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.

A Graphical User Interface for a Development Environment (GUIDE)

A. Bhatnagar

The Aerospace Corporation, PO Box 92957-M2/259, Los Angeles, CA 90009

Abstract. "GUIDE" (Graphical User Interface for a Data Environment) is an interactive IDL widget-based package that enables users to retrieve selected sets of satellite instrument data and perform various types of graphical analyses on them, without writing a specialized program. GUIDE simplifies the task of laying out multiple plots on a page by (1) using a page-editor to define and laying out multiple "panels" which contain the "plot objects", and (2) integrating data retrieval functions.

1. Introduction and Design Concept

GUIDE was developed in support of the POLAR CEPPAD (Comprehensive Energetic Particle Pitch Angle Distribution) experiment. Its fundamental software design was created with the following goals and requirements in mind:

- The system should require little or no knowledge of the underlying application software (IDL).
- The software routines should be as modularized as possible.
- The software should be machine independent to allow for portability and a consistent user interface. It should also be device independent to permit various types of output, such as publication-quality hardcopies or slides.
- The analysis and presentation routines should be time-span independent.
- The system should be capable of accessing archived event data given a time interval reference (i.e., start/stop time, orbit, date, etc.) as well as energy levels, channel number, etc.
- The system should offer the capability to retrieve, analyze and compare data from ancillary sources including ephemeris, geomagnetic indices, and data from other experiments.

2. Architecture and Function

A "page editor" was devised to create and modify three components: the Page Description, the Panel Description, and the Plot Object. These components characterize the layout and content of all the plots grouped within the given

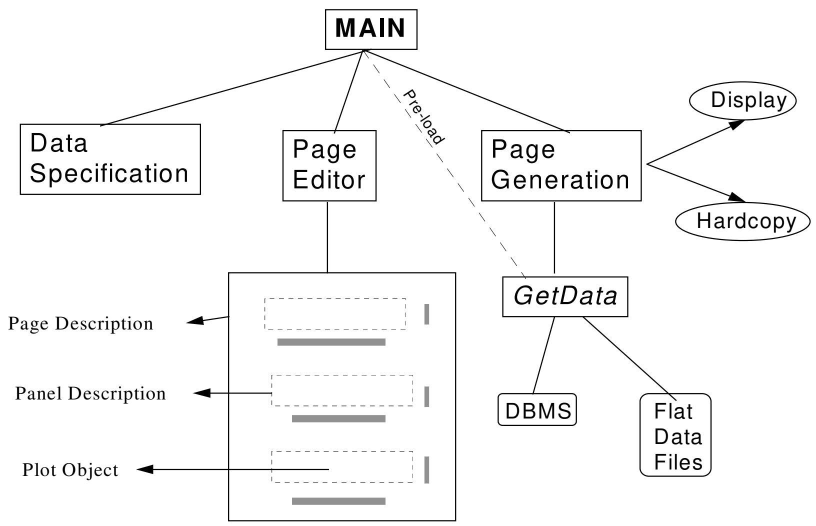

Figure 1. System design.

page. Figure 1 illustrates this subsumptive architecture with respect to other functional aspects of GUIDE, described below.

The Page Description File contains information pertaining to the overall number, size, and relative location of panels for the given page, but not the actual contents of those panels—that information is in the Panel Description Files. This natural separation facilitates a more flexible system by allowing panels to be reused.

The Panel Description File contains all the necessary content descriptors for that particular panel including the name of the routine generating the plot for that particular panel and the input parameters associated with it. The input parameters required by the plot object are divided into two categories—fixed and run-time. The distinction is based upon the stage at which their values are specified or altered. Fixed parameters include plot labels and titles, the number of series (lines) in a plot, or minimum and maximum flux values in a spectrogram. Run-time parameters, on the other hand, might include items whose values are typically altered for each run such as the start and stop times for a given event analysis.

Finally, the Plot Object is a procedure, or collection of procedures, containing the actual (IDL) code responsible for generating the specified plot, image, or any other pre-defined graphical object. At run-time, this module is called by the main GUIDE processing routine after it has set up system environment variables that define, for instance, window or viewport boundaries.

2.1. Page Generation

Using point-and-click mouse functions, GUIDE steps the user through the process of creating and laying out a Page Description File embedded with pointers describing the contents of each of its panels (see Figure 2). Once this is done and the user has specified values for the run-time parameters (see Figure 3), the user selects the menu option “Go!”. The main GUIDE routine interprets

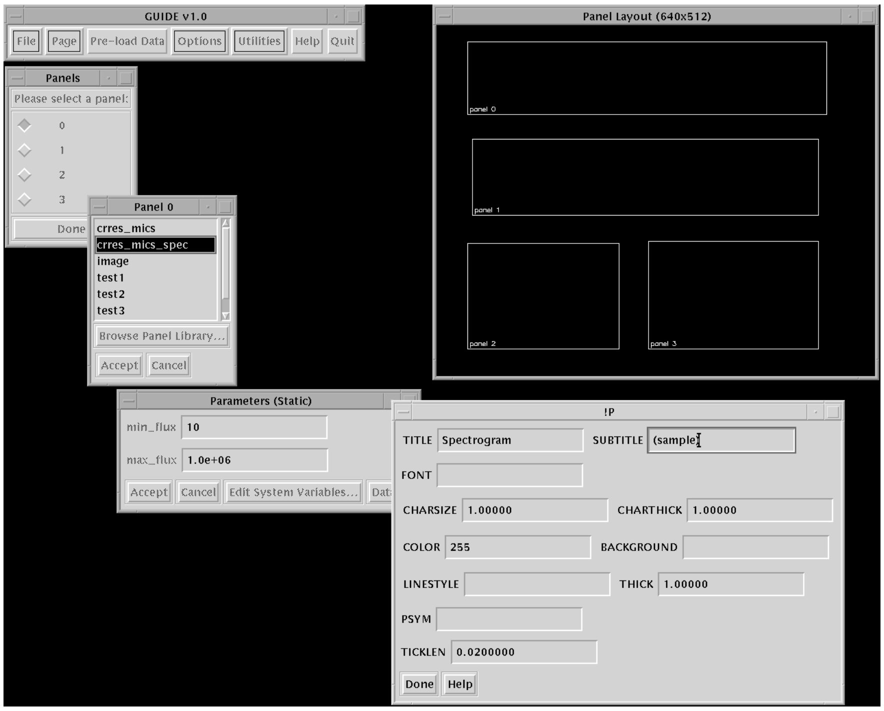

Figure 2. Creation of a new Page File.

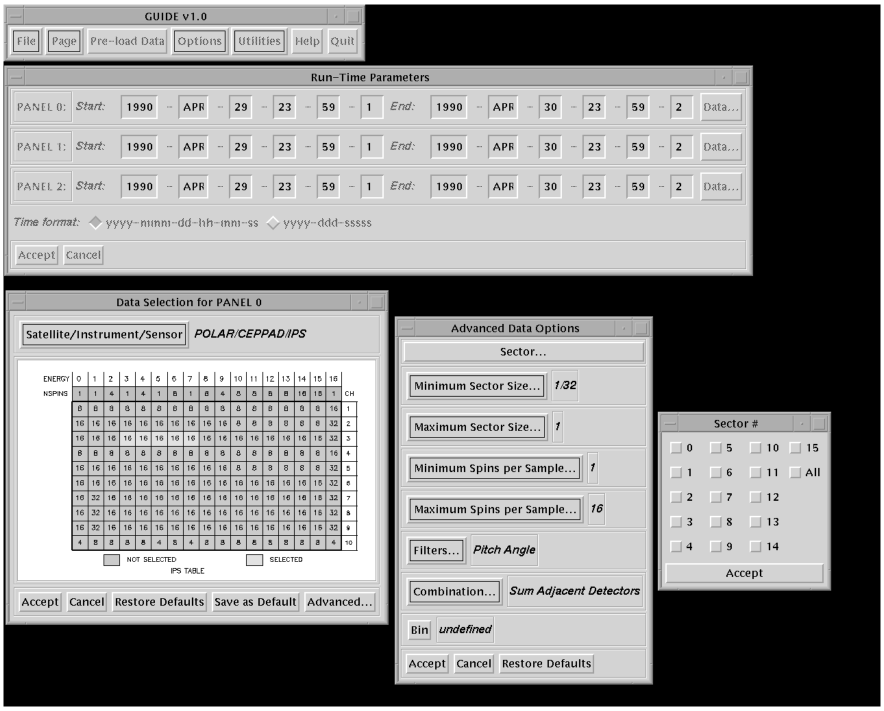

Figure 3. Run-time parameter specification with optimal filters.

the contents of the Page File, and for each panel (1) gets the location, and size of each panel from the Page Description File, (2) extracts the parameter list, parameter values, and plot object name associated with the current panel from the Panel Description File, (3) sets the plot viewport region based on the panel sizes, (4) invokes the appropriate plot object with the corresponding parameters to retrieve and process the pertinent data and display the plot, and (5) returns, restoring the default system environment.

2.2. GetData

In step (4) above, the plot object routine will typically make calls to a GetData procedure. These external procedures act as an application program interface (API) to the satellite/instrument specific dataset or database. For every distinct data type, there would be a separate GetData procedure which handles all the retrieval and processing specific to that data type. GetData is responsible for returning the data to common blocks accessed by the display routines, after reduction and filtering steps.have been performed. If no data match the selected criteria, or in the event of some other error condition, GetData returns a message to the calling program, the user is notified, and the generation of that panel ceases. Decoupling GetData from the GUIDE framework adds to the generality of the overall system by allowing GUIDE to transparently support multiple data types. GetData may be developed off-line and independent of GUIDE. There are, however, two basic input/output constraints imposed on GetData: (1) GetData must accept its inputs in a format pre-defined by GUIDE and its plot object routines, so that parameter lists match, and (2) the structure and location of the common blocks filled by GetData must be consistent with those utilized by GUIDE's plot object routines.

The user may also use the GUIDE menu option of "Pre-loading". This is essentially a direct top-level call to GetData for the purpose of verifying the existence of data within a given period and having it placed in common blocks so that it is readily available when needed.

3. Implementation Issues

GUIDE was developed using IDL 3.5.1 and uses the EXECUTE command to interpret the contents of the Panel Files at run-time and to call the Plot Objects. This facilitates accommodating new Plot Objects: (1) regardless of whether the code is all written in IDL or contains embedded CALL_EXTERNAL calls to C or FORTRAN images, all graphical output is done by IDL commands, (2) all Plot Object Files in the current directory are automatically added to the list used in building the "Browse Panel Library" widget, (3) when a user selects the new Plot Object in the panel definition stage, GUIDE automatically parses the Plot Object File and constructs the custom input widget, eliminating the need for hard-coding input widgets, and (4) GUIDE takes the parameter values provided by the user, stores them in the Panel Definition File, and passes them to the given Plot Object at run-time.

Acknowledgments. This work was supported under NASA Contract No. NAS5–30368 and through the co-development efforts of members of the Space Sciences Department of the Aerospace Corporation.