Astronomical Data Analysis Software and Systems IV
ASP Conference Series, Vol. 77, 1995
R. A. Shaw, H. E. Payne, and J. J. E. Hayes, eds.

Applying Public Access Programming Techniques To SAOimage

E. Mandel

Smithsonian Astrophysical Observatory, 60 Garden St., Cambridge, MA 02138

D. Tody

National Optical Astronomy Observatories, P. O. Box 26732, Tucson, AZ 85719

Abstract. This paper describes our application of the X Public Access (*XPA*) interface to the new version of *SAOimage*. *XPA* allows an Xt program to define named public access points through which data and commands can be exchanged with external programs. It makes possible the external control of the program's main functions, including image display, image zoom and pan, color map manipulation, cursor/region definition, and frame selection. It also supports "externalization" of internal algorithms such as file access and scaling. Finally, we describe how *XPA* is used to support user-configurable "quick-look" analysis of image data and bi-directional communication with other processes.

1. Introduction

Astronomical software development needs to adopt more "open systems" concepts and designs. Users increasingly want to extend existing software and to combine tasks from different systems in order to create their own heterogeneous research environments. Our software designs must reflect this increasing need for flexibility and extensibility without sacrificing functionality and without violating budget constraints.

The *SAOimage* display program has been very popular with astronomers over the past decade. In the age of "open systems", it needs updating to support extensible features. These features include providing an easy way to display arbitrary data file formats, control image display from an external process, and integrate analysis routines into the image display.

In order to update *SAOimage* efficiently and cost-effectively, SAO and NOAO will collaborate to develop a high-quality image display program, basing our work on the *ximtool* program developed by D. Tody for IRAF. *Ximtool* uses the NOAO widget server and "gterm" image widget to support customized graphical user interfaces (GUIs), image and line graphics, multiple frame buffers, user-defined color maps, region markers, etc. It provides a high-level image display programming interface as well as low-level access to gterm-image widget functions. Our aim is to layer open-ended functionality on top of *ximtool* in order to develop *SAOtng* (SAOimage: The Next Generation). In this way, SAO and NOAO can share development of image display, while satisfying our individual needs.

2. The X Public Access Mechanism

To add open systems features to *SAOtng*, we will utilize the X Public Access mechanism (*XPA*), described in Mandel & Swick (1994). *XPA* allows an Xt program to define named points of public access, through which data and commands can be communicated to/from external programs:

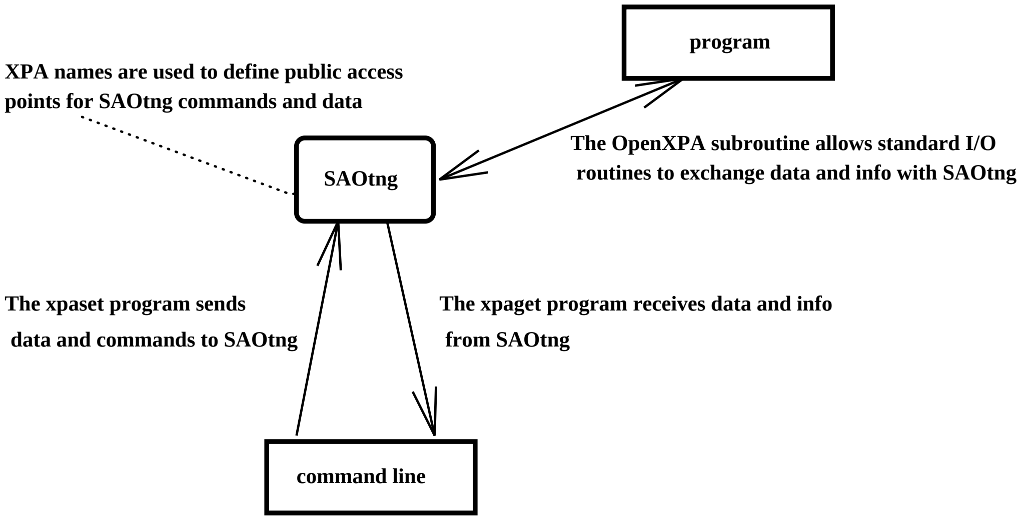

Figure 1. We have used *XPA* to extend the *ximtool* image server into an open-ended *SAOtng* display service that can cooperate with other processes and programs.

3. Controlling SAOtng

The standard *SAOtng* graphical user interface is a simple menu-based GUI that parallels the original *SAOimage* menu/command paradigm. It provides menus for image and frame selection, scaling, color maps, zoom/pan, regions, etc. All of these menu functions also are defined to be *XPA* public access points. Thus, *SAOtng* can be controlled directly through its menu-based GUI or externally through the *XPA* mechanism (see Figure 2).

For example, the *xpaset* program can be used to change the *SAOtng* scaling algorithm, and the *xpaget* program can be used to retrieve the current scale:

```
csh> echo "scale log" | xpaset SAOtng
csh> xpaget SAOtng scale
log
```

In fact, the *SAOtng* GUI uses *XPA* to send commands to itself. This implementation ensures that support for external control of *SAOtng* does not lag behind the internally-supported user interface: the GUI initiates commands as if it were just another external process!

4. External Support For File Access

A FITS file can be sent directly to *SAOtng* for display using *XPA*:

```
cat foo.fits | xpaset SAOtng
```

Other file formats are supported by writing two file access programs: a header access program to generate a FITS header of the full image, and a data access program to generate a FITS image of a specified data section (given the center, dimensions, and block factor). These access programs are defined at start-up time in a user-configurable ASCII file. To load an image, the programs are run externally (using the *system()* subroutine) to create FITS-format data for a specified data section, which are passed back to *SAOtng* for display using *XPA*. When a new file name is sent to *SAOtng*, the appropriate header access program is run to gather the overall image dimensions. Next, the data access program is run to extract the desired image section, which then is scaled to 8 bits and displayed. Finally, the data access program again extracts the data section and stores it as raw data for later use (e.g., re-scaling the image). Note that adding new formats does not require re-compilation of *SAOtng*.

5. Integration of Analysis Routines

Each file type can have user-defined analysis commands associated with it. These analysis commands are defined at start-up time by means of ASCII descriptions. The analysis commands associated with the currently displayed image are available for execution via an analysis menu. When selected from this menu, an analysis command is macro-expanded and then executed externally. Results can be displayed in a separate window or even can be sent back to *SAOtng*, i.e., a command can create an image and then send it to *SAOtng* for display:

```
$data | smooth ... | xpaset SAOtng "frame new"
```

In the example above, the "`$data`" macro is expanded into an *XPA* command that retrieves the FITS representation of the displayed image. The FITS image generated by the smooth program is piped into a new frame buffer of *SAOtng*.

6. Cooperation With Other Processes

XPA allows the re-use of GUI programs so that new systems can be built from existing high-level components. For example, the *SAOtng* "Load Image" menu option pops up the *XDir* program, a directory and file browser that also offers *XPA* services (see Figure 2). The *XDir* program then can be used to browse through directory trees using template filters. Double clicking on an image file

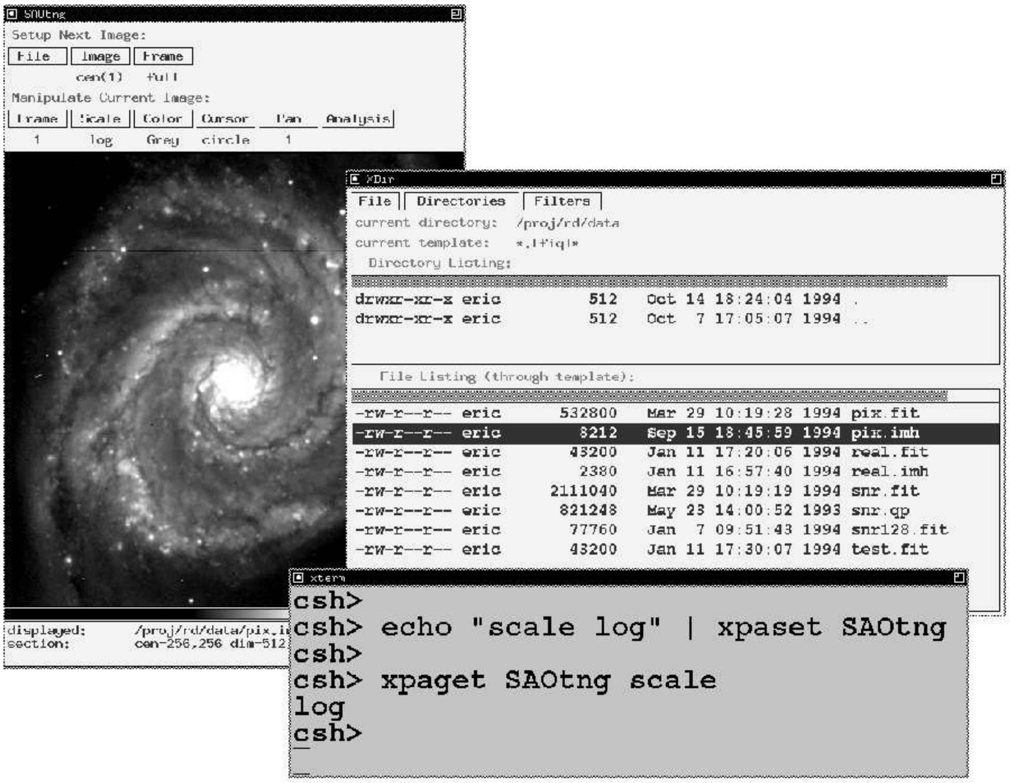

Figure 2. *SAOtng* has a simple menu-based GUI. It can be commanded externally (*lower right*) and can communicate with programs such as *XDir* (*upper right*).

causes *XDir* to send an *XPA* command to *SAOtng* to load the new image. Thus, *SAOtng* can make use of sophisticated *XDir* capabilities without linking them explicitly: GUI programs can be re-used in the same manner as subroutines!

7. Conclusion

SAOtng exemplifies a new type of astronomical analysis program that is extensible and that cooperates with other processes. It takes us closer to an era in which our heterogeneous analysis systems can work as an integrated whole.

Acknowledgments. This work was supported under NASA contracts to the *IRAF* Technical Working Group (NAGW–1921), the *AXAF* High Resolution Camera (NAS8–38248), and the *AXAF* Science Center (NAS8–39073).

References

Mandel, E., & Swick, R. 1994, PASP, 96, 198