dax: ds9 analysis extensions in CIAO

Kenny J. Glotfelty (SAO), Joseph Miller (SAO), Judy Chen (SAO)

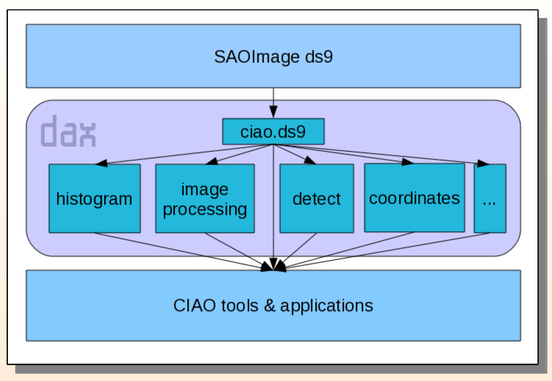
Chandra X-ray Center

Abstract

dax is a suite of scripts that allows various CIAO tools (written to support the Chandra X-ray Observatory) to be run using SAOImage ds9's analysis framework. This allows users to quickly leverage the functionality CIAO provides without having to invest in learning the syntax and semantics of each of the tools. This simplification of the interface benefits all astronomers since many of the CIAO tools are sufficiently generic that they can work with data sets from arbitrary observatories.

In this poster we will present the dax architecture, highlight some of the CIAO tasks that have been made available via dax, and discuss some of the pitfalls and limitations we encountered. We will also present some possible future directions for dax including additional analysis tasks and potential for remote analysis.

Architecture



The dax philosophy is 'Simple should be done simply."

This means not only should it be simple for users to do simple things, but the code to do it should be equally simple.

dax is a collection of scripts. It is a hierarchical ds9 analysis menu (ciao.ds9) and a suite of simple shell scripts that wrap CIAO tools.

The syntax of the ds9 analysis menu is simple but feature-rich. It provides basic UI elements: text-entry, radio/check-boxes, lists, key-bindings, etc. A comprehensive review of the analysis file syntax is beyond the scope of this poster; however we provide a simplified example showing a dax task, to get coordinates in various Chandra coordinate systems.

```
All Coordinates
*
menu
ds9_coord.sh $xpa | $text
```

A simplified version of the script 'ds9_coord.sh' is shown below

```
#! /bin/sh
ds9=$1

x=`xpaget $ds9 crosshair | awk '{print $1}'`
y=`xpaget $ds9 crosshair | awk '{print $2}'`
f=`xpaget $ds9 file `

dmcoords "${f}" op=sky x=$x y=$y mode=h verb=1
```

With these two simple pieces users can run the CIAO dmcoords which returns the coordinate transforms for the current crosshair location in a pop-up text box.

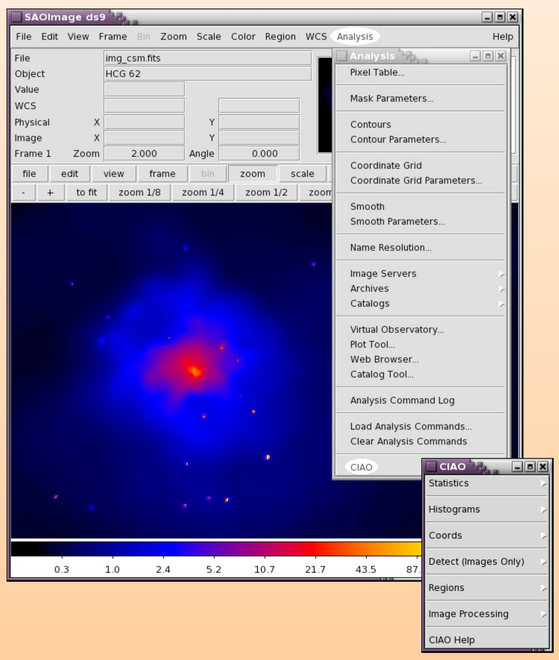
Loading Menu and Initialization

dax is loaded for CIAO users be default by means of a wrapper that uses the -analysis command line switch. While users can setup to have it loaded automatically via ds9 preferences, the path CIAO uses changes with each release and this was found to be the easiest for CIAO users.

By default, dax uses the ChiPS plotting package in CIAO to produce plots such as light-curves and spectra. In CIAO 4.3, we provide a start-up script that launches the ChiPS server when ds9 starts so that plotting happens more quickly and without the need for a terminal. This is a simpe tcl/tk script that is sourced at statup. If ChiPS is not installed, dax will use the BLT plotting packaged in ds9.

The CIAO ds9 wrapper script looks something like:

```
ds9 -analysis ciao.ds9 -source chips_startup.tk
```



Data Input to Analysis Task

The data being displayed in ds9 (most often in the current frame) are easily accessible to dax. ds9 provides several methods to access the data depending on the needs of the application. These are summarized below

$data

FITS image is piped to stdout

- WYSIWYG
- Only images
- If table is loaded then the image displayed is piped
  - Only minimal header information is passed along.
  - default is 1k x 1k (adjust under Binning menu)
- Easiest to use with image-in/image-out analysis tasks

$filename

Passes path, file-name, and extension to analysis task

- Access to tables or images
- Only applies to files loaded from disk. Cannot access files
  - retrieved remotely (SAMP, Image Servers, etc)
  - generated from analysis tasks
- Access to multiple blocks of data: eg GTI extensions
- Be wary of filter and extension selection syntax

$xpa

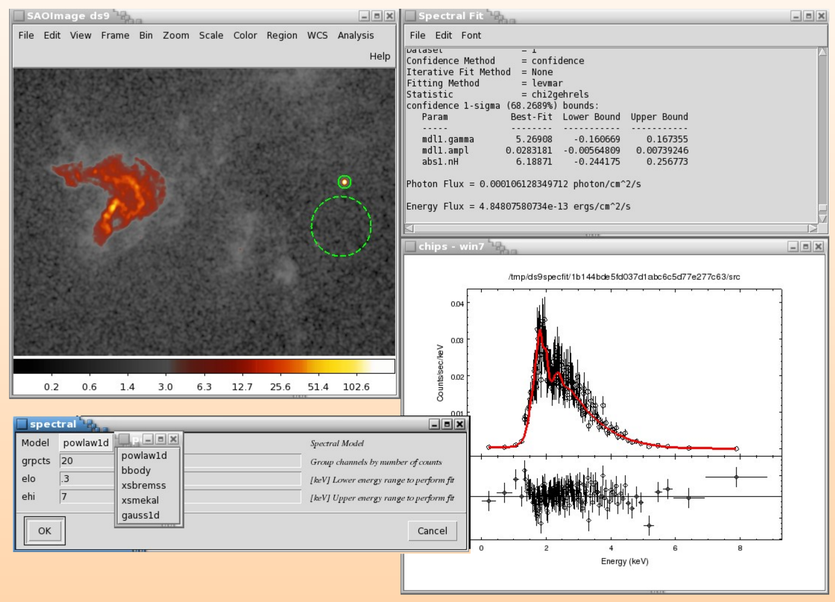
Passes the XPA access point name to the analysis task

- Can use xpa commands to access specific data elements.
- Race condition: non-locking, user can switch frames
- Provides the most flexibility and often requires the most code
- Multiple ds9's running at same time can produce unexpected results unless each is given unique name by user at startup via -title command line argument

There are pro's and con's to each method. Following the dax philosophy, most scripts make use of the $data and $filename macros as they are the simplest to use and work most of the time. The $xpa method is primarily reserved for those times when many parameters associated with the data are passed into the script or the task needs to be especially robust.

Possible Future Enhancements

Modeling and Fitting data is the next leap to take to get from data analysis to data reduction. For Chandra data, this often necessitates having access to an array of ancillary observation specific files and to a large volume of calibration data. This provides an entirely new challenge in data management. Below is an example proof-of-concept showing how to use the Sherpa fitting and modeling application via dax to fit a spectral model



Work is also underway to evaluate the costs and benefits of providing dax tasks as remote services. In addition to the UNIX shell, ds9 already supports a URL based execution model and VO standards such as the Universal Work Service REST bindings map well to the dax architecture.

```
$geturl(http://cxcserver/ciao_task?filename=$filename&xpa=$xpa...) | $text
```

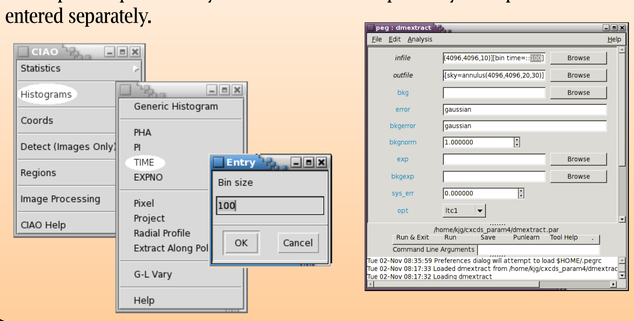
Task Selection

CIAO has over 100 tools and applications; dax makes only a few available. The following are some considerations that were made when selecting which CIAO tasks dax would enable:

- Typical run-time: GUI users are looking for quick feedback so tasks like the CIAO wavdetect tool which can take a long time to run are not included.
- Self contained input files: If a task needs auxiliary files (eg badpixel files or aspect solution) then its usefulness as a dax task is limited. As of ds9 6.1 a new analysis macro is available to selection additional files; however for multiple files this can stifle the GUI user experience.
- Mission independence: Many of the CIAO tasks are indeed mission independent; those that are Chandra-specific need to be well behaved for non-Chandra data and also not require arbitrary calibrations
- Output type: One of the dax challenges is how to return results in a useful format. If the output of a particular task is more often than not just used as input to another task then it would not be included in dax. The output should be a product upon which further analysis can be done, eg by dax.

User Parameters

dax takes a minimalist approach when faced with gathering input from users. By keeping the tasks small and very specific users do not need to know all the behind the scenes details; while at the same time similar tasks are grouped into single scripts. The PHA, PI, TIME, and EXPNO histograms all run a single dax script that calls the CIAO tool dmextract with the correct settings to create the requested plot. Many dax tasks do not require any user parameters to be entered separately.



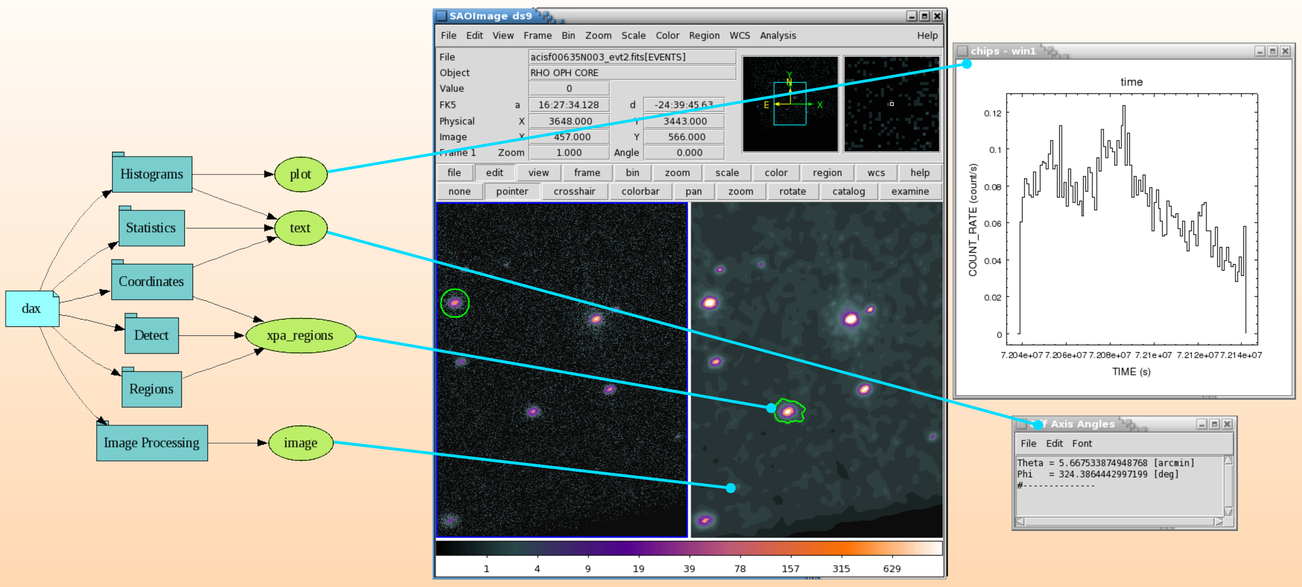
Special Considerations

ds9 is capable of displaying much more than simple 2D images from a single FITS file. For example it can now create mosaics from multiple files, 3-color (RGB) images, and display data from cubes up to 10 dimensions. All of these special data types can cause unexpected behavior in dax and other analysis tasks that are not expecting such diverse datasets.

When working with tabular data such as Chandra event files one has to be careful about filtering and binning syntax. ds9 uses the funtools syntax to specifying column filters and extensions. These are not always the same as CIAO virtual file specification nor the same as CFITSIO (and FTOOLS).

Regions are notoriously problematic. Different analysis systems use different syntax for shapes (box expressed as lower-left/upper-right, or center with x and y lengths) ; provide different shapes (eg, ds9's panda); and imply different intent when multiple shapes are drawn (shapes ANDed or ORed together).

Returning Results



ds9 supports 3 different output models: $data, $image, and $text. A fourth, $null, is available when the action of the analysis task is handles via other means. For example one dax task modifies the currently selected region to shift it to the data centroid. The regions are read and written via XPA. dax makes use of all these methods; though even when xpa is used to return results dax still uses the $text directive since it can also capture any processing errors.

dax tries to deliver data in the most useful format for quick evaluation at the same time trying not to leave the data "trapped" in ds9.

Conclusions

ds9 provides easy access to external analysis packages via its feature-rich analysis menu framework. dax uses this framework to give CIAO users access to many common analysis tasks. Using dax, CIAO users do not need to invest in learning the datamodel's filtering and binning syntax. Instead they can quickly jump into their analysis and get a feel for their data. Not all CIAO tasks are appropriate to be wrapped by dax. Those that do things such as apply event-by-event calibrations or produce intermediate analysis products are best left to users to run stand alone. Also tasks that take especially long to run are best left out of the GUI. Currently dax is a great tool to quickly explore ones datasets. More detailed analysis requires additional data products and calibrations. Future dax enhancements are being considered to access some of these additional data products and provide users with a small subset of the Fitting and Modeling capabilities in Sherpa. Finally as we move into the era of Cloud computing, many of the dax tasks are already well suited to be run remotely though data management challenges do remain.

This project is funded by NASA contract NAS8-03060 (CXC).