CXC-DM-001

CXC Data Model

Vol. 1

Introduction: Data analysis with the Chandra Data Model library

Jonathan McDowell
Chandra X-ray Center

October 22, 2001

Contents

CXC DM Documentation Summary

The CXC Data Model software package includes documentation for the design, the programming interface, and for users. This document summarizes the rest of the documentation tree.

Vol 1: Introduction: Data analysis with the Chandra Data Model library

This document; introduces the DM and describes the documentation tree.

Vol 2: Data Model Abstract Design

Describes the object-oriented data model concept, introducing the DM-specific terminology of blocks, descriptors, etc. and describing their properties.

Vol 3: Data Model Library Architecture

Describes the redesigned DM architecture, including layer diagrams of the software library's components.

Vol 4: Virtual File Reference

Describes the virtual file concept, design and syntax.

Vol 5: FITS Kernel Design

Describes the mapping of FITS format files to the DM.

Vol 6: IRAF Kernel Design

Describes the mapping of IMH and QPOE format files to the DM.

Vol 7: ASCII Kernel Design

Describes the design of the ASCII kernel.

Vol 8: C Programmers' Guide

User manual for applications programmers.

Vol 9: C Programmers' Reference

Reference manual for applications programmers.

Vol 10: User Introduction

Takes the user through simple use of the DM.

Vol 11: Toolkit User Guide

Reference guide to the DM toolkit.

Vol 12: User Reference

Reference guide to virtual file syntax.

Vol 13: Fortran Programmers' Reference

Reference manual for users of the Fortran wrappers.

Vol 14: Fortran Wrapper Design

Describes the design of the Fortran wrappers for the DM.

Data Analysis and the Chandra Data Model Library

1 Introduction

The Chandra Data Model (CDM) library was developed to support data analysis for the Chandra X-ray Observatory, one of NASA's orbiting Great Observatories. The library and its associated tools are designed to be multi-mission and can be used to manipulate a wide variety of astronomical data. Much of the library's power comes from its use of 'virtual files', which provide a flexible command-line user interface.

The CDM library[1,2,3] is a high level, object-oriented abstract interface to astronomical data which underlies the Chandra X-ray Center's CIAO data analysis system. The CDM library has been in use internally at the CXC since 1997, and the CDM tools were released to the public in 1999 as part of CIAO. The CDM is now a widely used and robust package.

The most important goals of an abstract data interface are to combine the power that comes from generality with an interaction that is close to the astronomer's naive expectation. A simple step along this path is to shield the user from implementation details without making them inaccessible. There are several levels of implementation detail, including the basic file format used (often FITS[4,5,6] in astronomy) and the file design. For instance, in different astronomical applications a spectrum may be represented by a simple image array, or as a table. The CDM makes it possible to successfully hide these different implementations from the user. The structure of the CDM helps the applications programmer develop tools which are designed to be generic rather than hard coded to the particular mission or application. Finally, the CDM provides an integrated 'virtual file' filtering capability which allows any CDM-enabled tool to operate on user defined subsets of their data files.

2 Data Analysis in X-ray Astronomy

In contrast to most sub-disciplines of astronomy, the primary datasets in high energy astronomy are not images but tables listing each individual photon and its properties. These 'event lists', together with appropriate software, permits astronomers to manipulate their data in sophisticated ways. Of course, for later stages of processing the traditional image file is also crucial, and many of the techniques used in the rest of astronomy are applied, albeit sometimes with modifications to handle the problem of small number statistics which is omnipresent in our photon-limited X-ray data.

The high energy astronomy community has been fortunate in leveraging the power of astronomy's FITS format as a common data format. In 1990 collaboration[7] between NASA-Goddard and SAO on the ROSAT data formats led to common high energy header conventions and file designs that permitted effective multi-mission analysis, that is, use of the same software to analyse data from different missions either separately or simultaneously. The IRAF/PROS[8] and FTOOLS[9] systems were primarly developed for the ROSAT and ASCA satellites respectively but have both been used for other missions, including Chandra.

The Chandra X-ray Observatory, launched in July 1999, combines high spatial and spectral resolution, and features instruments with complicated geometries, large numbers of pixels and many different operating modes. To address the data analysis challenge this presented, the Chandra X-ray Center developed the CXC automated pipeline processing system and the CIAO (Chandra Interactive Analysis of Observations) user analysis package. The pipeline system and CIAO share a common infrastructure and many individual software components, and both are layered on the CDM.

3 The motivation for the CDM

Most data analysis systems have an interface which closely maps to their data files. These data files, in turn, tend to be highly mission-specific and are tuned for both efficiency and historical compatibility. Local conventions are usually implicit in code, so the data files are not fully self-describing. This has the result that the same scientific analysis problem has very different implementation for different missions.

The user rarely cares about these issues - they want an interface which maps to the science, and they want similar problems to look similar. Users also would like fine control over presentation and units, and traceability to calibration and references.

Our partial solution to these problems is the data model approach[10], in which a general, object oriented description of the data is used in the interface, tied as closely as possible to the science concepts. The file design implementation is accessed by 'kernels', which map from the file to the abstract model. However, we must balance the generic approach with the ability to access the original raw information.

Although our software works with both FITS and IMH/QPOE (a special pair of formats native to the IRAF analysis package) and will soon be upgraded to also work with simple text files, most of the usage to date has been with the FITS kernel. We have enriched existing FITS conventions with back-compatible extra layers allowing us to improve the completeness of the data's self-description.

Early publication of the 'virtual file syntax' user interface[1,11] led to its adoption (with small differences) by the FTOOLS analysis system[12] developed by NASA-Goddard, which is also used for analysing Chandra data. We note that the abstract CDM approach in CIAO preserves coordinate and image blocking information, automatically updates exposure times and preserves filtering histories, and makes a number of different keyword conventions mutually transparent, reducing the need for the user to learn details of FITS. On the other hand, FTOOLS allows more direct access to the actual FITS file, which can be an advantage for some applications.

4 The CDM toolkit

The CDM tools make it easy to make filtered subsets of data tables. For instance, users may wish to select specific ranges of time, energy and sky region and make a photon event list which only contains data from those ranges.

The associated Region Library allows users to specify complicated two-dimensional regions, which is important for analysing the high spatial resolution Chandra data. For instance, one may wish to define an annular background region around a source while excluding a circle around an overlapping contaminating source. In the CDM such regions are defined in a simple language; the user can specify regions directly in the language, or generate them with an interactive graphical application and save them to a text file. I believe the availability of a well-defined (public, text-based) command-line interface is crucial and must drive the development of GUIs rather than vice versa, since scriptability and reproducibility are vital for systematic processing of large numbers of datasets - and for an individual user, large may be 'more than two'.

Another capability of the CDM tools is to make images from tabular data in various coordinate systems by forming 2-dimensional histograms.

We can also make 1-dimensional histograms from event data, generating spectra (by binning on energy), lightcurves (binning on time) and radial intensity profile (binning on spatial region).

The toolkit allows users to inspect the data in terms of its data model (high level) interpretation. For instance, instead of printing the names and values of header keywords used to describe coordinate systems within the WCS convention, the coordinate transformation itself is displayed as a symbolic equation.

To give a flavor of the CDM software, I give some examples below.

Example 1: Detector Image

Unlike integrating telescopes (e.g. HST WF/PC), X-ray telescopes often move with respect to the sky, and the celestial positions of the detected photons are reconstructed by matching the photon time-tags with the location of the telescope axis at that time as derived from star tracker data. The event list datasets give the position of each photon in both detector and celestial coordinates.

To make an image in detector coordinates from an event list data.evt, we use the **dmcopy** tool

```
dmcopy "data.evt[bin detx=8,dety=8]"  det8.img
```

This makes a 2-D histogram of the full field, with bins (image pixels) which are 32 times coarser than the native pixel size of the detector. Our detectors are too big to make a full field, full resolution image practical. Suppose we see a source near the center of the field at (4096,4096) in detector pixel coordinates. We can make a higher resolution image

```
dmcopy "data.evt[bin detx=3500:4500:2,dety=3500:4500:2]"  det2.img
```

Here we specify a starting and ending pixel for both detx and dety, as well as a binning factor of 2. But from other analysis we may know that the background is high outside a certain energy range, and particularly high during a solar flare at a particular mission time between 63940080 and 63940180 seconds.

```
dmcopy "data.evt[energy=500:2000,time=:63940080,63940180:][bin detx=3500:4500:2,dety=3500:4500:2]"
              det2f.img
```

This command makes an image which has been filtered in time and energy, illustrated in the figure.

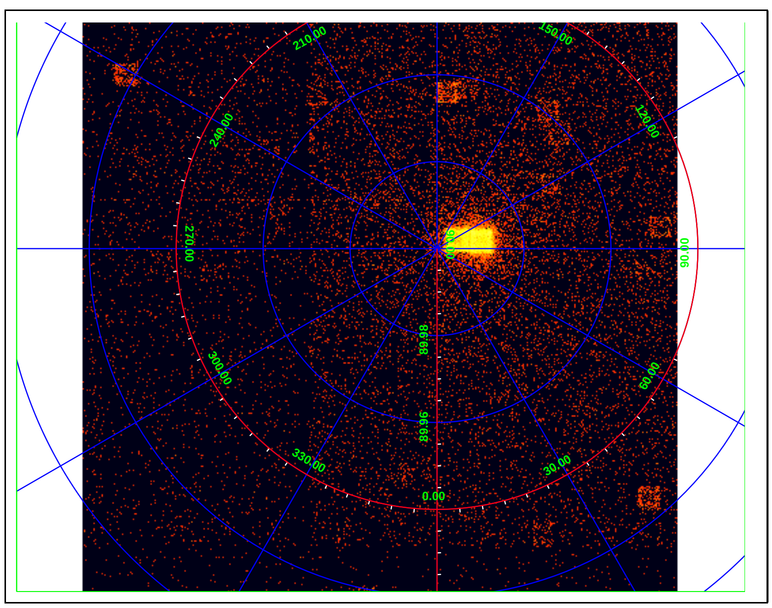

Figure 1: Detector image, filtered on energy and time.

Example 2: Sky Image

Now let's look at the same data but in sky coordinates:

```
dmcopy "merge3e.fits[energy=500:2000,time=:63940080,63940180:,
            detx=3500:4500,dety=3500:4500]
           [bin x=3200:4800:2,y=3200:4800:2]" sky.img
```

Here we have filtered on the same detector coordinate range, but made an image in (x,y), the sky coordinates.

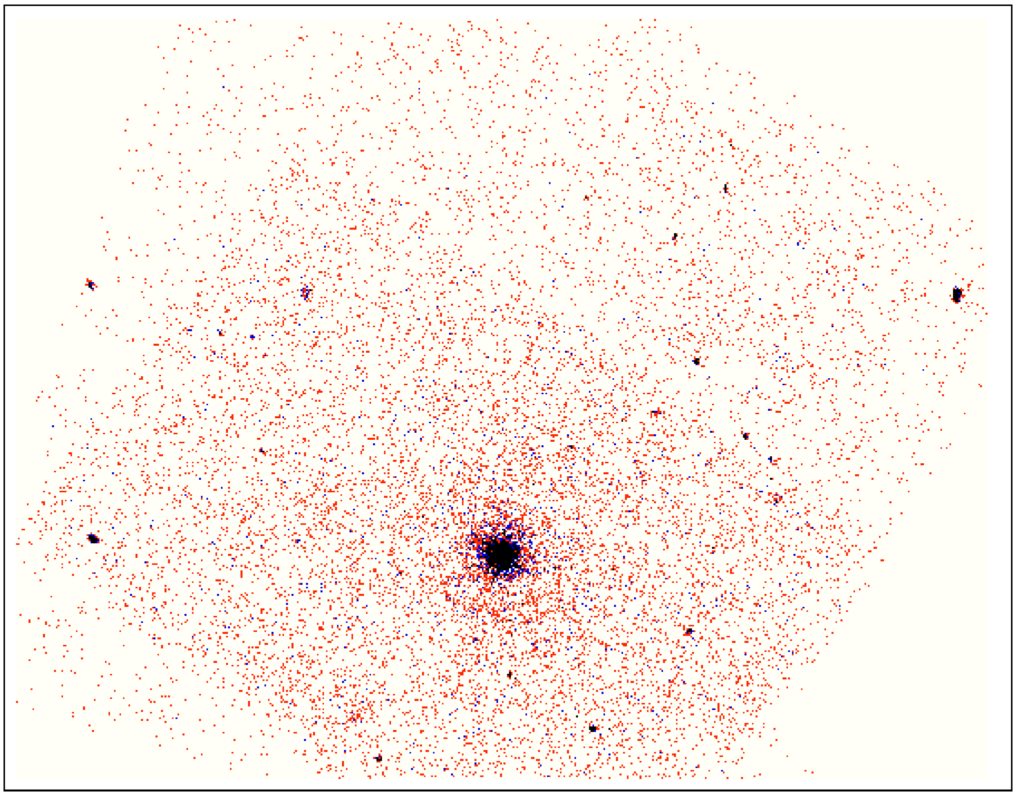

Figure 2: Sky coordinate image of the same data

Example 3: Merging tables

The whole field was created by merging three separate observations.

```
dmmerge "786.fits,787.fits[exclude sky=circle(4096,4096,100)],1730.fits"
                outfile=merge3e.fits
```

The three observations of the nearby galaxy M33 were first reprojected to a common aimpoint using a specialized tool, and the resulting photon event lists were then merged using the generic tool dmmerge. Data from the file 787.fits was edited to remove a central region where the data are compromised. Since all of the CIAO tools open their data files via the virtual file interface, there is no need to make a separate edited version of 787.fits on disk using dmcopy - the dmmerge tool can do the editing on the fly, since the filtering capabilities are inherent in the data I/O library and do not need to be added to each tool separately.

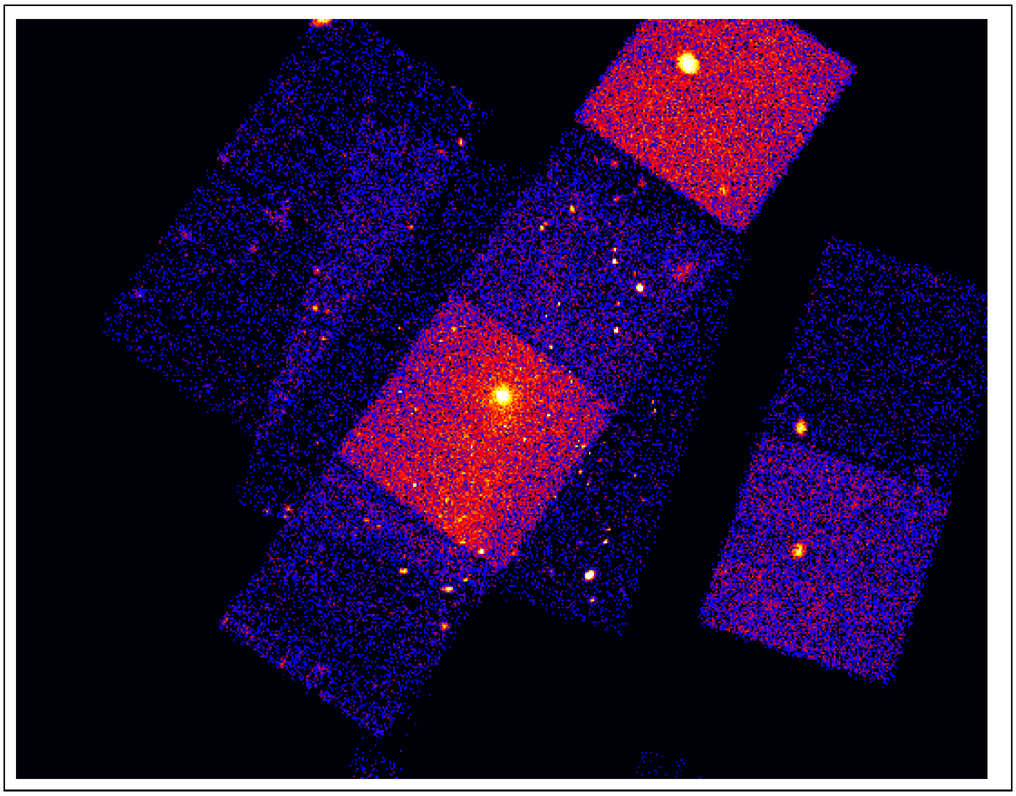

Figure 3: Three merged Chandra observations showing point sources in the nearby galaxy M33

Example 4: The interpolated join

A common operation in data analysis is the need to combine data which are sampled on different intervals. For instance, spacecraft housekeeping data might be sampled on a regular 2.5 second interval, while primary science data might have arbitrarily spaced time tags. A similar operation is common in analysis with derived products: consider a star catalog whose columns include a photometric color BVCOL and a sorted lookup table which maps BVCOL to TEFF, the stellar effective temperature. We'd like to be able to add a new column to the catalog giving the effective temperature for each star; this is a bit like a database join operation, except that the exact value of the star's BVCOL may not be in the lookup table - we have to interpolate. I call this procedure an 'interpolated join', and limited support for it is provided in the forthcoming CIAO 2.2 release with the dmjoin tool:

```
dmjoin catalog.fits joinfile="lookup.fits[cols bvcol,teff]"
     join=bvcol outfile=result.fits
```

5 The CDM object model

The design of the CDM is based on a small number of fundamental objects:

- the Descriptor object, which corresponds to a named data array, table column, header keyword, filtering range, or coordinate transformation.
- the Block object, which describes a single table or image with its associated metadata.
- the Dataset object, which describes a related collection of tables or images. In FITS, the Dataset corresponds to a single FITS file.
- the Stack object, which describes a more general collection of tables or images, possibly a set of files.

The same physical quantity - an (RA,Dec) pair say - might be a pair of header keywords, or table columns, or might be implicit in a coordinate transformation on a pair of pixel values. The CDM Descriptor makes explicit the parallels between these cases, and provides conventions to handle cases where the parallel is not complete (for instance, specifiying the units is not supported for FITS header keywords, but a local convention has been developed which is supported by our software and FTOOLS). Our design even allows these boundaries to be blurred - in our forthcoming release, a table column access routine will succeed if the named object is really just a header keyword, and will pretend the object is a column with the same value in each row of the table. This makes the code more robust to small changes in file design.

The CDM Block corresponds to a FITS HDU, but we have used different terminology to allow for different ways of storing tables and images in other file formats. In our IRAF/QPOE kernel, the Dataset actually maps to a directory in which individual IMH and QPOE files are stored, and the Block maps to the individual files. The Dataset copy operation then implicitly defines a mapping between an arbitrary FITS file and a set of IMH/QPOE files. The Block consists of Header, Data and Data Subspace subcomponents, each with identical substructure. The Data Subspace, a concept introduced in the CDM, contains a dynamic record of the filtering applied to the data.

The Stack object, which is currently implemented at the toolkit level rather than in the CDM library layer, allows a set of similar files to be handled by a single command. The dmcopy and dmlist tools will operate on stacks of files, but there is no real efficiency gain in doing this instead of running the tools multiple times. However, stacks are very useful in more specialized tools: the main calibration tool for Chandra's ACIS camera, acis_process_events, needs to be fed both a photon event file and an 'aspect solution'. The aspect solution is not sliced in time on observation boundaries, so the user may need to use parts of several aspect files. This detail is handled by making a text file containing the list of required aspect filenames, and passing it to acis_process_events as a stack with a simple CDM time filter applied; the program can equally well accept a simple single aspect file, and no special handling is required to distinguish the two cases.

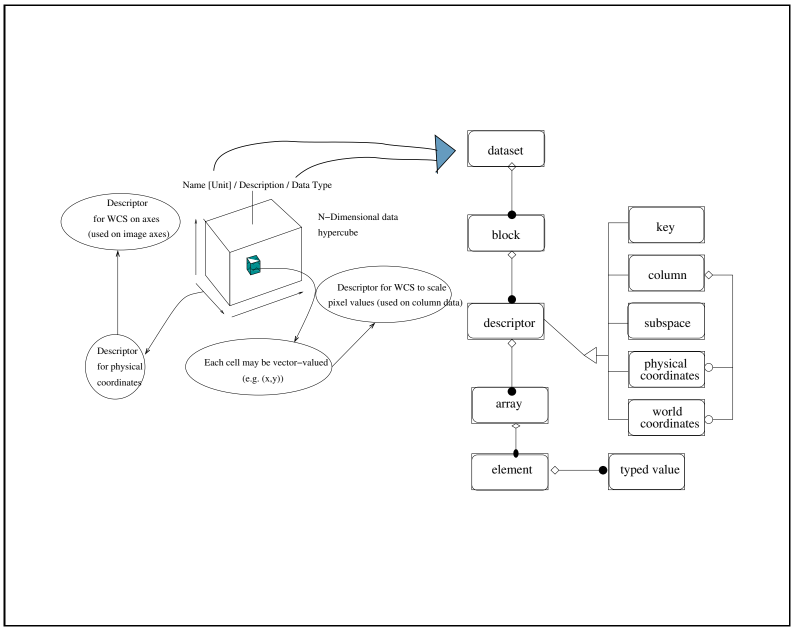

Figure 4: Conceptual structure of the CDM

6 The CDM virtual file syntax

The full virtual file syntax, which defines a single Block object, is

```
dataset_name[block_name][filter-cmd][cols-cmd][bin-cmd][opt-cmd][rename-cmd]
```

The block name allows the user to select an individual block within a datafile; if omitted, the library attempts to guess which block is the one with the main data in it. The filter command selects a subset of rows within a table or pixels within an image. The cols command selects a subset of columns within an image, possibly renaming some of them. The bin command rebins an image, or creates an image by binning up specified columns of a table. The opt command allows the user to tweak internal parameters of the library. The rename command allows the virtual output block to be given a different name from the underlying block. Full details of the syntax are available on the CIAO web page.

7 Enhancements for the NVO Era

The National Virtual Observatory (NVO)[13] will require higher level toolkits for making multiwaveband analysis easier. The CDM tools will work at an elementary level on datasets from other wavebands (we have tested simple image and coordinate operations on data from ISO and HST) but deeper analysis is required to make tools which are well tuned to NVO applications. Higher level objects (spectral time series, sets of images) have different file designs for different disciplines. A wavelength calibration may be stored in a header (optical, IRAF), a separate file (HST), or another table column (Chandra gratings). The user should not be required to know these details to access the information, which implies a way of creating software mappings between the science concept (spectrum with calibration) and the file implementation. The CDM approach, in which analysis is already separated from the file details, is well suited to support such capabilities.

ACKNOWLEDGMENTS

The CDM was developed at the Chandra X-ray Center (CXC), located at the Smithsonian Astrophysical Observatory (SAO). The software was developed in collaboration with Martin Elvis, Michael Noble, Kenny Glotfelty, Scott Randall and Oly Oberdorf. I acknowledge many useful discussions with Pepi Fabbiano, Aneta Siemiginowska, Antonella Fruscione, Doug Burke, Doug Tody, Keith Arnaud, and Bill Pence. The CDM work was supported by SAO's CXC grant from NASA. A version of this article was presented at SPIE in Aug 2001.

REFERENCES

1. Chandra X-ray Center, "The CXC Data Manipulation User's Guide", Release 1.1. This guide and the CIAO software are available from asc.harvard.edu.
2. J.C. McDowell, M.S. Noble, K. Glotfelty, O. Oberdorf, and S. Randall, "The CXC Data Model Programmer's Guide", Release 1.99 (available from hea-www.harvard.edu/ jcm/asc/docs).
3. J.C. McDowell and M. Elvis, "Propagating Uncertainties and Units in Data Structures", ADASS 4, 195 (1995).
4. D.C. Wells, E.W. Greisen, and R.H. Harten, "FITS: A Flexible Image Transport System", Astron. Astrophys. Suppl. Series 44, 363-370, 1981.
5. W.D. Cotton, D. B. Tody, and W.D. Pence, "Binary Table Extension to FITS", Astro. Astrophys. Suppl. Series 113, 159, 1995.
6. NOST FITS Standard, NASA FITS Support Office, 2000 (fits.gsfc.nasa.gov).
7. M.F. Corcoran, W.D. Pence., R. White, M. Conroy, "The ROSAT Implementation of a Proposed Multi-Mission Data Format," ADASS 2, 549 (1993).
8. D.M. Worrall, M. Conroy, J. Deponte, F.R. Harnden, Jr., E. Mandel, S.S. Murray, G. Trinchieri, M. Vanhilst, B.J. Wilkes, "PROS - Data Analysis for ROSAT", *Data Analysis in Astronomy - IV*, 145 (1992).
9. W.D. Pence, J. Blackburn, and E. Greene, "FTOOLS - a FITS Utility Package For Multiple Environments", ADASS 2, 541 (1993).
10. A. Farris and R.J. Allen, "Search for a Common Data Model for Astronomical Data Analysis Systems", ADASS 1, 157 (1992).

11. J.C. McDowell, M. Noble, and M. Elvis, "AXAF Data and Data Manipulation Software: the ASC Data Model," Legacy **7**, 64 (1998).
12. W.D. Pence, "CFITSIO v2.0: A New Full-Featured Data Interface," ADASS 8, 487 (1999).
13. R.J. Hanisch, "Technology Drivers for the Virtual Observatory", AAS 197th Meeting, Paper 122.02, 2000.