GUI Tools in AIPS++

Paul Shannon

National Radio Astronomy Observatory[1]*, 520 Edgemont Road, Charlottesville, VA 22903*

Abstract. For the astronomical user AIPS++ currently provides the Glish interpreter, a growing number of Glish analytical clients written in C++, and a handful of Motif and (recently) Glish/Tk programs. These clients and programs are loosely coupled to the interpreter, and thus to each other, using Glish as a "software bus". Messages pass back and forth among these various Glish clients, creating the interactive feel of an integrated application. This paper describes a few AIPS++ GUI tools.

1. Introduction

In AIPS++, Table objects provide the fundamental storage mechanism for data. A Glish Table client allows for easy access to the data in a table; once the table data is in glish, it can be manipulated by the user, and sent to various clients for specialized manipulation (fitting, for instance) and to GUI clients for display.

The table browser is one of the GUI clients. It is rather like a conventional, graphical spreadsheet application, which displays and edits the cells of a table. AipsView (see Crutcher, 1996) is another, and a vector plotter is a third. All three of these provide (or will soon provide) access to their most important capabilities via a simple glish functional interface.

Glish is, in essence, the glue which connects these clients. We have just started to use a binding of the popular Tk widget set to Glish. This makes it possible to glue the clients togther with a pleasing GUI, whenever the task at hand is susceptible to that approach.

2. Background: The Glish Table Client

AIPS++ Tables are quite complicated. A number of storage managers can be used – providing optimized access or minimal storage requirements for different classes of data; key/value pairs may be attached to the table as a whole, or to any column, and the value of the keyword may be of any supported type: scalars, arrays, keywords, or even sub-tables. Fortunately, the Table client hides this complexity from view, and it is very easy to get scalars and vectors (and

[1]The National Radio Astronomy Observatory is a facility of the National Science Foundation operated under cooperative agreement by Associated Universities, Inc.

higher-dimensioned arrays) from the table with simple Glish function calls. For example:

```
table := open_table (name);
x := get_column (table, some_column_name);
y := get_column (table, some_other_column_name);
plot_xy (x, y, title);
```

3. Vector Plotting

We are currently using a commercial Motif widget (XRT/graph) as the basis of this GUI client. This widget lies at the heart of a C++ program, and its capabilities appear within Glish as simple function calls. (Even if we change widgets, perhaps to a public domain graphing widget, this glish interface will remain the same).

```
plot_xy (x, y, dataSetName, style)
plot_xy2 (x, y, dataSetName, style)
plot_y (y, dataSetName, style)
plot_y2 (y, dataSetName, style)
time_y (x, y, dataSetName, style)
time_y2 (x, y, dataSetName, style)
append_xy (datasetNumber, x, y)
clear ()
marker (x:y:none)
number_of_selections ()
get_selection ()
get_x (dataSetNumber)
get_y (dataSetNumber)
set_point_size (dataSetNumber, size)
set_point_color (dataSetNumber, color)
set_point_style (dataSetNumber, style)
set_line_color (dataSetNumber, color)
set_line_width (dataSetNumber, width)
set_line_style (dataSetNumber, style)
set_x_axis_label (label)
set_y_axis_label (label)
set_x_scale (min, max)
set_y_scale (min, max)
set_legend_geometry (newGeometry)
legends_off ()
legends_on ()
query_data ()
query_styles ()
reverse_x ()
reverse_y ()
set_printer (printerName)
ps_print ()
ps_print_to_file (filename)
```

4. Browsers

There are currently three different data browsers in AIPS++: one for tables, one for arrays (of any dimension), and one for keywords. The table browser is most commonly used of the three; arrays or keyword sets which are embedded in tables can be examined by popping up the appropriate browser. The browsers are presently being revised in a modest way, to allow them to be used to modify (as well as examine) data, and to provide a sensible Glish interface.

The browsers are all built around XbaeMatrix (currently version 3.8, we will soon move to version 4.0) – a Motif widget in the public domain. In addition to the modest revisions mentioned above, it is essential that we modify (or fool) the widget so that it can page through a large AIPS++ table, rather than reading the entire table into memory.

5. Glish/Tk

The Tk widget set is often used to provide GUI capabilities for public domain languages (Python and Scheme, for example). The AIPS++ project has a provisional – but very pleasing – binding of Tk to Glish, in which every Tk widget appears as a standard Glish agent, capable of sending and receiving events (see Schiebel, 1996). This allows for the speedy and robust development of Glish scripts which present a graphical interface to the user – buttons, text entry, listboxes, menus – and where widget events – pressing a button, entering text, selecting one or more items in a listbox – result in messages going out to other Glish clients on the software bus. The user of one of these scripts is not aware that a Glish script is running: all the user sees is a conventional Motif-like program on the screen.

In our first use of Glish/Tk we created a GUI tool for filling data from FITS GBT logging files into AIPS++ tables (see Garwood, 1996), and for interactive selection and plotting of table columns. We also have a protoype on-line help system for AIPS++, and one of the viewers is written Glish/Tk. (A WWW browser and dumb terminal viewing are also supported.) We expect that Glish/Tk will see a great deal of use in the months to come.

References

Crutcher, R. W. 1996, this volume
Garwood, R. W. 1996, this volume
Schiebel, D. W. 1996, this volume