444444444444444444444444444444444444444444444444444444444444444444444444444

NIRSPEC

UCLA Astrophysics Program **U.C. Berkeley** **W.M.Keck Observatory**

444444444444444444444444444444444444444444444444444444444444444444444444444

Tim Liu

January 29, 1996

NIRSPEC Software Design Note 5.01
Graphical User Interface

1 Introduction

NIRSPEC is controlled through a software user interface which provides both window-style graphical input and command-line input. The graphical user interface (GUI) is the primary control interface that will be used by most users, whereas the command-driven interface provides scripting capability that may be more efficient for experienced users.

2 Requirements

The NIRSPEC GUI will be an X-Window application. It is designed to have the look and feel of the HIRES and LWS GUIs which are implemented with a click-on-icon/pop-up-menu style. The requirements for the GUI are that it should be intuitive, user-friendly, and robust. The following basic functions and features are required for the GUI:

! instrument setup
! observing parameter setup
! instrument status display
! exposure status display
! DCS status display
! input parameter check, software interlocks, and automatic timeouts
! exposure abort and motor motion abort
! engineering interface which provides various diagnostic tools

Additionally, a graphical spectral format simulator and an exposure time estimator will be developed for the GUI. The simulator helsp the user in positioning the detector array on the echelle format to ensure the array covers the desired spectral range. The exposure estimating program determines the integration time needed to reach a given signal-to-noise under various instrument configurations.

3 Interface/Application Model

The user interface is a graphical representation of the application to the user. It acts as an intermediary by handling the dialog between the user and the underlying application as shown in

Figure 1. The NIRSPEC software will perform many complex tasks which require large sizes of code for both the application and the user interface. To handle the connections between the application and the interface in an efficient manner that simplifies development, maintenance, and revision, the application code will be separated from the interface code as much as possible. The connections are handled by discrete modules. The application talks to the instrument via keywords (see Software Design Note 6.00).

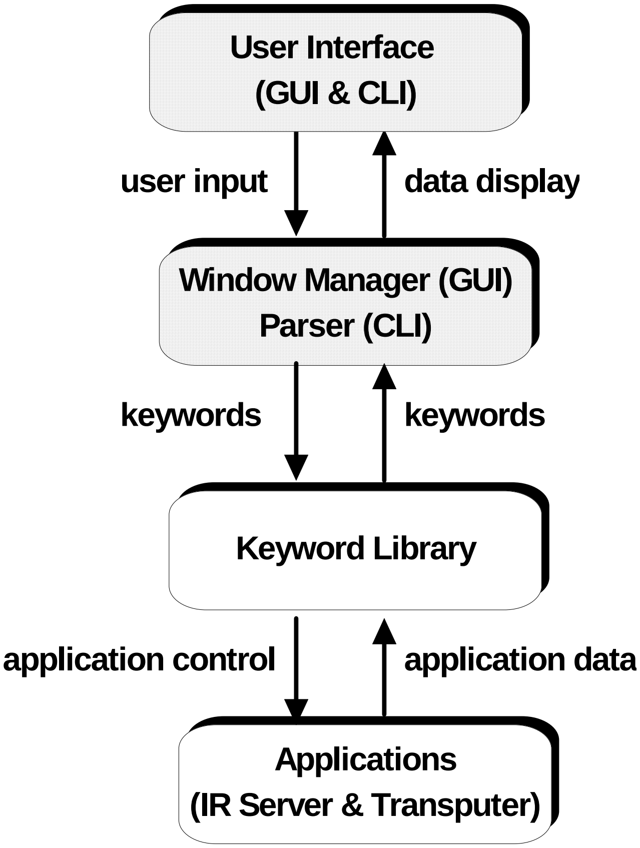

Figure 1 User interface control flow

4 Screen Layout

The current NIRSPEC design assumes a single computer monitor to be used for running both GUI and quick-look. On the Sparcstation screen, the GUI consists of two graphical windows: an instrument control console and an observer’s console. The instrument control console has an instrument cartoon as the user control panel, while the observer’s console provides observing parameter setup, exposure status display, and a command line entry. Moreover, the workstation screen will be covered by three more windows for quick-look: slit-view camera image display, spectrometer image display, and spectral plot. Some screen space should also be reserved for displaying other programs such as telescope DCS. The screen layout of the GUI and the quick-look is shown in Figure 2.

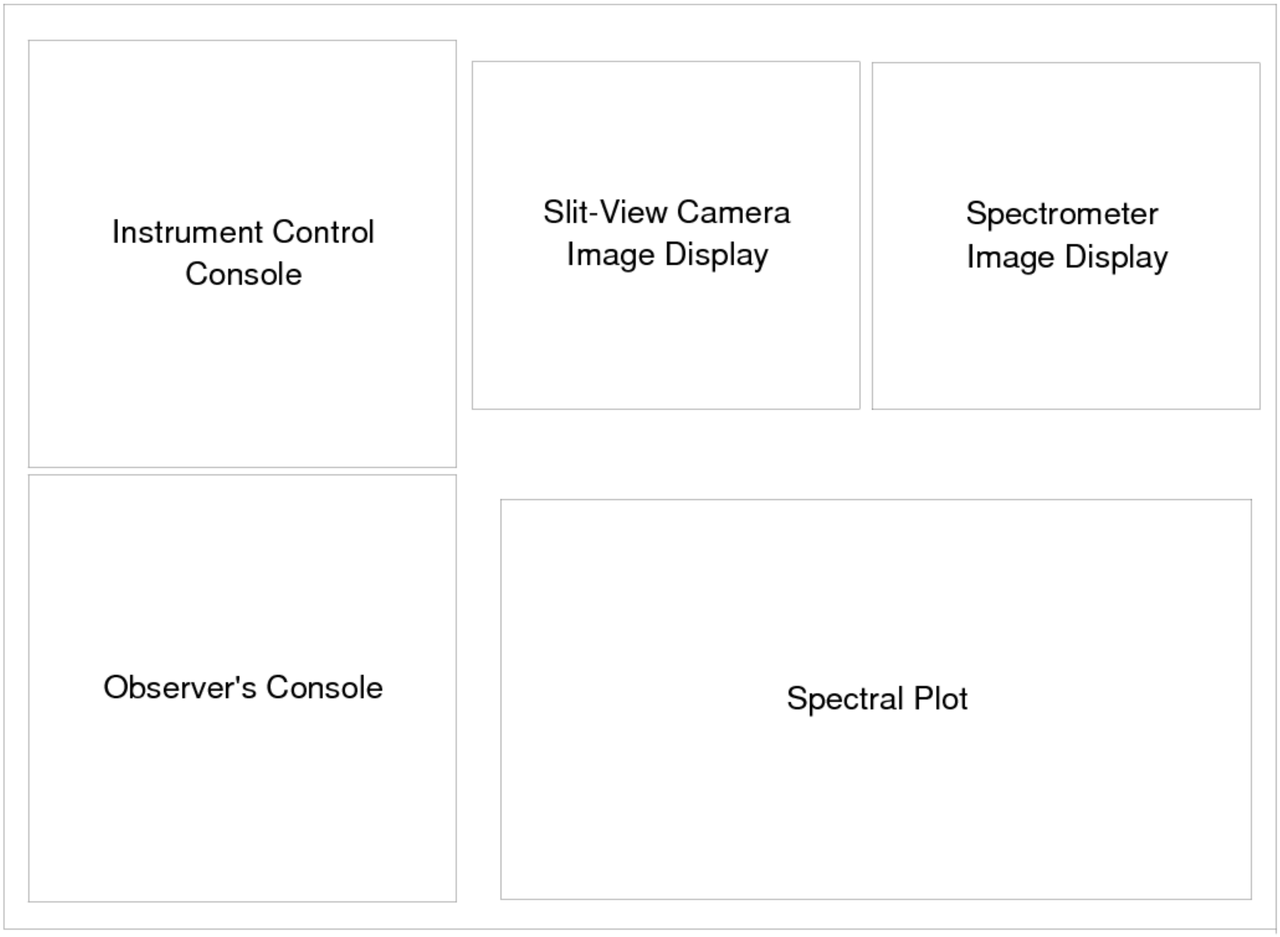
Instrument Control Console
Slit-View Camera Image Display
Spectrometer Image Display
Observer's Console
Spectral Plot

Figure 2 GUI Screen Layout

5 Programming Tool

Many software programming tools are provided by vendors to facilitate the development of GUIs. The Keck Software Coordination Committee (SCC) recommends DataViews, an X-Window tool from V.I. Corporation which is capable of building complex graphical interfaces. DataViews has been used for several user interfaces on Keck such as the LWS GUI xlws.

DataViews consists of two separate tools: DV-Draw and DV-Tools. DV-Draw is a menu-driven graphical editor that lets one create complete graphical interfaces, from designing screens to defining user interactions. This multipurpose interactive editor integrates geometry, graph, dynamics, input, data sources, and prototype rule editing. The drawing process requires no programming. DV-Tools is a library of graphics routines that builds the structure and control flow of the user interface. With this tool, one can integrate DV-Draw views into powerful graphical user interfaces for various applications. DataViews is installed on the Sparcstation in the directory /crab/home1/dv. For a quick guide to learning DataViews, see the Appendix in this design note.

6 GUI Design

Designs of the Instrument Control Console and Observer's Console have been produced using DV-Draw as shown in Figures 3 and 4.

6.1 Instrument Control Console

The dominant feature in this console is the instrument cartoon which draws all of the instrument components that require the user's interaction. They include: calibration lamps, image rotator, filter wheel, slit-view camera, slit wheel, echelle, grating, and detector. Each of these components has a push button and a status display box. When one wants to access to a component, like changing a filter position, he/she clicks on its button and a selection menu will pop up. The status display box will be updated once a new selection is made. The instrument console also provides several pull-down menus like Setup, Engineer, and Help. The Setup menu has various instrument configuration and setup functions, such as default instrument setup, configuring instrument from a previously stored setup file, etc. The Engineer menu provides various engineering diagnostic tools to exercise sub-assemblies like routines to manipulate stepper motors. The Help menu will provide on-line help for the program. The Quit button quits both Instrument Control Console and Observer's Console.

6.2 Observer's Console

The Observer's Console has two display panels: Current Exposure and Next Exposure. The Current Exposure panel shows the observing parameters for the on-going exposure and its integration status. One can abort the integration by clicking the Abort button. During the integration, one can set up the next observing sequence from the Next Exposure panel. A formal or test exposure is started by clicking either the Go or Test button. At the bottom of the console is the command-line entry window. Three pull-down menus: Setup, File, Options, are also available at the top of the console window. The Setup menu provides various observing setup functions, like setting up image directory. The File menu has functions to handle image files, like FITS BITS parameter, automatic file overwrite. The Options menu will contain some utility functions.

.

7 Program Structure and Control Flow

The graphical views created by DV-Draw are manipulated by the programmatic modules of DV-Tools to build the structure and control flow of the interface program. The following model program shows the typical structure of the NIRSPEC GUI program:

```
Initialization
   Initialize DataViews environment
   Initialize application environment
   Set up windows and wait screens
      Load views and windows
      Lay out and create drawports within windows
   Initialize and open data sources, rebind buffers as necessary

   Set up event handling
      Set the event filters
      Posts requests for control loop callbacks
   Draw first displays

Control loop
   Handle data
      Gather any new data from application
      Process data
         Update application based on data
         Update dynamic objects displayed in the interface
   Handle events
      Gather any events
      Process events
         Update application based on event
         Update interface based on event

Termination
   Destroy structures that have been set up
      Destroy views and drawports
      Close screens and data sources
   Terminate DataViews
   Clean up application
   Exit
```

Appendix: A Quick Guide to Learning DataViews

(I) Setup

In order to run DataViews programs from any directory, you need to initialize the DataViews environment first by typing

```
% source /crab/home1/dv/DVinit
```

You can add this command line to your .login file so that the initialization is automatically done when you log in.

(ii) Run Demo

DataViews provides demos that include the following features: Application Demos, Feature Demos, DataViews Tutorial and DV-Draw. To run the demo, type

```
% RunDemos &
```

Follow the instruction in the demo program to go through the entire demos once started.

(iii) Learn DV-Draw

To run DV-Draw, type

```
% DVdraw &
```

You can use the "DV_Draw Step-Through" document from DataViews to learn the features of DV-Draw, starting with simple features and progressing to more complex ones. For more complete coverage of the DV-Draw features, refer to the "DV-Draw User's Guide" document.

(iv) Learn DV-Tools

The starting place is the document "DV-Tools Getting Started" which introduce you to DV-Tools concepts and programming techniques. The document uses six example programs to help you learn DV-Tools basics.

Before running any of the examples, you should copy the entire contents of the example programs directory into your own directory. To do this, first make your own example program directory from your home directory:

```
% mkdir dv/examples
```

Then go over to the DataViews example program directory:

```
% cd /crab/home1/dv/DVtools/examples/programs
```

Next, you create a tar file of the entire directory, including subdirectories:

```
% tar -cvf [your home directory]/dv/examples/tarfile .
```

To restore the tar file, go to your own example program directory and type

```
% tar -xvf tarfile
```

Now you have copied the contents of the example program directory to your own area. You can modify, link and run the example programs by going through the "DV-Tools Getting Started" document. Consult "DV-Tools User's Guide" for more functions of DV-Tools.

In addition, DataViews provides a DV-Tutor on-line tutorial which includes the DV-Tools Coding Lessons section. Use the "DV-Tutor Lessons Guide" document to learn the on-line lessons. You should make your own copy of the directory tree /crab/home1/dv/DVtutor/lessons.