James Larkin **Jan 13, 1999**

NIRSPEC Software Programming Note 01.01
Software Overview

1.0 General Description

The NIRSPEC software consists of many individual programs written in several computer languages. The packages can generally be grouped into three categories: the server, hardware control and the clients. The server is a C program that has no direct user interface, but does use the KTL library (developed by CARA) of Remote Procedure Call (RPC) routines to allow flexible interconnections with other programs. This allows other packages (called clients) like command line interpreters (CLI's) and graphical user interfaces (GUI's) to connect and communicate with the server, even across the internet. These connections are bi-directional and servers can become clients of each other. This allows elaborate networks of interconnected servers to exist, and it supports efficient communications between systems such as the instrument, the telescope drive and control system (DCS), actuator control system (ACS) and even the adaptive optics system. The server also maintains a link via the SCSI port to the hardware system. The server and associated keyword library are detailed in NSPN 07.00 and NSPN 06.00.

The hardware is controlled by a set of transputers on custom boards within the electronics system. They handle all levels of hardware control including: array clocking and data acquisition, motor moves and status, and temperature monitoring and control (ALADDIN only). The transputers support parallel processing via the OCCAM programming language. OCCAM and transputers are detailed in programming documents: NSPN 08.00, NSPN 17.00, NSPN 18.00, NSPN 20.00, NSPN 21.00 and NSPN 22.00.

The clients are a varied set of programs that serve as an interface between the server and the user. The diagram on the next page shows all of the dedicated NIRSPEC clients and the overall connectivity of the software. For normal operation of the instrument at the telescope, it is anticipated that all of these clients will be running simultaneously, but not on the same computer or even for the same user. The three primary clients that the user will have sole and complete control of are: the dataviews gui which can fully control the instrument at the single mechanism or exposure level and which displays the current instrument status; the quicklook gui which displays the latest images and can be used for simple data reduction; and the echelle format simulator (EFS) which fully controls the instrument through scripting. A secondary set of user gui's are: the data reduction pipeline (DRP) which reduces spectral data in real time; the DRP's version of the quicklook for displaying reduced data; the ximrot and watch_imrot programs that control the image rotator; and the montemp program for cryogenic temperatures. The final gui's are for the telescope operator: the guider programs control either the pxl or slit viewing camera

guider and are integrated into the normal Keck guider program; the eavesdrop program gives a simpified version of the instrument status including exposure and guiding status.

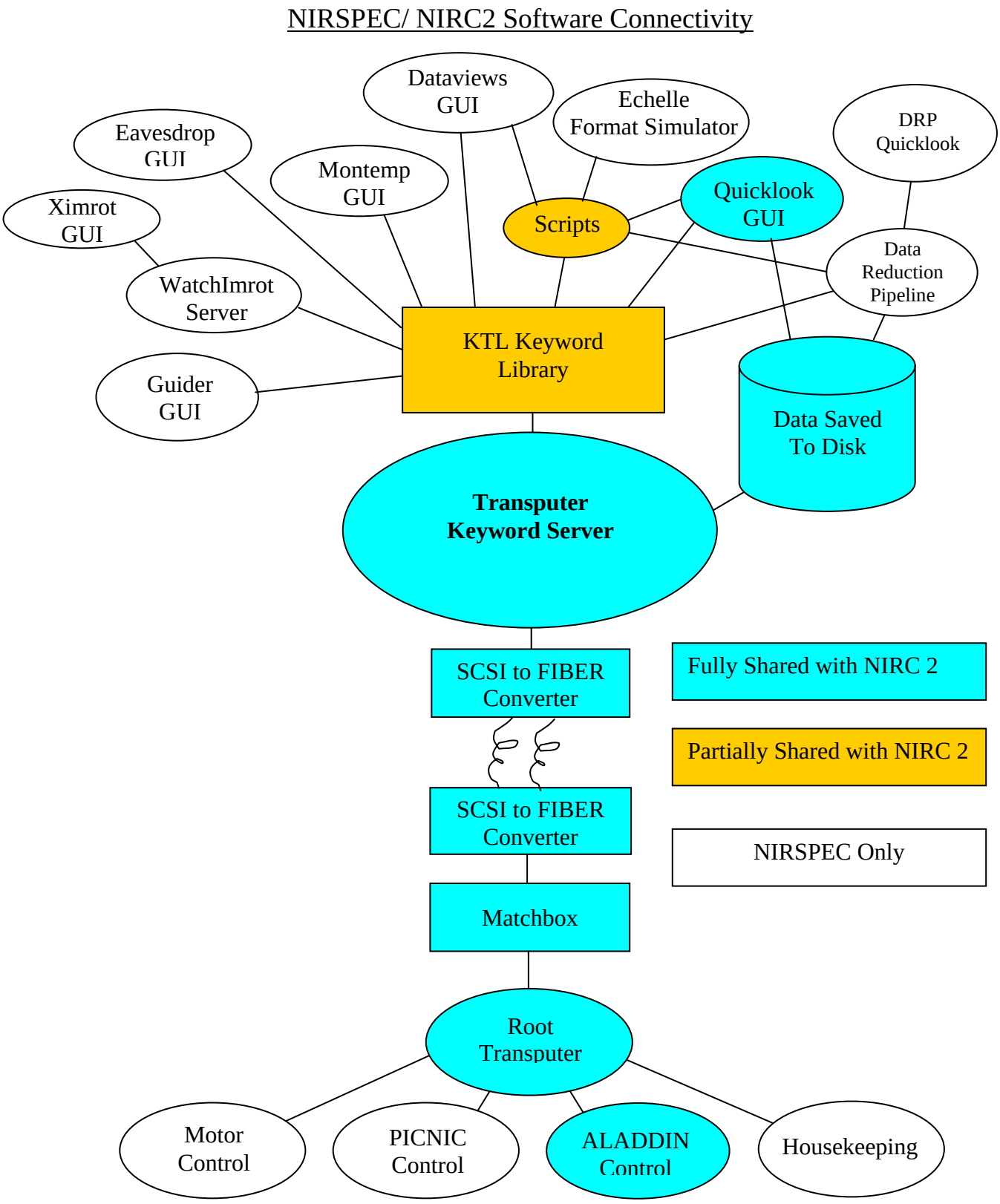

Figure 1. Software connectivity.

2.0 Possible GUI layouts on the screens

With so many gui's its important to arrange them in such a way that overlapping windows don't hinder the user's efficiency. We assume that the user has a two headed display and that the OA has a dedicated instrument display. We further assume that the data reduction pipeline is running on a separate computer so its processing doesn't slow the data taking computer. This means the gui's are spread over four monitors (with three users including the OA). The gui's mentioned above could then be arranged in the following manner:

2.1 Primary control screen

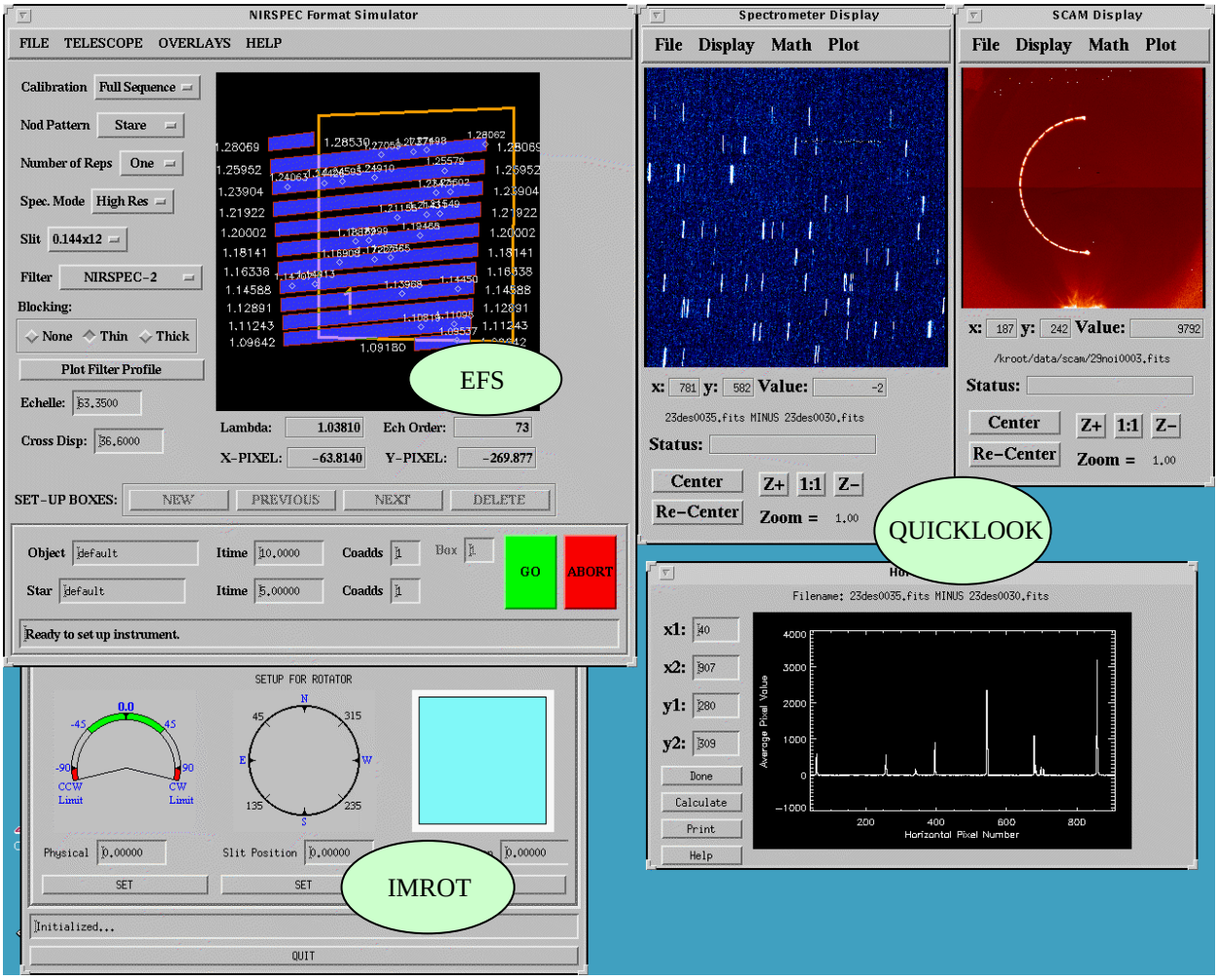

Figure 2. Possible screen layout with the EFS, quicklook and imrot.

2.2 Secondary control/ status screen

The user could use his second monitor to display the instrument status. If a third head is available, telescope status (facsum) could also be displayed without conflict. If only two screens are used, a facsum could be placed over the montemp program.

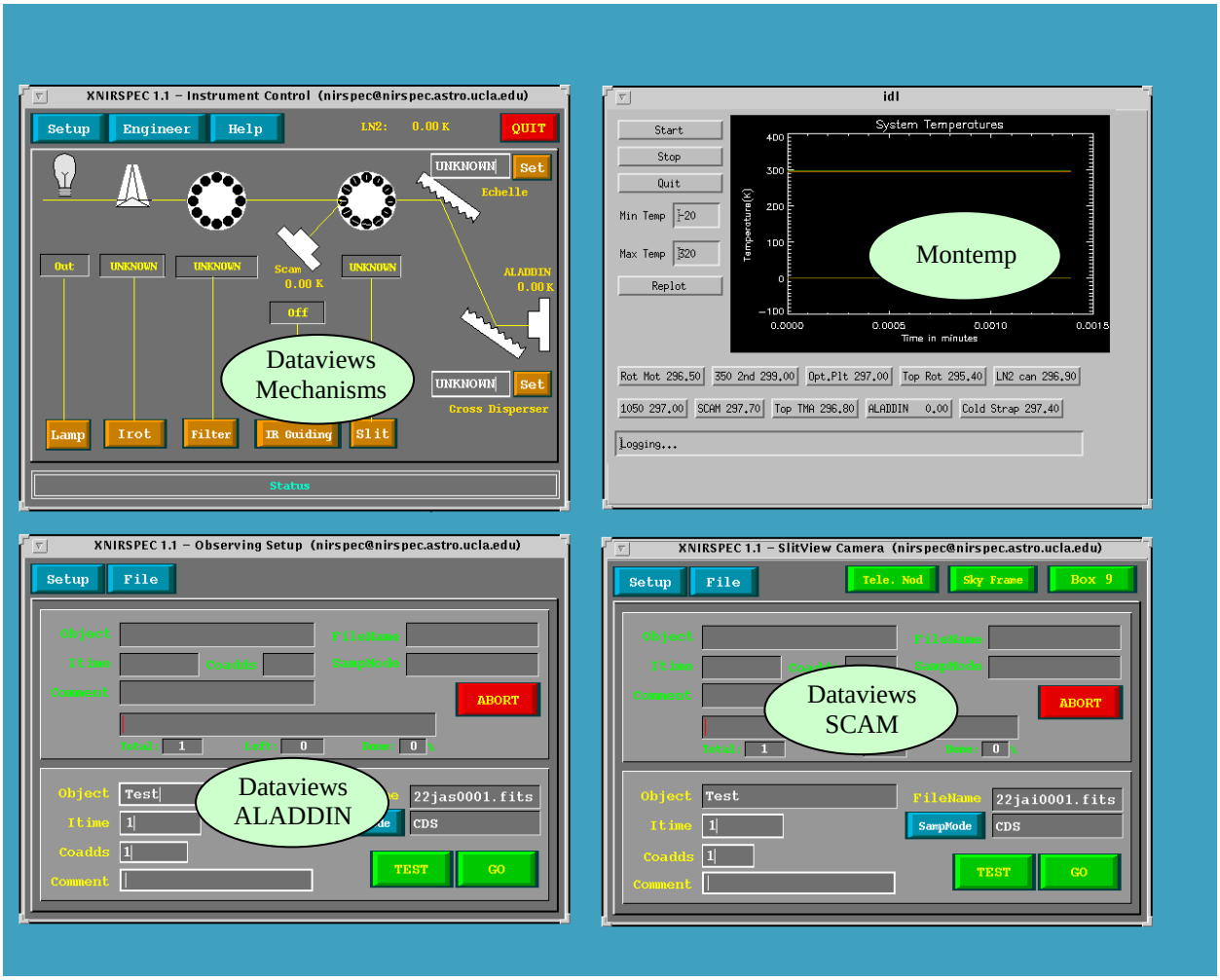

Figure 3. Possible status screen with dataviews, and montemp.

2.3 DRP screen

The DRP is cpu intensive so it should only be run on a completely separate computer with cross mounted disks. Using KIDL (KTL language adapted for IDL) the DRP monitors script progress and the completion of exposures. At the end of a script, it reduces the data frames and then spawns a special version of the quicklook that gives wavelength feedback both in the main window and in the vertical cutting mode.

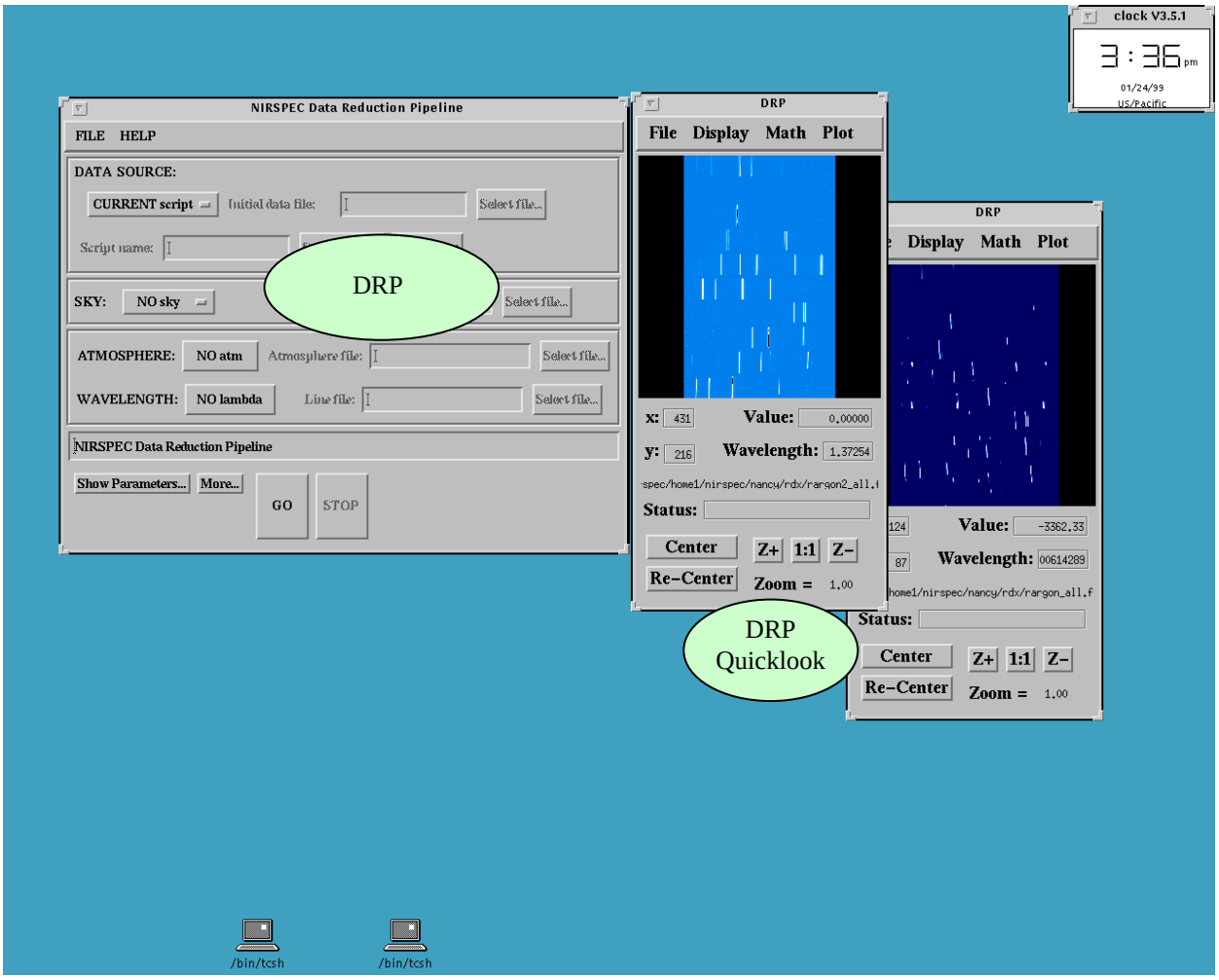

Figure 4. Possible screen with the DRP and drp quicklook.

2.4 OA instrument screen

Traditionally, the OA has controlled the guiding cameras used by the various instruments. This will continue with NIRSPEC and it will include control of the slit viewing camera when needed (embedded objects). In addition to the guider program, NIRSPEC has a special eavesdrop program that fairly slowly updates the exposure and guiding status of the instrument. These should be the only windows required by the OA for NIRSPEC functions.

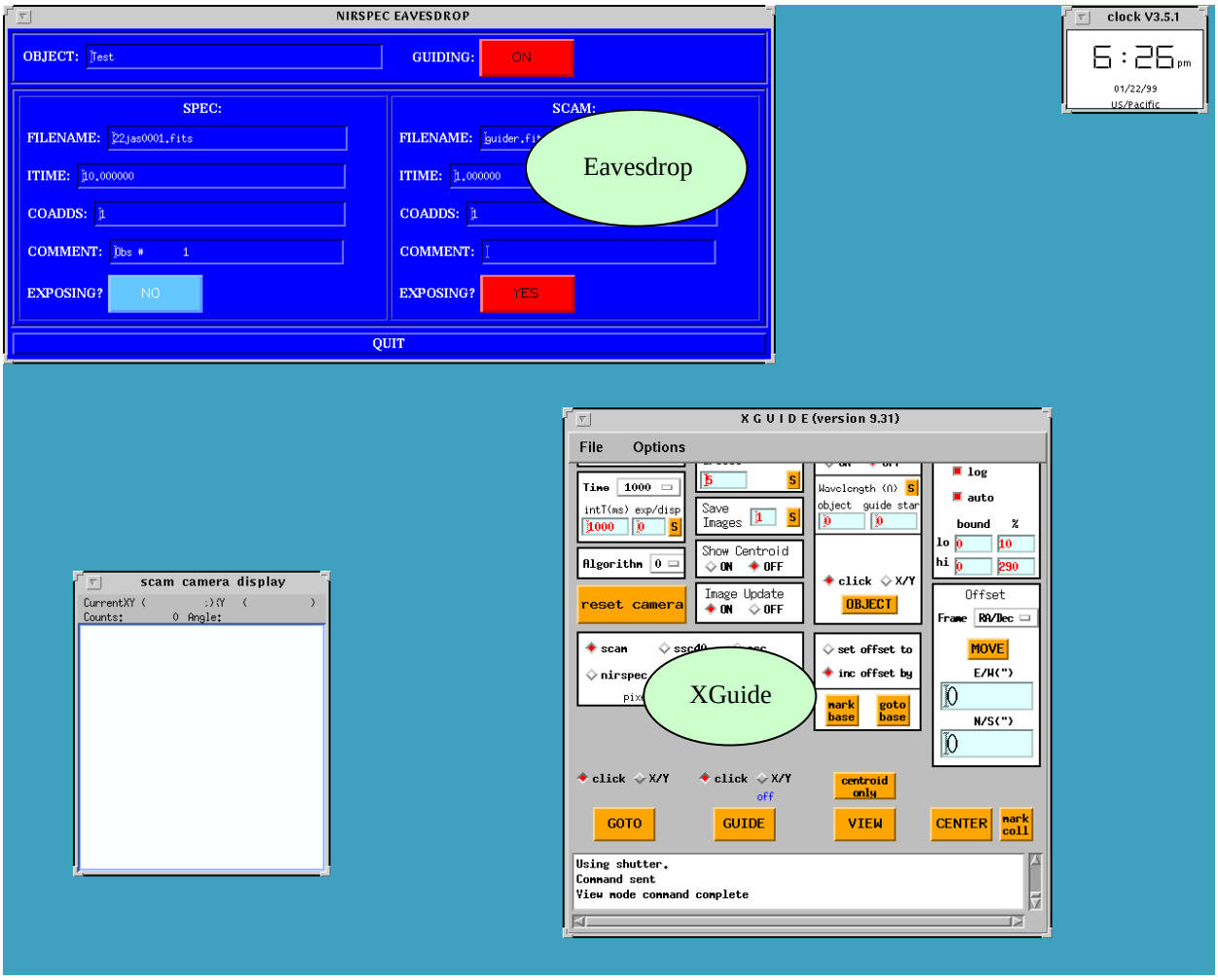

Figure 5. Possible screen for the Telescope Operator showing the NIRSPEC Eavesdrop, and xguide windows.

3.0 Individual Software Package Descriptions

nirspec_server_bin: The core c program that performs the transputer communications and manages keyword values and general instrument activity. All hardware commands must go through the server and by caching all keywords, it maintains an accurate state table of the instrument. The server uses RPC procedures to communicate with other high level routines.

Dataviews Gui (a.k.a. xnirspec or dataviews): The primary client of the server. It is written in c with a dataviews front end. The real Dataviews is actually a program for developing general graphical interfaces and it is used by CARA for many of its GUI's. Since xnirspec is the only major gui created with Dataviews for NIRSPEC, the instrument team began simply calling it dataviews. There are three primary windows for instrument control and status display. One window controls the mechanisms and provides a graphical display of the instrument status. Among the menu options of this window are motor initialization, engineering level control, and raw transputer communications. A second window controls the ALADDIN array. It is split to show both the current (or previous) exposure and the setup for the next exposure. One of the menu options allows the user to setup the data path, filenames and numbering and other observing information. The third window is virtually identical to the ALADDIN window but controls the SCAM. Xnirspec passes all commands on to the server via KTL. A full TCL command line was originally developed for the dataviews, but it has been commented out to avoid confusion with the UNIX shell.

quicklook: An IDL package that can monitor data taking and will automatically display images. It provides a basic image analysis capability. It uses the KIDL packages for server communications. It can attach to either the real or simulated server or run stand alone, or be driven from other IDL programs such as the DRP.

nirspec_efs: The EFS is an IDL package that calculates the echellogram patterns and allows for easy graphical configuration of the instrument. To control the instrument it generates scripts that are autonomous csh programs. These can be saved and read back in later or can be run right away. Scripts can be as simple as moving only a few mechanisms to a full observing script with nodding along the slit and slewing to calibrator stars.

data reduction pipeline: The Data Reduction Pipeline monitors script activity from the EFS to the server. As data frames are written, it reads them in and uses a sophisticated matrix transfer analysis of the instrument to extract spectral information. It is written in IDL and allows for a great deal of user configuration. It uses KIDL to connect to the server.

ximrot: XIMROT is a dataviews gui that displays the image rotator status, and allows the user to specify rotator positions. It sets the desired angles in the DCS and monitors the telescope position. It also has an engineering popup for NIRSPEC specific functions like initializing the rotator.

watch_imrot: The image rotator server that monitors DCS rotator demands and implements them with NIRSPEC. It uses a simple servo algorithm to read back the rotator position and DCS requested positions and then calculates the rotator velocities. Operates at 1 Hz. Maintains KTL links with the DCS and NIRSPEC.

montemp: Montemp is a simple IDL gui that can monitor all of the system temperatures. It produces graphs of the history of each sensor and writes entries in a log file every 5 minutes. It uses KIDL to monitor the server.

cam_server: The guider software also has a client server based system using KTL. The camera server is generic for all Keck guiders and speaks both to other camera servers and to the telescope control software DCS in order to guide. For normal use, the server is automatically started when you run the guider software.

xguide: The guider gui client that is the actual interface of the user or OA to the guiders. Xguide can control any Keck guider including the scam and pxl camera of nirspec. xguide is started automatically with the aut_start command; as is the cam_server.

eavesdrop: The eavesdrop program is a simple IDL gui that provides a minimal NIRSPEC status display for the OA. Once a second it updates one of its twelve keywords that it monitors. It shows the object names, exposure settings, exposure status and whether or not the scam can be used as a guider.

tsptest: TSPTEST is a separate C program that is capable of transputer level communications with the instrument. It can only be run independently of the server and the server can not be running since they each expect a dedicated communications link to the transputer. It is only used for engineering purposes.

occam: Occam is the computer language that is used for the transputer code. The transputers are responsible for all instrument activities including clocking and reading the arrays, controlling the motors and monitoring temperatures. It is broken into several individual packages for different transputers.