuals* are not always the best way to provide documentation to users when step-by-step instructions represent their initial needs. Therefore we decided to produce documents which are easy to update and easy to expand: the *data analysis threads*.

Threads are processing recipes designed to teach users by leading them, step-by-step, through procedures which range from basic to significantly complex; each thread describes the process in the same easy-to-follow manner. Since the first pioneering data analysis threads appeared on the `CIAO` web pages with the `CIAO` 1.1 releases (in December 1999) a lot of effort has been devoted to improving the thread structure, the thread content and especially the number of available threads in order to make them more reliable, easy to read, user-friendly and scientifically useful. Data analysis threads have a uniform and slim format (information common to various threads is summarized outside the threads and linked from them); in them each analysis tool is clearly identified and linked to its ahelp file; a history block is added at the end of each thread to make it easier to find the date and details of updates; and finally all threads are available in PDF format so that good quality hardcopies may be created. Threads are written in XML language and then automatically translated into HTML and PDF via the XSLT and HTMLDOC programs.[22]

The threads simply show how to perform a certain task, often without explaining why it is important or where it falls into the larger analysis session. To address these issues we developed inter-connected layers in the webpages which we called "analysis guides" and "why topics".

Looking at the large amount of documentation online, it can be difficult for `CIAO` user to know where to start. The analysis guides serve as a "roadmap" through the threads. Each guide takes a certain instrument configuration (e.g. ACIS) or type of analysis (e.g. extended sources) and lists the relevant threads. Along with the listing is a brief explanation of the purpose of each task, as well as the specific order in which they should be run, if any. The goal is to help the user to complete the analysis without requiring that every thread be read.

On the other hand, the threads sometimes do not contain enough information for the scientist interested in the details. Just as the analysis guides create a layer above the threads, the why topics are intended to dig deeper. Some of the topics describe aspects of the *Chandra* Observatory and data obtained with it, while others provide information on why certain science decisions are made.

All of this information combined - the guides, threads, and why topics - enable the user to tailor the analysis to a particular dataset or scientific goal. Currently more than 100 analysis threads are available from the `CIAO` web site together with several analysis guides and why topics.

In addition to web-based documentation, the `CIAO` system provides access to help files through the command line via the `ahelp` system. `ahelp` gives access to the `CIAO` documentation both in ASCII format or HTML format. An HTML version of all the ahelp files is also available on the `CIAO` web site. Help files are arranged by subject (or "context") and cross-correlated through keywords. Each `ahelp` file has a well-defined structure which includes a synopsis of the tool, the tool syntax, the description and examples. Ahelp files exist not only for tools or applications, but also for more generic subjects (called "concepts") (e.g. `ahelp subspace, ahelp merging_rules`). All ahelp files are written using the XML language and then translated into ASCII and HTML.

In a system as large as `CIAO`, maintaining up-to-date documentation is the biggest challenge and since a variety of documents is presented to the users it is very important to insure uniformity among them. One attempt to produce multiple documents in different format from the same source has been done in the design of the `Sherpa` documentation.

All `Sherpa` commands and concept ahelp files are written using specially-tagged XML files which are then translated in XML, HTML and latex format. The `Sherpa` manual is written entirely in latex and includes sections only pertinent to the manual (e.g. verbose descriptive paragraphs and introductory material) plus all the sections about concepts and commands generated from the `ahelp` files. This documentation design is good in principle to maintain the documentation current, avoiding duplication and automatically generating different kinds of documents from one source.

ACKNOWLEDGMENTS

The `CIAO` software package is the product of many years of work on the part of very numerous people and the authors of this paper, all current or recent members of the CXC Science Data System group (SDS) are purely reporters on behalf of a much larger group of contributors.

In particular we would like to acknowledge the valuable contributions from past members of the SDS group: David Davis, Adam Dobrzycki, Holly Jessop, Joel Kastner, Casey Law, Eric Schlegel, and Jennifer West.

None of the CIAO design and development would have been possible without the expertise and hard work of the members of the CXC Data System (DS) group and in particular Giuseppina Fabbiano, Janet DePonte Evans, Ian Evans, Mark Cresitello-Dittmar, Stephen Doe, Gregg Germain, Kenny Glotfelty, Tom Calderwood, Judy Chen, Roger Hain, Helen He, Janine Lyn, Joseph Masters, Warren McLaughlin, Robert Milaszewski, Joe Miller, Dan Nguyen, Jim Overly, Brian Refsdal, Chris Stawarz, Michele Wilson plus numerous past members of the DS group.

Support for development of `CIAO` is provided by the National Aeronautics and Space Administration through the *Chandra* X-ray Center, which is operated by the Smithsonian Astrophysical Observatory for and on behalf of the National Aeronautics and Space Administration under contract NAS 8-03060.

REFERENCES

1. D. Tody, "The IRAF Data Reduction and Analysis System," in *Instrumentation in astronomy VI; Proceedings of the Meeting, Tucson, AZ, Mar. 4-8, 1986. Part 2 (A87-36376 15-35). Bellingham, WA, Society of Photo-Optical Instrumentation Engineers, 1986, p. 733.*, D. L. Crawford, ed., pp. 733–+, 1986.
2. J. K. Blackburn, "FTOOLS: A FITS Data Processing and Analysis Software Package," in *ASP Conf. Ser. 77: Astronomical Data Analysis Software and Systems IV*, R. A. Shaw, H. E. Payne, and J. J. E. Hayes, eds., pp. 367–+, 1995.

3. K. Banse, D. Ponz, C. Ounnas, P. Grosbol, and R. Warmels, "The MIDAS Image Processing System," in *Instrumentation for Ground-Based Optical Astronomy, Present and Future. The Ninth Santa Cruz Summer Workshop in Astronomy and Astrophysics, July 13- 24, 1987, Lick Observatory. Editor, L.B. Robinson; Publisher, Springer-Verlag, New York, NY, 1988. LC # QB856 .S26 1987. ISBN # 0-387-96730-3. P.431, 1988*, L. B. Robinson, ed., pp. 431–+, 1988.
4. J. McDowell, "Data analysis with the Chandra data model library," in *Proc. SPIE Vol. 4477, p. 234-240, Astronomical Data Analysis, Jean-Luc Starck; Fionn D. Murtagh; Eds.*, J.-L. Starck and F. D. Murtagh, eds., pp. 234–240, Nov. 2001.
5. I. N. Evans and others, "The Chandra X-ray Observatory data processing system," these proceedings.
6. J. E. Davis, "The Formal Underpinnings of the Response Functions Used in X-Ray Spectral Analysis," *ApJ* **548**, pp. 1010–1019, Feb. 2001.
7. D. E. Graessle and others, "The Chandra X-ray Observatory Calibration Database (CalDB)," these proceedings.
8. R. K. Smith, N. S. Brickhouse, D. A. Liedahl, and J. C. Raymond, "Collisional Plasma Models with APEC/APED: Emission-Line Diagnostics of Hydrogen-like and Helium-like Ions," *ApJ Letters* **556**, pp. L91–L95, Aug. 2001.
9. P. Freeman, S. Doe, and A. Siemiginowska, "Sherpa: a mission-independent data analysis application," in *Proc. SPIE Vol. 4477, p. 76-87, Astronomical Data Analysis, Jean-Luc Starck; Fionn D. Murtagh; Eds.*, J.-L. Starck and F. D. Murtagh, eds., pp. 76–87, Nov. 2001.
10. K. A. Arnaud, "XSPEC: The First Ten Years," in *ASP Conf. Ser. 101: Astronomical Data Analysis Software and Systems V*, G. H. Jacoby and J. Barnes, eds., pp. 17–+, 1996.
11. D. J. Burke, A. Siemiginowska, S. Doe, and D. Nguyen, "Extending and Enhancing the Capabilities of the Sherpa Modeling and Fitting Tool with S-Lang," in *Astronomical Society of the Pacific Conference Series*, P. Shopbell, M. Britton, and R. Ebert, eds., pp. 105–+, Dec. 2005.
12. S. Doe, P. Freeman, R. Smith, D. Burke, and A. Siemiginowska, "Extending Sherpa Data Analysis with S-Lang," in *ASP Conf. Ser. 314: Astronomical Data Analysis Software and Systems (ADASS) XIII*, F. Ochsenbein, M. G. Allen, and D. Egret, eds., pp. 404–+, 2004.
13. W. A. Joye and E. Mandel, "New Features of SAOImage DS9," in *ASP Conf. Ser. 295: Astronomical Data Analysis Software and Systems XII*, H. E. Payne, R. I. Jedrzejewski, and R. N. Hook, eds., pp. 489–+, 2003.
14. S. Doe and others, "Sherpa: Goals and Design for Chandra and Beyond," in *ASP Conference Series: ADASS XV*, in press.
15. G. Germain, W. McLaughlin, R. Milaszewski, J. Miller, J. DePonte Evans, I. Evans, and D. Burke, "CHIPS - CIAO's Powerful New Plotting Package," in *ASP Conference Series: ADASS XV*, in press.
16. J. C. Houck, "ISIS: The Interactive Spectral Interpretation System," in *High Resolution X-ray Spectroscopy with XMM-Newton and Chandra*, G. Branduardi-Raymont, ed., Dec. 2002.
17. M. S. Noble, "VWhere: A Visual, Extensible 'where' Command," in *Astronomical Society of the Pacific Conference Series*, P. Shopbell, M. Britton, and R. Ebert, eds., pp. 237–+, Dec. 2005.
18. F. Primini, M. Noble, and The CXC Science Data Systems Group, "Extending the Capabilities of CIAO with S-Lang-based Tools," in *Astronomical Society of the Pacific Conference Series*, P. Shopbell, M. Britton, and R. Ebert, eds., pp. 17–+, Dec. 2005.
19. M. S. Noble, J. C. Houck, J. E. Davis, A. Y. Young, and M. Nowak, "Using the Parallel Virtual Machine for Everyday Analysis," in *Astronomical Society of the Pacific Conference Series*, C. Gabriel, D. Arviset, D. Ponz, and E. Solano, eds., p. in press, 2006.
20. J. E. Davis, J. C. Houck, G. E. Allen, and M. D. Stage, "Analyzing the Cas A Megasecond in Less than a Megasecond," in *Astronomical Society of the Pacific Conference Series*, P. Shopbell, M. Britton, and R. Ebert, eds., pp. 444–+, Dec. 2005.
21. M. Karovska, J. DePonte Evans, and the CIAO test team, "Testing of CIAO," in *ASP Conference Series: ADASS XV*, in press.
22. E. C. Galle, D. J. Burke, C. Stawarz, and A. Fruscione, "The CIAO Website Unveiled," in *Astronomical Society of the Pacific Conference Series*, P. Shopbell, M. Britton, and R. Ebert, eds., pp. 473–+, Dec. 2005.