Astronomical Data Analysis Software and Systems IX
ASP Conference Series, Vol. 216, 2000
N. Manset, C. Veillet, and D. Crabtree, eds.

The Data Inspector: Data Visualization for the Chandra Monitoring Project

J. G. Petreshock[1], S. J. Wolk[1], M. Cresitello-Dittmar[1]

Harvard-Smithsonian Center for Astrophysics, Cambridge, MA 02138

Abstract. Early on in support of the development of the monitoring system for the Chandra X-ray Observatory (CXO) we discovered the need for a very general, very flexible system to visualize data products. We developed the Data Inspector (DI). The DI is a package of procedures designed to create a generalized multi-purpose data analysis and visualization toolkit focused around Chandra requirements. We will discuss the DI package and the functionalities provided within to meet the Monitoring and Trends Analysis project requirements.

1. Introduction

The primary vehicle for general data interpretation is through various techniques of data visualization. The M&TA[1] project (Wolk et al. 2000) for the CXO is tasked with characterizing system changes to maximize scientific return. As a result of this requirement we determined that a multipurpose, flexible data visualization package was needed.

To facilitate M&TA analysis we mandated the following requirements for the design of our visualization tool. The tools should accept all Chandra X-Ray Center Data System (CXCDS) data products in the form of Flexible Image Transport System (FITS) files as input and utilize standard FITS keywords and those keywords specified within the CXCDS system. To aid analysis efforts during an anomaly resolution events, all plots will have abscissa (generally time) aligned axes with limit violation codes drawn in the background. The capability shall be provided to save analysis plots in hardcopy version for use in spacecraft and SI reports. The final primary requirement was the provision of an interface for manipulation of plot styles to provide flexibility for report generation.

2. The Data Inspector

The DI was written using the latest technologies available with Interactive Data Language (IDL[2]) version 5.0+. Users do not need any prior knowledge of IDL,

---

[1]Chandra X-ray Science Center

[1]`http://asc.harvard.edu/mta`

[2]`Research Systems Inc. Boulder CO, http://www.rsinc.com`

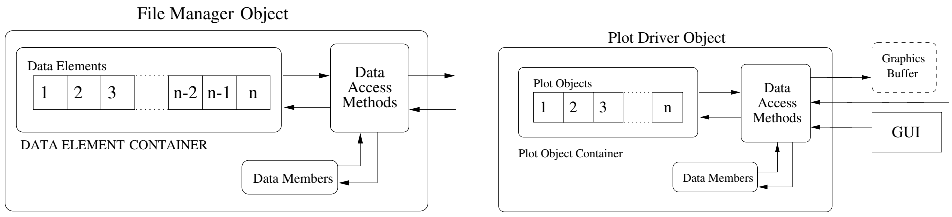

Figure 1. DI class design drawings.

however the more familiar the user is with IDL the more easily they will be able follow the operation of this software. Using Object Oriented (OO) design the DI utilizes many built-in features of IDL to perform a multitude of tasks. Such features include; the Object Draw (OD) capability, the Direct Draw (DD) capability, as well as analysis routines.

The current version, written under Solaris 2.6, provides the following functionality; strip plot generation, 3-D visualizations, data analysis, and hardcopy generation. The DI performs many tasks which can be subdivided into four general categories; File I/O, Storage, Visualization, and Analysis.

2.1. File I/O

The DI was designed to read data stored in FITS binary extensions with multiple extensions. These FITS files are read using the generalized FITS reader MRDFITS developed by Tom McGlynn as part of the IDL Astronomy Users's Library (Landsman 1994).

Although the DI is best suited to handle scalar columns this is not necessary. The DI is capable of producing three basic output formats; GIF or JPEG formats images, encapsulated and non-encapsulated postscript, and ASCII text files.

2.2. Data Storage

The data for each column within loaded FITS files are stored as instantiations of the data element class. This class consists of access functions. The data members consist of pointers to abscissa and ordinate, as well as range information such as minimum and maximum values. At the cost of added memory the latter data members reduce the need to perform potentially time consuming calculations.

The data element objects are stored within an object container and are accessible through method functions within a controller class; the file manager class. The file manager class design is described in Figure 1.

2.3. Visualization

The DI provides three primary methods for visualizing data. These include: strip chart plots, 3-D visualizer, and the table viewer.

The strip chart plotting allows users to generate abscissa aligned plots. The plot driver class, described in Figure 1, provides controls for layout handling. These controls consist of the number of plot columns and rows, scale factors

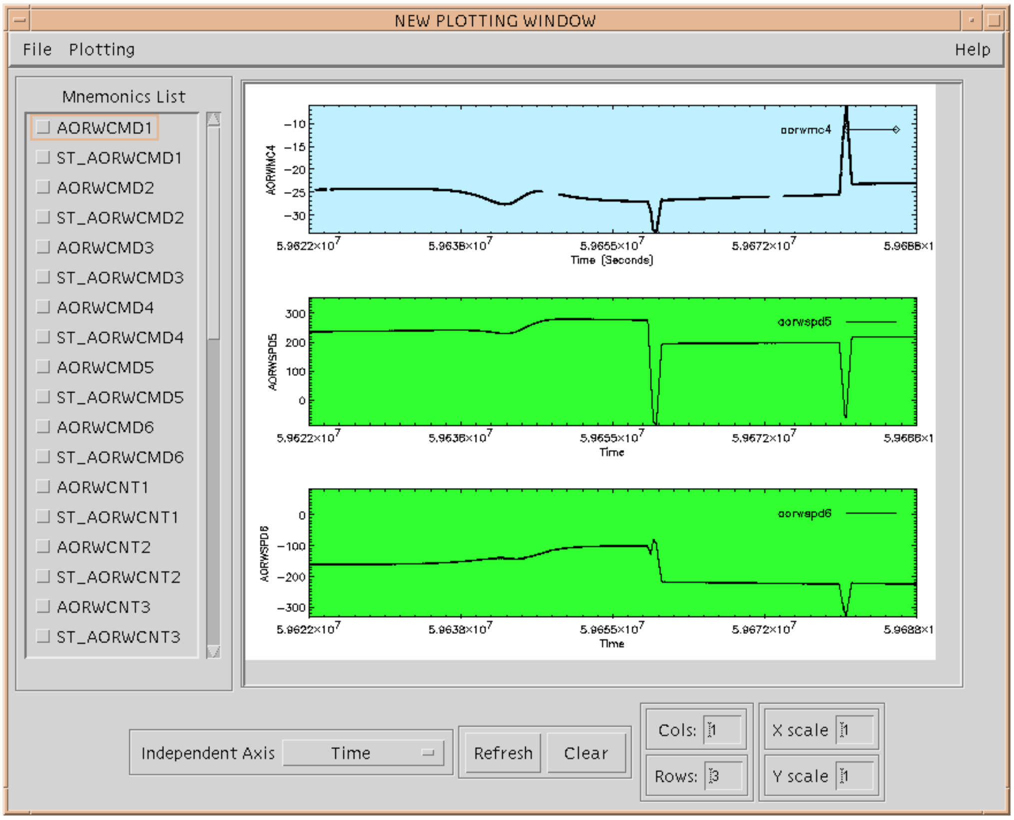

Figure 2. Strip chart plotting GUI.

for the image window size, and abscissa axis selection. Also, a plot style editor is provided by the plot driver to adjust the plot styles (i.e. line style, color, symbol, titles, etc.). For efficiency reasons, the strip plotting is executed via the DD methods. The M&TA limit violation codes that are drawn in plot background are defined within an ASCII file that contains the red and yellow (both upper and lower) violation limits. This file can hold multiple entries for each data set in order to describe limits that vary according to spacecraft state.

The 3-D visualizer provided the capability to plot three data sets within a Cartesian (xyz) coordinate system. The 3-D viewer was designed to utilize the OD graphics capability provided by IDL. The track ball object is employed to apply mouse driven events to change the viewing perspective.

A human readable table of data values can be created from a user selectable subset of the loaded data sets using the table viewers. This tool allows users to inspect the numeric values more clearly than is possible through the plotting tools.

2.4. Analysis

The DI provides two primary data analysis tools; the data modeling class, and the expression generation class. The primary analysis tool is the data modeling class which wraps the IDL fitting routines into a Graphical User Interface (GUI) for fitting datasets loaded within the strip plot windows. This class is capable of producing linear, linear with outliers, gaussian + quadratic, polynomial, exponential and many other fitting models.

The secondary analysis tool is the Expression Generation Gadget (EGG). This class provides an interface between the user and file manager object to generate new datasets using expressions formated in a similar fashion to those used on the IDL command line. Operations are not limited to *+-/, but any IDL function can also be used as long as the expression is properly generated.

The DI provides the functionality to calculate the derivative between any pair of datasets via a 3-point Lagrangian interpolation method. In a similar fashion, the DI can be used to calculate correlation coefficients between any pair of datasets. The statistical summary task produces a table reporting the the minimum, maximum, mean, and standard deviation of all the loaded datasets. There are also specified procedures to visualize the focal plane instruments and the thermal environment of the mirrors.

3. Conclusion

The DI has already been field tested and has been proven to be a useful tool for data visualization during both calibration of the CXO and flight operations. All of the tasks we described are currently performed in an interactive mode through GUI interfaces. However, the initial design will allow these classes to be used to generate scripts for use generating trend analysis reports, and SI reports.

IDL provides many useful aspects for developing software. The IDL language provides rapid prototyping and development times, a flexible software interface design. Another benefit is the potential for cross platform application development. More information regarding the DI can be found at the Data Inspector homepage[3].

Acknowledgments. This project is supported by the Chandra X-ray Center under NASA contract NAS8-39073.

References

Landsman, W. B. 1994, in ASP Conf. Ser., Vol. 77, Astronomical Data Analysis Software and Systems IV, ed. R. A. Shaw, H. E. Payne, & J. J. E. Hayes (San Francisco: ASP), 437

Wolk, S. J. et al. 2000, this volume, 28

[3] `http://asc.harvard.edu/~sopdev/mta/DI`