The NEWFIRM Observing Software: From Design To Implementation

P. N. Daly[a], N. C. Buchholz[a], M. J. Fitzpatrick[a], D. Mills[a], F. G. Valdes[a], N. Zarate[a], R. Swaters[b], R. G. Probst[a] and M. Dickinson[a]

[a]National Optical Astronomy Observatory, 950 North Cherry Avenue, Tucson AZ 85719, USA; [b]Department of Astronomy, University of Maryland, College Park MD 20742, USA

ABSTRACT

NEWFIRM is the wide-field infra-red mosaic camera just delivered and commissioned on the Mayall 4-m telescope on Kitt Peak. As with other major instrumentation projects, the software was part of a design, development, implementation and delivery strategy. In this paper, we describe the final implementation of the NEWFIRM software from acquisition within a `MONSOON` controller environment, directed by the observation control system, to the quick-look functionality at the telescope and final delivery of standardized data products via the pipeline. NEWFIRM is, therefore, the culmination of several years of design and development effort on several fronts.

Keywords: Software, Control Systems, Data Products, Pipeline Reduction, Guiding

1. INTRODUCTION

The **N**OAO **E**xtremely **W**ide **F**ield **I**nfra-**R**ed **M**osaic (NEWFIRM) camera program has now delivered a $27.'6 \times 27.'6$ field-of-view (`FOV`), $0.''4$ per pixel, 1–2.4μm focal plane instrument to the Mayall 4-m telescope on Kitt Peak. First light was achieved during February 2007 and several further commissioning runs have exercised all parts of the system. The new instrument is competitive with `WFCAM` on `UKIRT` and `WIRCAM` on `CFHT`[1] but provides the additional capability of utilizing non-standard cryogenic filters for special science programs.

In terms of the software, NEWFIRM was envisaged as one of a new breed of instrumentation[2,3] that follows an end-to-end paradigm. That is, data acquired at the telescope is driven by scientific use cases[4] and data product quality standards rely upon consistent handling of the data with little human intervention. Such standardized data products are becoming more common in all observatories and NEWFIRM would conform to this emerging practice via the use of observing recipes.[5] These are pre-defined data acquisition and reduction protocols for the most common observing scenarios. Thus, at the top-level, the system would be driven by scripts that could be prepared in advance. Once acquired, the data would undergo a quick-reduce process[6] to ensure data quality at the telescope before being fed into a slower data reduction and analysis pipeline. Such raw and derived data products would emerge—and be delivered to the principal investigator—via the NOAO science archive.

The principal elements of the end-to-end NEWFIRM observing software are shown in Figure 1.

2. OBSERVATION CONTROL SYSTEM

The initial software design was reported in previous papers.[7–9] Specifically, it was decided—at least for the high-level observation scheduling software—to adopt an existing package after a review of the state-of-the-art. We chose `DRAMA`[10] since this was compatible with `Tcl/tk`, which is heavily used in the `GWC`[11] infrastructure software at Kitt Peak. We adopted this approach rather than the more fashionable `Java` implementations seen today. Thus the current, high-level software collection known as the *NEWFIRM Observation Control System*[12] (`NOCS`) consists of a set of `DRAMA Tcl/tk` GUIs that provide access at 2 levels: 1) via the `DRAMA` command-line task `ditscmd`; and 2) directly from the GUI. In practice, we discourage level 2 access by observers since their scripts contain sufficient level 1 commands to complete a scientific observation. The `NOCS` command set is shown in Table 1 and a real-world example is shown in Table 2.

Further author information: (Send correspondence to P.N.D.)
P.N.D.: E-mail: pnd@noao.edu, Telephone: 1 520 318 8438

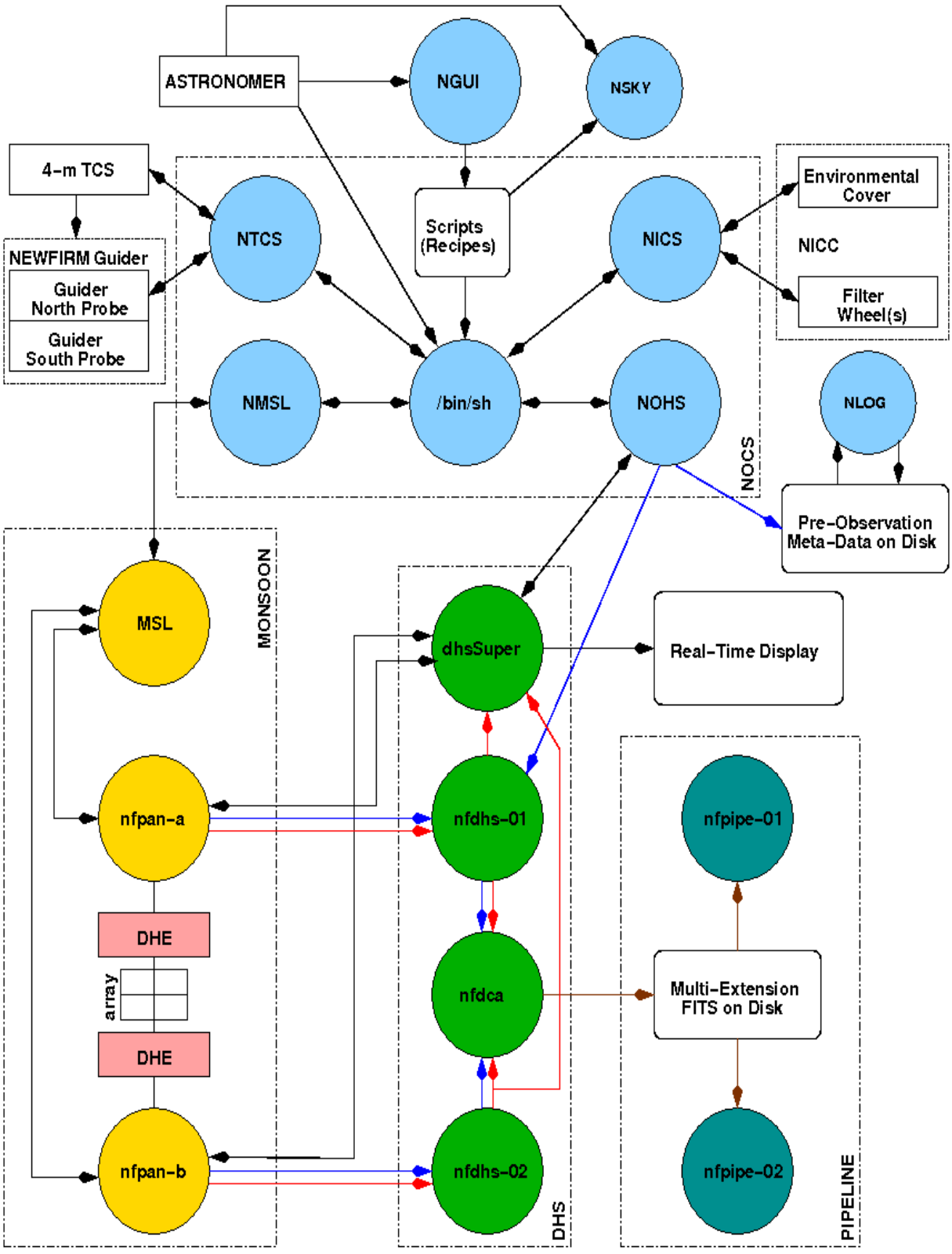

Figure 1. NEWFIRM integrated software architecture.

Table 1. NOCS command set sufficient for all observing protocols. All tasks support init, help, status and version commands. The commands ntcs_cam1, ntcs_cam2 and nmsl_gpxGetState are implemented but not fully commissioned.

Command:	Parameter(s):	Description:
nics_ecover	position	tell NICC to open or close the environmental cover
nics_filter	filter	tell NICC to set to given filter
ntcs_focus	value (μm)	tell TCS to set telescope to given focus
ntcs_gmode	mode	tell TCS to turn guiding corrections on or off
ntcs_instname	NEWFIRM	tell TCS to display the instrument name
ntcs_moveto	RA Dec Epoch	tell TCS to slew to given co-ordinates
ntcs_offset	RA ($''$) Dec ($''$)	tell TCS to offset in RA, Dec by given amount
ntcs_cam1	RA Dec Epoch	tell guide camera 1 to slew to given co-ordinates
ntcs_cam2	RA Dec Epoch	tell guide camera 2 to slew to given co-ordinates
nohs_newobs	meta-data ...	tell DHS the next observation is about to begin
nohs_endobs	(none)	tell DHS the current observation is complete
nmsl_gpxReset	(none)	tell MONSOON to reset itself
nmsl_gpxExit	URGENT	tell MONSOON to exit
nmsl_gpxAsyncResp	(none)	tell MONSOON an asynchronous event is acknowledged
nmsl_gpxSetMode	mode	tell MONSOON to set to given mode
nmsl_gpxSetMemCfg	IGNORE	tell MONSOON to set to known memory configuration
nmsl_gpxSetAVP	parameter=value	tell MONSOON to set a given parameter to given value
nmsl_gpxGetAVP	parameter	tell MONSOON to report a given parameter from each PAN
nmsl_gpxGetState	(none)	tell MONSOON to report state
nmsl_gpxStartExp	(none)	tell MONSOON to start an exposure

2.1 NICC and NICS

The *NEWFIRM instrument components controller* (NICC) is an autonomous sub-system that handles all functionality associated with the environmental cover, two filter wheels and various temperature sensors inside and outside the dewar. This software runs on a dedicated PC/104 stack, containing:

- VersaLogic Jaguar CPU module, running Fedora Core;
- WinSystems model PCM-COM8 8-channel serial module;
- ACCES I/O model 104-AI12-8 analog input board;
- two ACCES I/O model 104-AIM-32 analog multiplexers.

An engineering GUI is provided for very low-level access to the system but major elements—specifically for command and control—are provided by the mechanisms of the GWC software. These include an interface control document,[13] programmed as an API, that defines the commands and parameters that can be understood by higher-level clients.

The *NEWFIRM Instrument Control System*[14] (NICS) is the DRAMA Tcl/tk GUI that handles the interface between the NICC and the NOCS and allows scripting control of, primarily, the filter wheel. Since the filters can be changed out, both interfaces are driven by a well-maintained configuration file:

Table 2. A DARK script. This is a real, executable script—minus embedded comments—where line 20 has been shortened for brevity. Further details are provided in the text in § 2.7.

Line #:	**Command:**
00	#!/bin/sh
01	. $NEWFIRM_SBIN/nocs-functions.sh
02	trap 'trapSignal /home/observer/exec/DARK.sh $LINENO' SIGINT SIGTERM
03	ditscmd nmsl nmsl_init
04	ditscmd nmsl nmsl_gpxSetAVP Argument1="intTime=10.0"
05	checkReturnValue $? nmsl nmsl_gpxSetAVP "intTime=10.0"
06	ditscmd nmsl nmsl_init
07	ditscmd nmsl nmsl_gpxSetAVP Argument1="coadds=1"
08	checkReturnValue $? nmsl nmsl_gpxSetAVP "coadds=1"
09	ditscmd nmsl nmsl_init
10	ditscmd nmsl nmsl_gpxSetAVP Argument1="fSamples=1"
11	checkReturnValue $? nmsl nmsl_gpxSetAVP "fSamples=1"
12	ditscmd nmsl nmsl_init
13	ditscmd nmsl nmsl_gpxSetAVP Argument1="digAvgs=4"
14	checkReturnValue $? nmsl nmsl_gpxSetAVP "digAvgs=4"
15	ditscmd nics nics_init
16	ditscmd nics nics_filter Argument1="Dark"
17	checkReturnValue $? nics nics_filter "Dark"
18	EXPID=`$NEWFIRM_BIN/msd`
19	ditscmd nohs nohs_init
20	ditscmd nohs nohs_newobs Argument1="headers..." (see text)
21	checkReturnValue $? nohs nohs_newobs "headers..."
22	ditscmd nmsl nmsl_init
23	ditscmd nmsl nmsl_gpxSetAVP Argument1="expID=$EXPID"
24	checkReturnValue $? nmsl nmsl_gpxSetAVP "expID=$EXPID"
25	ditscmd nmsl nmsl_init
26	ditscmd nmsl nmsl_gpxStartExp
27	checkReturnValue $? nmsl nmsl_gpxStartExp "(none)"
28	ditscmd nohs nohs_init
29	ditscmd nohs nohs_endobs
30	checkReturnValue $? nohs nohs_endobs "(none)"

```
#+
# NEWFIRM Filter Wheel List 14-Jan-2008
#
#NICC  \NGUI    \Combination           \FW1 Pos\FW2 Pos\Serial Number(s)
#-
J      \J       \JX        + Open      \1      \8      \Unknown
H      \H       \H1+2Block + H         \2      \1      \Unknown
```

```
K       \K       \KXs       + Open      \3      \8      \Unknown
1056    \F1056W  \1060Block + 1056      \4      \4      \Unknown
1063    \F1063W  \1060Block + 1063      \4      \5      \Unknown
J1Yale  \J1Yale  \J1        + J1Block   \5      \6      \Unknown
J2Yale  \J2Yale  \J2        + J2+3Block \6      \7      \Unknown
J3Yale  \J3Yale  \J3        + J2+3Block \7      \7      \Unknown
H1Yale  \H1Yale  \H1+2Block + H1        \2      \2      \Unknown
H2Yale  \H2Yale  \H1+2Block + H2        \2      \3      \Unknown
Dark    \Dark    \KXs       + J1Block   \3      \6      \Unknown
Open    \Open    \Open      + Open      \8      \8      \Unknown
```

2.2 NTCS

The *NEWFIRM Telescope Control System*[15] (NTCS) is the DRAMA Tcl/tk GUI that handles the interface between the 4-m telescope control system (TCS) and the NOCS. Although is has been commissioned to support large telescope slews, these are currently disabled due to safety concerns. Modest telescope offsets, however, are fully integrated and provide the (vital) dither and map functionality required by science programs. The 4-m TCS also handles guider on/off commands, secondary focus demand positions and the name of the currently used instrument so these are all available from the NOCS (as shown in Table 1).

2.3 NOHS

The *NEWFIRM Observation Header System*[16] (NOHS) is the DRAMA Tcl/tk GUI that handles the interface between various telemetry systems and the NOCS. It is a passive task that subscribes to GWC streams, accepts script-level meta-data and sends pre- and post-observation meta-data to the DHS. Indeed, the nohs_endobs command is the action that kicks the entire data handling process into gear.

2.4 NMSL

The *NEWFIRM MONSOON Supervisor Layer* (NMSL) is the DRAMA Tcl/tk GUI that handles the interface between the MONSOON supervisor layer[17,18] (MSL) and the NOCS. This task also subscribes to the array temperature telemetry stream and does *not* allow command and control of array voltages if the temperature is outside a specified range. Further array-safety protocols are also being developed and implemented.

2.5 NGUI

A deliberate design decision, early on in the project, was to provide only one interface to the system. Since scripting was identified early on, the default interface is the script generating engine which was christened with the (relatively non-descriptive) name *NEWFIRM Graphical User Interface*[5,19] (NGUI). NGUI was designed to be completely separate from those parts of the NOCS that must exist within the telescope environment. Indeed, observers can download the latest version of NGUI and run it on their laptops—the only requirement is a working version of Tcl/tk. No compilation is required. Thus, observers see the same interface and control the system the same way as they would to create their science scripts.

Although we tried re-using existing 'observing tools', we adopted the in-house solution for several reasons. First, most observing tools used Java or other languages that were not in common use at KPNO. Second, they tend to be very large pieces of software and hard to understand. By contrast, NGUI supports all 18 observing recipes required by the NEWFIRM science drivers and takes barely 200 lines of Tcl/tk to implement each recipe. Adding new recipes is very easy even for the most complicated protocols (*e.g.,* as shown in Figure 2) since 'building-block widgets' are used to create the appropriate dialog box.

2.6 Other NOCS Items

2.6.1 NSKY

One problem with NGUI is that it takes positions and offsets from 'entry' and 'spinbox' widgets (see Figure 2) and not from a sky map. Rather than add this functionality, we chose to implement a small utility to facilitate the visualization of dither or map patterns. This is NSKY and its dependencies are Tcl/tk, ds9 and xpans. It has proved invaluable as a debugging tool for the script generating engine, NGUI.

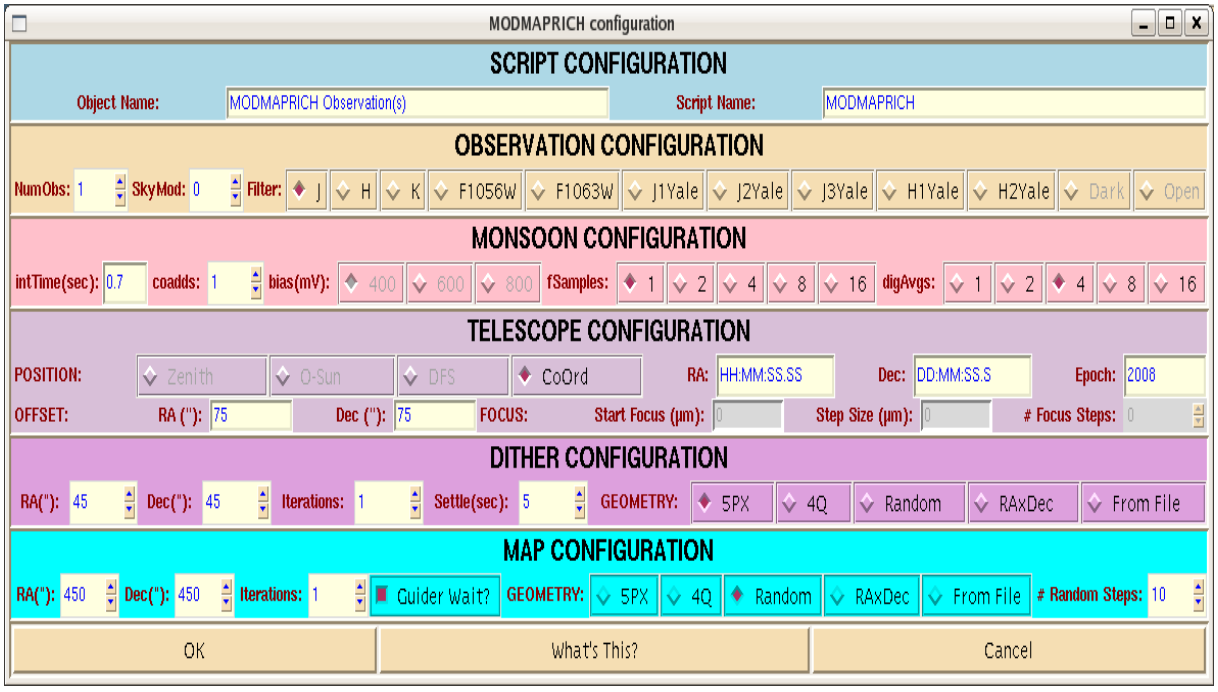

Figure 2. `NGUI` dialog box for the MODMAPRICH recipe, one of the most complex.

2.6.2 NLOG

This is a little-used utility that reads pre-observation meta-data and automatically generates observing logs.

2.7 Step-by-Step Through a Script

Figure 3 shows the message sequence chart for a generic NEWFIRM observation. The simplest script is a DARK observation as shown in Table 2. Typically, each script is divided into blocks that address the described subsystems. Using the script in Table 2 as an example, lines 00–02 start the standard Unix system management (fast) shell and sources $NEWFIRM_SBIN/nocs-functions.sh, which provides 2 essential functions: a) `trapSignal` which is invoked when the observer presses Ctrl/C; and b) `checkReturnValue` which checks the return status of the previous command and aborts the script on error. After that the elements of the script may change according to the recipe invoked. Following the example, lines 03–14 tell the `MONSOON` system the observing parameters (integration time, number of coadds, Fowler samples and digital averages). Lines 15–17 tell the `NICC` which filter to place in the beam.

Line 18 gets a unique (and dynamic) exposure identifier (`EXPID`). The `EXPID` is the single piece of meta-data that ties everything together. `MONSOON` passes pixel and meta-data to the `DHS` using the `EXPID` as the identifying tag. The `NOHS` does the same for pre- and post-observation meta-data. Since `EXPID` is dynamic, the script can be re-used time and time again.

Lines 19–21 send pre-observation meta-data to `DHS`. The argument on line 20 provides the `NOCS` meta-data such as (in this example) `NOCNUM=1 NOCNPOS=1 NOCFSMPL=1 NOCOBJ=DARK.ws.1x10s*` `NOCNO=1 NOCTOT=1 NOCSKY=1 NOCDHS=DARK NOCSCR=DARK NOCDGAVG=4 NOCTIM=10.0 NOCCOADD=1 NOCFIL=Dark NOCTYP=DARK NOCID=$EXPID NOCBIAS=400 NOCSYS=kpno_4m`. Lines 22–27 tell `MONSOON` the `EXPID` and start the exposure. After the exposure completes, lines 28–30 send post-observation meta-data to `DHS`.

*Note that we have here an encoded 'verboten' character. A single space is not allowed so it is encoded as '.ws.'. The `KTM` (see § 5.1) decodes this to restore the original string in the science meta-data set. Other verboten characters are handled in a similar way.

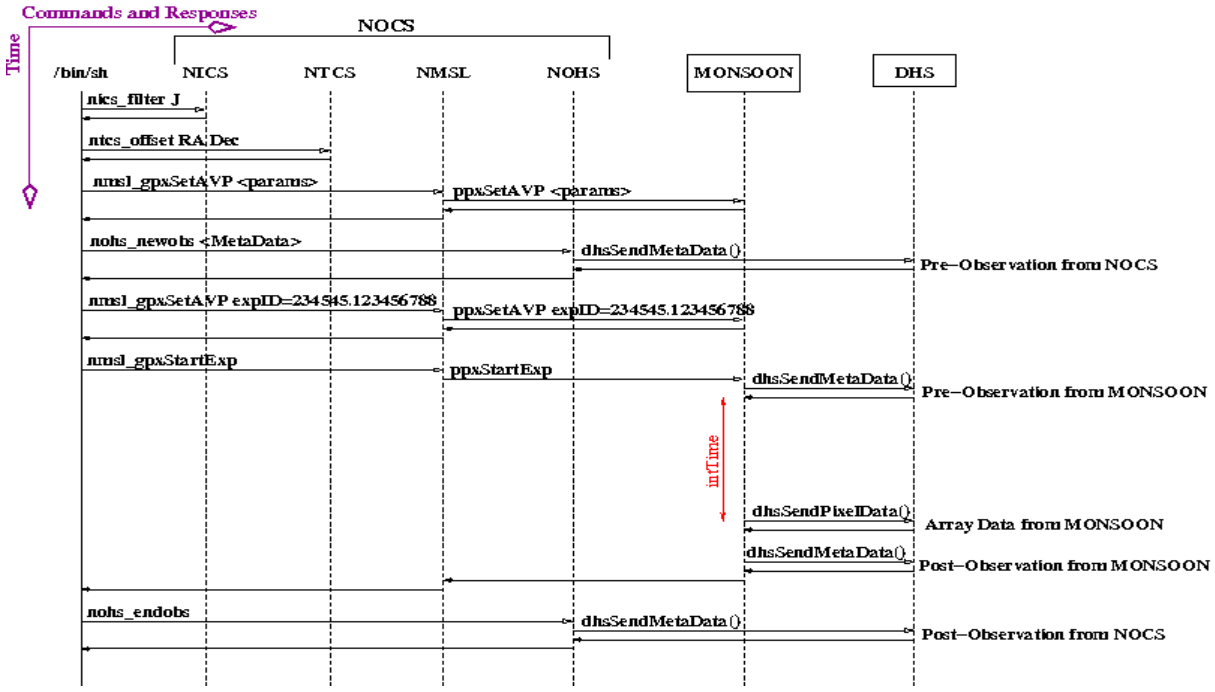

Figure 3. NEWFIRM message sequence diagram for a typical observation.

All NOCS sub-systems have counters to test for command expiration. If a command times out, the status returned is sufficient to abort the executing script. Note that a side-effect of using /bin/bash scripts is that the command line returns a simple SUCCESS or FAILURE string to the observing terminal (via the checkReturnValue function), but the command responses and interim status messages show up in the GUIs.

3. MONSOON GENERIC PIXEL SERVER

For NEWFIRM, the detector focal plane consists of four 2048×2048 arrays arranged as a 2×2 mosaic. MONSOON is the NOAO scalable architecture for focal plane detectors and is described elsewhere,[19] but some innovations to support NEWFIRM were:

- A supervisor layer (MSL) to support a single entry point for multiple pixel acquisition nodes (PANs). The MSL[18] is configurable for any number of slave PANs with a single master;
- Multiple digital averages and/or Fowler samples (non-destructive reads) of the arrays;
- co-addition internal to the MONSOON acquisition environment; and
- full support of ppxAbortExp, ppxPauseExp and ppxResumeExp[†].

4. DATA HANDLING SYSTEM

The *NEWFIRM Data Handling System* (NDHS) borrows some architectural approaches from an earlier project for the NOAO CCD mosaic imagers, the *Mosaic Data Handling System* which is described elsewhere.[20] The NDHS is responsible for collecting all of the data and meta-data produced by the NOCS and the MONSOON array controller, and for ultimately creating the final science image. It also provides the *Real-Time Display* (RTD) and image rectification required to properly orient the image sub-rasters. Final assembly of the image, into a

[†]These commands have been implemented in the MONSOON base code set and tested on other systems and should be commissioned within the NEWFIRM environment by the time of the symposium.

multi-extension FITS (MEF-FITS) file, filters the meta-data needed for the final science header and adds computed values such as the image world coordinate system (WCS). Meta-data not stored in the science image are saved to a separate engineering file for later analysis. Finally, the DHS triggers the various procedures needed to stage the image for archiving and the quick-look pipeline system.

The design of the system is driven primarily by the need to manage the data in a number of time domains simultaneously:

1. The instrument is capable of a "burst mode" in which new images can be generated at a rate of several per second. The typical observing cadence is normally much slower, however;
2. The DHS processing time scales with the amount of compute power available and is relatively stable at ∼5-6 seconds per image with the current hardware configuration;
3. Pipeline processing requires a variable number of images depending on the observing recipe being used and, in some cases, the pipeline cannot begin until an entire observing sequence is complete.

To meet these needs, we settled on an architecture consisting of a distributed set of applications connected via an event-based messaging system. A DHS supervisor application provides the main user GUI and serves to sequence the messages needed to move data through the system at the appropriate time.

A key element of the architecture is that data collection is independent of DHS processing. We utilize a large memory buffer to store raw frame data until it is ready to be processed, adding data whenever a readout occurs and continually processing any data remaining in the buffer. This allows us to collect data at a rate faster than it could naturally be processed, and to take advantage of the times between and during exposures (*e.g.,* when moving the telescope or during long integrations) to process any backlog. Some elements of the system, such as the RTD, respond with each new image in order to provide user feedback for the data as it is read out.

4.1 Architectural Components

The NDHS is comprised of several independent applications and components running on a number of machines, although these are commonly referred to singularly as the DHS. Major components and applications of the DHS are described below.

4.1.1 DHS API

To minimize mis-communication between the development teams, a programming interface that handles the details of data transfer and messaging to the DHS was developed. This API establishes communications with the DHS supervisor layer and the serialization of the data and meta-data being sent to the DHS, while hiding any complexity from the other programming groups.

4.1.2 Shared Memory Cache Interface

A *Shared Memory Cache* (SMC) interface provides a common API to all applications needing access to the image readout. These data are stored in memory to optimize processing times, and various pages of data associated with a particular image are all tagged with a common EXPID value to allow easy searching. The SMC acts as the buffer between applications collecting new data and those processing the backlog that can occur when readouts happen faster than they can be processed by the system. This buffer can be increased to accommodate a larger backlog simply by adding more memory to the system.

4.1.3 Message Interface

The abstract messaging interface allows applications to send messages and data via the message bus to other applications without needing to establish direct communications. This interface is currently layered on the *Parallel Virtual Machine* (PVM) message system but could easily be re-implemented to use a different underlying transport mechanism without affecting the DHS applications.

4.1.4 Real-Time Display

The RTD provides a rendering of the raw image at the moment it is read out from the system. The display server is a modified XImtool optimized to accept display connections from the multiple hosts in the DHS system and to write small packets of data to arbitrary parts of the display while building up the complete image. An intensity-scaled and fully rectified image of the focal plane is normally available within a second of the end of the readout.

4.1.5 Supervisor

The DHS supervisor is responsible for providing the main user interface and control features of the DHS as well as managing the various event messages in the system that trigger the next stage of processing. All applications in the DHS communicate with the supervisor either through a direct socket connection (*e.g.,* the PAN software using the DHS API) or via the message bus. A configuration file is used to tell the supervisor how many applications are available and how they should be deployed across the various machines; the next stage of processing is begun only after all applications have reported completion of the current stage, to avoid synchronization issues in the system.

4.1.6 Collector

The collector is responsible only for reading data and meta-data from the PANs and NOCS and storing the result in the SMC. These collectors—one each for the PANs and a separate one for the NOCS—communicate using the DHS API running in those applications. The application itself is completely oblivious to the type of data being read; its only job is to read and store the data as fast as it can.

4.1.7 SMCMgr

The *Shared Memory Cache Manager* (SMCMgr) is responsible for rectifying the image data in memory, *i.e.,* rotating or flipping each sub-raster (in memory) of the focal plane so it can be properly displayed and stored in a FITS file. This application also acts as the RTD client, displaying those parts of the focal plane where it has data. Lastly, the SMCMgr is responsible for managing the SMC on the machine, *e.g.,* it will free any pages in the SMC once image processing is complete or will flush images from the system during error recovery. Readout data is not removed from the SMC until a successful transfer of all data is completed, allowing the image to be recovered in the event of a failure in the system.

4.1.8 PXF

The *PiXel Feed* (PXF) application is responsible for processing all SMC pages with a common EXPID, beginning with the data flow from the machines collecting the readout pages to the single application on the message bus that will assemble the full science image. Data and meta-data pages are "packetized" into smaller chunks for transfer.

4.1.9 DCA

The *Data Capture Agent* (DCA) is responsible for assembling the final FITS image, processing the final science header meta-data, and triggering the post-processing scripts postproc and sysproc. Data and meta-data packets arriving from the various PXF applications on the message bus are random but tagged with information that lets the DCA know to which part of the focal plane the packet belongs. By this stage, the image pixels have already been rectified and need no further processing. The meta-data are not in standard FITS format and need massaging or translation by the *Keyword Translation Module* (KTM, see § 5.1) to be compliant. Post-processing scripts are executed to transfer the final image to the scientist's workstation and trigger archive and pipeline processing.

4.2 DHS Configuration and Deployment

The DHS may be configured to run on any number of machines, although optimal performance is achieved by spreading the applications to machines that have dedicated processing requirements and where system loading will not interfere with other aspects of the system (*e.g.,* by having the DCA run on a dedicated machine so it doesn't hinder data collection). Software development was done primarily on a single system running all the applications of a full DHS and simulators for the instrument software with no appreciable loss of performance, and the system is indeed designed to be quickly reconfigurable in this manner in the event of catastrophic hardware failure.

Typically, the DHS supervisor will be deployed to the computer used to run other instrument control GUIs and which serves as the scientist's primary workstation. This is a fairly lightweight application and presents no real load on the system. Conversely, the DCA represents the majority of processing time in the system when active and so is deployed to its own machine. The collectors, SMCMgr, and PXF application must all reside on the same machine since they use the SMC to store and access the data during processing. In the current configuration, one machine running this bundle is assigned to each of the PANs controlling half of the focal plane, and an additional collector is assigned to only one of those machines to handle the meta-data coming from the NOCS with each exposure. The system is scalable to larger and more complex focal planes by simply adding new hardware dedicated to processing data from additional PANs.

All DHS applications are connected via the message bus and share a common Ethernet routing. The PAN-to-collector connection is made using a second dedicated Gigabit Ethernet card in the machine to assure the highest possible throughput when collecting data. The system startup script reads a configuration file that determines both the application deployment on the machines as well as the network connections used. This allows us to easily use different configurations when the system is being run in a development environment, an engineering laboratory or at the telescope.

5. PIPELINE PROCESSING SYSTEM

The NEWFIRM software architecture contains 2 post-acquisition data processing pipelines: the *NEWFIRM Quick Reduce Pipeline* (QRP) and the *NEWFIRM Science Pipeline* (NSP). The latter is currently under development with an anticipated beta-test in semester 2008B and will run at a pipeline data center.

5.1 Keyword Translation Module

The *NEWFIRM Keyword Translation Module* (KTM) handles the meta-data telemetry collected by the DHS data collection agent (DCA) from NOCS and MONSOON. It provides a programmable interface between the acquisition systems and the science and engineering data handling systems so those "up-stream" systems don't need to concern themselves with keyword name spaces, value formats, derived meta-data or auxiliary information not directly related to their function. An operation benefit of this architecture is that it was possible to fix meta-data problems quickly during commissioning independently of the acquisition components.

The KTM is responsible for interpreting, formatting, remediating, augmenting, and disposing of the meta-data. It ensures the meta-data is complete and compatible with archive, pipeline and user reduction systems such as IRAF. The primary output of KTM is the set of keywords and values for the headers of the final MEF-FITS files. Primarily, engineering telemetry is excluded from the FITS files and, instead, the KTM provides another output file with all the raw telemetry. NOAO is now beginning a project to flow this data to an engineering archive service that stores it in a database and provides access facilities for the engineers and instrument scientists.

The DCA places the telemetry in a number of keyword "databases" which are accessed using an API with Tcl bindings by the KTM Tcl script. Note that the DCA simply passes this telemetry blindly and does not modify or interpret any of the content. The input to the KTM is a set of "database" pointers and the outputs are files and database pointers for the primary and extension headers of the MEF-FITS file. The KTM also reads auxiliary data files that control the behavior of the KTM and provide additional meta-data, such as initial WCS descriptions. When the KTM completes, the DCA populates the MEF-FITS file so the KTM does not need to be involved in the mechanics of FITS headers. It does, however, have to generate the keywords and values that conform to the FITS standards.

Rather than code all the possible telemetry in the KTM, a configuration file is used. This file lists all the expected telemetry keywords and provides flags for whether the information is to be included in the science meta-data. It also provides a level of remapping for the telemetry keywords (which need not be FITS style or length) and the comment strings. For the most part, the KTM simply passes on the information marked as science meta-data to the final headers.

One might worry about new telemetry information being added by the acquisition systems. The KTM includes any telemetry values not defined in the configuration file in the FITS header. Later, the KTM and configuration file can be updated for this new telemetry. This decouples the acquisition software from coordinating telemetry meta-data with the KTM in a tight fashion.

For the QRP, the KTM produces a trigger file but does not, in and of itself, trigger the pipeline. Instead, the contents of the file provide sequence information used by a post processing command to define the trigger filenames. The KTM manages this information because the trigger filenames encode sequence and end-of-sequence information provided by the NOCS. A complexity also handled by the KTM is the situation where the sequence ends prematurely, either by user initiative or by system failure. Because the information about sequences and which element of a sequence is being read out is part of the meta-data, only the KTM is aware of sequences; the DCA is only concerned with the current exposure.

5.2 Post Processing

The DCA provides a facility to execute post-readout processing commands. These commands are passed the EXPID, filename and directory. The NEWFIRM system uses this facility to queue exposures to archive via the data transport system and to the quick-reduce pipeline. The latter makes use of a file produced by the KTM to specify filenames for the file triggers expected by the QRP. The pipeline triggering requires sending light-weight files to a remote machine that is part of the quick-reduce pipeline cluster. This is done using IRAF networking, though other remote file transfer methods could be used.

5.3 Focus Routine

The NEWFIRM project included development of an IRAF package for users to reduce and analyze NEWFIRM data. This package is available as part of the observing environment. The main task used in this environment is nffocus to analyze focus sequences.

There is a FOCUS recipe that offsets in order to take a sky exposure then returns to the starting point, where a sequence of exposures is taken while the focus is changed as specified by the observer. The brighter sources in each (selected) exposure of the sequence are cataloged. If a sky exposure is used, the source detection is done in difference mode, which is essentially detection with sky subtraction. The cataloged sources include a full width at half maximum (FWHM), and those with FWHM more than $2\frac{1}{2}\times$ the mode are filtered from the output.

The sources in the catalogs are matched spatially. Initially only those sources matched in all exposures are used, to avoid biases, and sigma clipping eliminates significant outliers from extended sources. The FWHM values of the matched stars as a function of focus value are displayed and the user has a number of options for deleting bad data and for examining the focus variations spatially. At each step the task computes a "best" focus value which the observer may adopt, or the observer may estimate from the graphs.

5.4 Quick Reduce Pipeline

The QRP provides data quality feedback for the observer during the night. It has also proven valuable for instrument scientists to catch problems early by accessing the pipeline results remotely. Data quality parameters measured by the pipeline include the sky brightness, seeing and an approximate photometric zero-point. The pipeline also delivers first-pass calibrated images to the observer.

Observations generally consist of scripted sequences of exposures, so the QRP was designed to process these sequences as a coherent data-set. These sequences are typically dithered exposures, allowing the construction of stacked images with bad pixels and mosaic gaps removed. A consequence of the sequence-based design is that only limited processing is performed until the last exposure of the sequence is completed. This limits how quickly results are provided and depends on the number of exposures in the sequences, the cadence of the observations,

and the hardware dedicated to the processing. The pipeline is built on a distributed and parallel processing framework[21] which allows scaling by adding additional computing resources. The current pipeline runs on two dual-core machines but there are plans to increase the amount of processing during the sequence and to increase the computational resources.

The QRP is made aware of a new exposure through a file trigger event initiated by the DCA. Two light-weight content files are written to the QRP trigger directory with the path to the FITS file and the directory specified by the user for receiving data products. The trigger event is a third file created by a simple touch. A separate zero-length trigger file is used to avoid the pipeline attempting to access the content of the files before the information is completely written. An additional file may be written to indicate that the exposure is the last in the sequence.

An interesting architectural feature is that, while the trigger files are sent from the observing computer, the path to the data is to the DHS computer. The pipeline only *pulls* a copy of the data, so this is safe. It would be unsafe to reference the copy provided to the observer since this file could be modified before the pipeline accesses the data.

The primary data product of the QRP are single stacked images constructed from the sequence. The pipeline detects and creates stacks based on automatically identified overlaps after astrometric calibration. The individual images are dark subtracted, linearity corrected and flat fielded. The dark and flat field calibrations use master calibrations derived from dark and flat field sequences. The QRP also processes these calibration sequences and stores them in a calibration library. Note that in keeping with the goal of quick best effort reductions, these calibrations are optional and skipped if no suitable calibration sequences have been received, processed and checked into a calibration library.

Sky subtraction is one of the most important processing steps for infra-red data in general and for the QRP in particular. Meta-data from the sequences define whether offset or in-field sky subtraction is performed. In-field sky subtraction generally consists of a running median with a time sorted window of at least 5, and no more than 9, exposures. The running median excludes the exposure being calibrated and uses the statistics of the pixel values to clip sources before computing the median.

Each exposure is astrometrically calibrated using 2MASS sources, automatically detected, in the exposure. The WCS includes distortions from the optics, unavoidably present because of the wide-field of view. Each array has its own WCS, which maps the relative geometry of the mosaic and the optical distortion in the telescope and camera. The astrometric calibration is required to allow approximate photometric data quality comparisons with the 2MASS sources and to re-project all data to common final stacked images. If the astrometric calibration fails, the basic instrumental calibration is still performed but the exposure is not included in the final stacked data product and the data quality does not include the 2MASS calibrated zero point.

Bad pixel masks are used to cosmetically remove detector defects and to exclude bad pixels from the final sequence stack or stacks.

Because of the quick-look purpose of the QRP, only the stacked images are provided to the observers, though this could be easily extended to the individual exposures if needed. The images are MEF-FITS files which may be displayed or otherwise used by the observer. Observers are cautioned that these files are *not* final science data products. Such products will be produced, within a few days, by the in-development NSP running at a pipeline data center.

A key product of the pipeline are data review web pages. There are two types of web pages. A tabular summary, one row per sequence, displays information about the sequences from the meta-data and derived data quality. It is presented much like an observing log. The entries in the summary table include links to a page for each sequence. These sequence pages provide postage stamp and large PNG graphics for each processed exposure and for the stacked images.

The observer is not expected to control or interact with the QRP, which is designed to run automatically with minimal technical support. There are currently no resources for night-time maintenance of the QRP and starting, stopping, and responding to problems is handled remotely by pipeline personnel. However, observers do like to know whether the pipeline is running. Therefore commands are provided to check the status of the pipeline

(plstatus), showing whether the pipeline machines and pipeline processes are up and running, and sequences (qrpstatus) showing at what stage of processing each sequence is at.

The QRP stores data quality metrics in a pipeline database. A user command (dqquery) is provided to generate reports from this database. The text output may be used to examine the latest data quality measurements or to feed them into plotting programs for graphical display.

6. THE GUIDER

The NEWFIRM dedicated guider is a KPNO-built sub-system that comprises hardware to support dual units, offset north and south of the detector array. Each guide unit has a SBIG STV cooled CCD camera mounted on a 4-axis compumotor 6k6-controlled stage, and supports motion in RA, Dec, Focus and astigmatism corrector lens rotation. Each sub-assembly is connected to other systems by Ethernet *iServers* to the STV serial port and computer serial port, Ethernet to the compumotor port and analogue video output from the STV camera. Output video from each unit is fed into the video switcher, from where it may be directed to a number of display stations, and optionally fed into the frame-grabber based generic Linux guiding system.

The NEWFIRM guider software package is responsible for positioning the 2 (north and south) camera stages at the appropriate position to center a designated guide-star. Automatic adjustments of focus and corrector lens angle are also supported. The package includes diagnostic GUI interfaces and scripting level control. Remote control is provided by an interface to the software message router. A TCS plug-in is provided to permit local telescope operator control of the guider, and assist in guide-star selection. A second component is a slightly modified version of the KPNO standard Linux software guider. This runs on a dedicated computer and samples the video output of the guider cameras to feed the TCS.

The software design takes a layered approach consisting of:

- Low-level C code built as loadable shared libraries;
- Tcl script code to wrap the low-level functionality;
- User interface—Tk GUIs which utilize the scripts;
- High-level command and response via the GWC message router.

6.1 Guide-Star Selection

The guide-star selection GUI is shown in Figure 4 can be used in two forms. First, it can be run as a stand-alone GUI and, second, it can be embedded into the local telescope operator's XTCS telescope control GUI. The ds9 image display program is utilized to present a view of the current pointing, and overlays show the detector array, guide probe roaming areas and current locations. Guide-star selection is a simple point-and-click process. Automated guide-star selection from a pre-defined observation is also supported. In this case the NOCS—or other suitable 'observation manager'—issues the guide-star target request via the GWC message router. Note that NSKY is a derivative software product of the guide-star selection tool.

6.2 Sequence of Events for Guiding

In normal operations the following set of commands and other messages will flow across the interfaces of the NEWFIRM guider sub-systems:

1. NOCS issues newfirm.camera1.target or newfirm.camera2.target request via GWC router;
2. guider server process receives target request from GWC router;
3. guider server process issues stage motion commands to 6k6 via *iServer* socket;
4. guider server process waits for motion completion;
5. guider server commands Linux guider to acquire star (via socket);

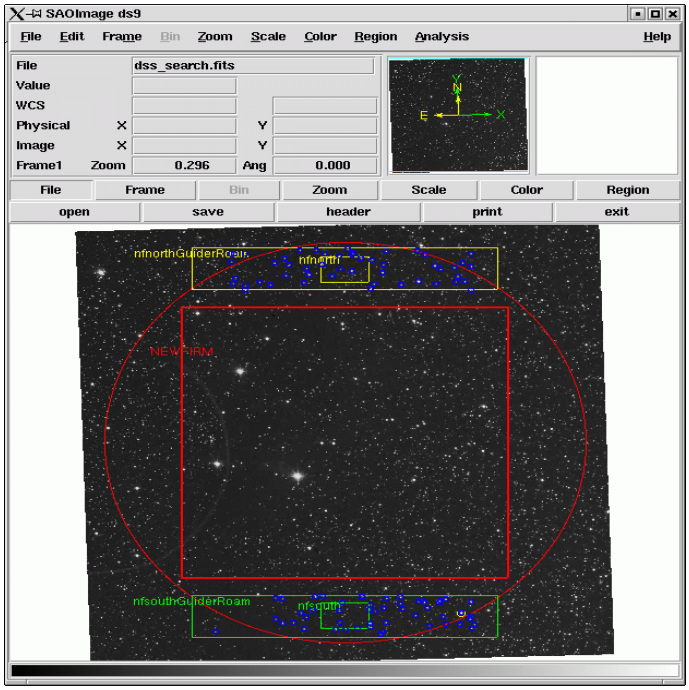

Figure 4. NEWFIRM guide-star selection dialog. The image was re-formatted to fit this page.

6. guider server commands Linux guider to start guiding;
7. guider server sends command completion status to initiator via `GWC` router;
8. Linux software guider sends guide corrections to `TCS` using remote procedure calls (`RPCs`);
9. guider server publishes status telemetry stream periodically throughout.

7. CLOSING REMARKS

The NEWFIRM software development project has taken several years to complete but places NOAO in a good position for future instrumentation projects since much of the code can be re-used. As is common with such large-scale instrumentation projects, however, we encountered several problems along the way—some of which we were able to resolve whilst others require further investigation:

Network Issues we see unexplained network glitches that interrupt the normal data flow. Although these glitches are non-fatal, they are annoying and may be related to the private data network setup where the `DHCP` lease expiry time is set at 90 minutes;

Hardware Issues the dedicated machines that run the NEWFIRM software are 64-bit architecture. This uncovered several bugs in (legacy) software packages so some effort was expended in fixing the 32-bit to 64-bit conversion. These problems were solved and have allowed other KPNO telescopes—using the same infrastructure software—to migrate to 64-bit machines;

Telescope the telescope requires some 'settle' time after an offset and this has resulted in putting empirical `sleep` commands within scripts that require dithering;

Guiding the telescope currently does not point well enough to always get the guide star in the (small) probe FOV automatically, thus we are still using the guider system in a manual mode for primary acquisition. Further, the SBIG units have to be controlled via a slow serial port connection which makes adjusting them tedious. The relatively complex guider hardware means that hardware failures become more likely (limit switch problems, compumotor controller and/or network glitches).

ACKNOWLEDGMENTS

The principal author wishes to thank Tony Farrell (AAO), head-honcho of `DRAMA` development, for astonishingly good, remote support of `DRAMA` on a foreign telescope.

REFERENCES

[1] Probst, R. G. et al., "First light with newfirm," in [*Ground-based and Airborne Instrumentation for Astronomy II*], McLean, I. S. and Casali, M. M., eds., *Proc. SPIE* **7014** (2008).

[2] Daggert, L. et al., [*NEWFIRM: Operating Concepts Definition Document*], NOAO Engineering and Technical Services Group, 950 N. Cherry Avenue, Tucson AZ 85719 (2001).

[3] Daggert, L. et al., [*NEWFIRM: Functional Performance Requirements Document*], NOAO Engineering and Technical Services Group, 950 N. Cherry Avenue, Tucson AZ 85719 (2001).

[4] Probst, R. G., [*NEWFIRM Scientific Use Cases for OCS Definition*], SDN9502 NEWFIRM Project, 950 N. Cherry Avenue, Tucson AZ 85719 (2004).

[5] Daly, P. N., [*NEWFIRM Observation Recipe Book*], SDN9006 NEWFIRM Project, 950 N. Cherry Avenue, Tucson AZ 85719 (2004).

[6] Davis, L., [*NEWFIRM Data Handling System: Quick Reduce Functionality*], SDN9501 NEWFIRM Project, 950 N. Cherry Avenue, Tucson AZ 85719 (2002).

[7] Autry, R. G. et al., "Newfirm: The widefield ir imager for noao 4-m telescopes," in [*Instrument Design and Performance for Optical/Infrared Ground-based Telescopes*], M. Iye, A. M., ed., *Proc. SPIE* **4841** (2003).

[8] Probst, R. G. et al., "Program status of newfirm, the wide-field infrared camera system for the noao 4-m telescopes," in [*Ground-based Instrumentation for Astronomy*], Moorwood, A. and Iye, M., eds., *Proc. SPIE* **5492** (2004).

[9] Daly, P. N., "Newfirm software—system integration using opc," in [*Astronomical Data Analysis Software and Systems XIII*], F. Ochsenbein, M. A. and Egret, D., eds., *ASP Conf. Series* **314** (2004).

[10] T. J. Farrell, J. A. B. and Shortridge, K., "Tcl/tk with drama: A natural for building user interfaces to instrumentation systems?," in [*Astronomical Data Analysis Software and Systems IV*], R. A. Shaw, H. E. P. and Hayes, J. J. E., eds., *ASP Conf. Series* **77** (1995).

[11] Gillies, K., [*Programmer's Guide to the Generic WIYN Client*], NOAO Mountain Programming Group, 950 N. Cherry Avenue, Tucson AZ 85719 (1996).

[12] Daly, P. N. and Pickup, D. A., [*NEWFIRM Observation Control System Book*], SDN9000 NEWFIRM Project, 950 N. Cherry Avenue, Tucson AZ 85719 (2004).

[13] Daly, P. N., [*Instrument Component Controller GWC Interface*], SDN9001 NEWFIRM Project, 950 N. Cherry Avenue, Tucson AZ 85719 (2004).

[14] Daly, P. N., [*NICS Drama Interface*], SDN9003 NEWFIRM Project, 950 N. Cherry Avenue, Tucson AZ 85719 (2004).

[15] Daly, P. N., [*NDGI Drama Interface*], SDN9005 NEWFIRM Project, 950 N. Cherry Avenue, Tucson AZ 85719 (2004).

[16] Daly, P. N., [*NOHS Drama Interface*], SDN9004 NEWFIRM Project, 950 N. Cherry Avenue, Tucson AZ 85719 (2004).

[17] Buchholz, N. C. and Daly, P. N., "The generic pixel server dictionary," in [*Advanced Software, Control and Communications Systems for Astronomy*], Lewis, H. and Raffi, G., eds., *Proc. SPIE* **5496** (2004).

[18] Buchholz, N. C. and Daly, P. N., "The monsoon generic pixel server software design," in [*Advanced Software, Control and Communications Systems for Astronomy*], Lewis, H. and Raffi, G., eds., *Proc. SPIE* **5496** (2004).

[19] Daly, P. N., [*NGUI: The NEWFIRM Graphical User Interface*], NEWFIRM Project, 950 N. Cherry Avenue, Tucson AZ 85719 (2008).

[20] Valdes, F. and Tody, D., "The noao mosaic data handling system," in [*Optical Astronomical Instrumentation*], D'Odorico, S., ed., *Proc. SPIE* **3355** (1998).

[21] Valdes, F. et al., [*The NOAO High Performance Pipeline System*], NOAO DPP Document PL001, 950 N. Cherry Avenue, Tucson AZ 85719 (2006).