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

SAOTk and TclXPA: Tcl/Tk Extensions for Astronomy

W. Joye, E. Mandel

Smithsonian Astrophysical Observatory, Cambridge, MA 02138

Abstract. SAOTk is an integrated set of Tcl/Tk canvas widgets for astronomical imaging and data visualization. The widget set is composed of the Frame, Panner, Magnifier, and Colorbar widgets. In addition to "classical" support for imaging FITS data, manipulating colormaps, region marking, coordinate readout (including WCS), etc., SAOTk widgets also support arbitrary image scaling and rotation, advanced PostScript printing, truecolor graphics support, and image mosaics.

TclXPA is a Tcl package that implements both the client and server aspects of the XPA messaging system within Tcl. TclXPA allows a Tcl program to exchange data and commands with Tcl programs, Xt programs, or Unix programs.

SAOTk and TclXPA are being used in a wide variety of astronomical applications. These applications include general visualization and analysis, real-time instrumentation and calibration, and interactive modeling. These Tcl extensions can be utilized in any standard Tcl/Tk environment to build custom data analysis and visualization applications.

1. Introduction

A common denominator among astronomical software is the need to display and manipulate FITS data. In many applications, such as proposal submission tools, survey and catalog tools, mission planning, instrument calibration, or general analysis, users need to visualize FITS data. But to develop high-quality astronomical visualization software, one almost must make a career of it. The developer must become an expert on world coordinate systems, graphics hardware, 2D and 3D computer graphics concepts, the FITS file standard, legacy software and file formats, and interprocess communications.

One of the strengths of Tcl/Tk is the concept of *comprehensive, reusable, industrial-strength* widgets. Examples include the Canvas and Text widgets. Because such widgets are complete and self-contained, they can be integrated into fully-featured applications with minimal buy-in. The SAO/HEAD Research and Development group has developed a set of *comprehensive, reusable, industrial-strength* Tk widgets for astronomical imaging and data visualization, called SAOTk. We also have developed interprocess communications support that encompasses the Tcl environment, called TclXPA.

2. Goals

The SAOTk and TclXPA projects have had several goals in mind during development. The first goal is minimal buy-in. This means that there should be no installation required. No auxiliary files, libraries, or scripts should be needed. The widgets and libraries should be easy to use and encourage rapid prototyping. Finally, it should be possible to build complete, stand-alone applications using these tools.

The second goal of these projects is support for a wide variety of hardware platforms. Newer machines provide fast CPUs, vast amounts of memory, and truecolor graphics. Older equipment consist of slower CPUs, limited memory, and pseudocolor graphics. The widget set must allow users to take advantage of the new capabilities, while not excluding the older, more limited equipment.

The third goal is cross-platform compatibility. The widget set must run on a wide variety of software platforms, including Solaris, Linux, and Windows.

Finally, the fourth goal is support for very large data sets. With a number of new mosaic ccd ground-based instruments and space-based observatories coming on-line in the near future, data sets of size .5GB to 1GB and greater are now becoming common.

3. SAOTk, TclXPA and Tcl/Tk

When designing SAOTk and TclXPA, it was decided that no alterations of Tcl/Tk should be necessary. These packages use the existing Tcl/Tk API. The SAOTk widgets are implemented as Tk Canvas widgets because the latter provides support for both image data and line graphics, flexible layout management and event processing. The TclXPA package is implemented as Tcl commands. Both packages conform to Tcl/Tk package initialization procedures.

In applications built with SAOTk, TclXPA, and Tcl/Tk, Tk is used to define the GUI, while SAOTk and TclXPA (along with other widget sets, such as BLT and TkTable) provide the functionality. Stand-alone applications are created by use of ET or Mktclapp.

4. SAOTk: Tk Imaging Widgets

The SAOTk widget set consists of four widgets: Frame, Colorbar, Magnifier, and Panner. SAOTk provides support for 8-bit pseudocolor and 8-bit, 16-bit, and 24-bit truecolor environments.

The Frame widget supports display of FITS images, FITS binary tables, FITS mosaic images, and raw data arrays. The FITS keywords `BLANK, BZERO,` and `BSCALE` are supported. Data may be accessed via the file system, standard I/O, or Tcl Channels. Memory management via mmap, allocated memory, or shared memory is also supported.

The Frame widget allows arbitrary rotation and zooming of the displayed data. Render time is based on the size of the window, not the size of the data. Scaling, byte-swapping, and binning are all implemented *on the fly*.

Support for world, image, physical, and IRAF mosaic coordinate systems is provided. The widget supports standard region shapes (circle, ellipse, box, etc.), region properties (color, font, text, etc.) and region file formats (SAOtng, PROS, SAOimage). True PostScript printing is provided with both level 1 and level 2 drivers, with options such as paper size and print resolution.

The Colorbar widget manages the color environment. It has 20 built-in colormaps, and support for external colormap formats from SAOtng, SAOimage, and RTD/Skycat. The Panner widget provides *panning* functionality using a panner bounding box, along with an image/wcs compass. The Magnifier widget provides a close up view of the current cursor position and can be configured with a pixel cursor.

5. TclXPA: Public Access to Data and Algorithms

The XPA messaging system provides seamless communication between many kinds of Unix programs, including Tcl/Tk programs and X programs. It also provides an easy way for users to communicate with XPA-enabled programs by executing XPA client commands in the shell or by utilizing such commands in scripts. Because XPA works both at the programming level and the shell level, it is a powerful tool for unifying any analysis environment: users and programmers have great flexibility in choosing the best level or levels at which to access XPA services, and client access can be extended or modified easily at any time.

A program becomes an XPA-enabled server by defining named points of public access through which data and commands can be exchanged with other client programs (and users). Using standard TCP sockets as a transport mechanism, XPA supports both single-point and broadcast messaging to and from these servers. It supports direct communication between clients and servers, or indirect communication via an intermediate message bus emulation program. Host-based access control is implemented, as is as the ability to communicate with XPA servers across a network.

XPA client and server subroutines from the standard XPA library can be used to add C-based XPA support to Tcl/Tk programs. Once an XPA service has been defined, it can be added to the Tcl event loop (either vwait() or the Tk event loop) simply by calling the XPATclAddInput() routine before entering the loop. However, the TclXPA package goes beyond support for C-based XPA calls within Tcl. This package allows one to write XPA servers and to make XPA client calls within the Tcl environment, using the Tcl language directly. When the TclXPA package is loaded, Tcl versions of all XPA routines are available for communication with other XPA-enabled programs, including X, Tcl, and Unix programs. The TclXPA interface has been designed to match the Unix XPA interface as nearly as possible, so that programmers can use one standard interface for interprocess communication.

6. Applications

One of the first applications to be developed using the SAOTk and TclXPA packages is DS9. DS9 is an astronomical imaging and data visualization application. DS9 supports FITS images and binary tables, multiple frame buffers, region manipulation, many scale algorithms and colormaps, and easy communication with external analysis tasks. It is highly configurable and extensible.

DS9 is a stand-alone application. It requires no installation or support files. Versions of DS9 currently exist for Sun Solaris, Linux, and Windows. All versions and platforms supports a consistent set of GUI and functional capabilities.

DS9 supports advanced features such as multiple frame buffers, mosaic images, tiling, blinking, geometric markers, colormap manipulation, scaling, arbitrary zoom, rotation, pan, and a variety of coordinate systems. DS9 also supports FTP and HTTP access.

The GUI for DS9 is user-configurable. GUI elements such as the coordinate display, panner, magnifier, horizontal and vertical graphs, button bar, and colorbar can be configured via menus or the command line.

7. Conclusion

More information about the *SAOTk* widgets, *DS9*, and *XPA* (including our latest software offerings) can be found on the World Wide Web at:

http://hea-www.harvard.edu/RD/

Acknowledgments. This work was performed under a grant from NASA's Applied Information System Research Program (NAG5-3996), with additional support from the AXAF Science Center (NAS8-39073).