The MONSOON Generic Pixel Server Software Design

Nick C Buchholz[*], Philip N. Daly
National Optical Astronomy Observatory, Major Instrumentation Group,
950 N Cherry Ave., Tucson, AZ 85719-4933, USA

ABSTRACT

MONSOON is the next generation OUV-IR controller development project being conducted at NOAO. MONSOON was designed from the start as an “architecture” that provides the flexibility to handle multiple detector types, rather than as a set of specific hardware to control a particular detector. The hardware design was done with maintainability and scalability as key factors. We have, wherever possible chosen commercial off-the-shelf components rather than use in-house or proprietary systems.

From first principles, the software design had to be configurable in order to handle many detector types and focal plane configurations. The MONSOON software is multi-layered with simulation of the hardware built in. By keeping the details of hardware interfaces confined to only two libraries and by strict conformance to a set of interface control documents the MONSOON software is usable with other hardware systems with minimal change. In addition, the design provides that focal plane specific details are confined to routines that are selected at load time.

At the top-level, the MONSOON Supervisor Level (MSL), we use the GPX dictionary, a defined interface to the software system that instruments and high-level software can use to control and query the system. Below this are PAN-DHE pairs that interface directly with portions of the focal plane. The number of PAN-DHE pairs can be scaled up to increase channel counts and processing speed or to handle larger focal planes. The range of detector applications supported goes from single detector LAB systems, four detector IR systems like NEWFIRM, up to 500 CCD focal planes like LSST. In this paper we discuss the design of the PAN software and it's interaction with the detector head electronics.

1. INTRODUCTION

In the past 18 years the authors have written, modified or used more than 12 separate and distinct hardware/software systems designed to control focal plane detectors and capture the data generated by them. When the limitations of current controllers forced NOAO to start a new controller development project, the software design began along with the hardware design. The MONSOON controller was, from its inception, designed to allow the control of any astronomical focal plane, such as IR FPA’s, CCD’s, orthogonal transfer arrays or OTA’s, etc. It was designed to be extensible so that the same set of electronics and software could be replicated to control massive homogeneous focal planes. A key feature of the MONSOON system design was that a single software system would be used to handle the various focal plane arrangements and detector types that had been proposed.

The MONSOON software interface is based on the Generic Pixel Server, or **GPX**, interface[1,2]. This interface was first discussed at the Nov. 2001 ACCORD conference in Santa Cruz. The interface describes the interaction between an image acquisition system or pixel server and external systems. The interface will be used for all new instruments, focal planes and/or detector types under consideration for development at NOAO. A proposal for the Generic Pixel Server interface definition was presented at the February 2002 AURA Software Conference held at the STSI in Baltimore, MD. Further development of that interface as NOAO ICD 4.0 was done in the spring of 2002 and a complete GPX interface, including a software library which implements the interface commands, has been developed in conjunction with the MONSOON project software effort. With very few exceptions the original interface definition presented in 2002 has held up to the rigors of actually writing code to control detectors and acquire data from IR and OUV detectors.

The MONSOON software design was guided by a set of requirements[3,4] designed to minimize the impact of hardware changes and maximize the flexibility and reusability of the software system. First, the software system needed to be configurable at runtime. This would allow the system to handle the details of each of the current and projected detectors and focal plane configurations. Second, the details of the underlying hardware and operating system should be confined to a small set of routines so that moving to new hardware would be as painless as possible. This included a decision to use operating system constructs that were common to a wide range of systems. Third, the system needed to

[*] Further information email: NCB: nbuchholz@noao.edu, PND: pnd@noao.edu

be flexible enough to handle new observing modes and methods without a major redesign and rewrite of the basic system. These goals have for the most part been achieved as will be discussed in later sections

2. MONSOON Functionality

MONSOON is an image acquisition system. Like a digital camera, the user configures the MONSOON system for image acquisition in a particular mode. The user then initiates an exposure or series of exposures and the captured images are archived for future reference. The user communicates with the system through the MSL, which handles the details of coordinating the PAN/DHE pairs to take the exposure. Because MONSOON is designed for massive focal planes the MSL does not handle the data at any time.

No provision is made for the control of telescope or instrument mechanisms. In addition, the image or data handling system deals with the details of archiving, displaying and processing the data sent to it by the MONSOON GPX. MONSOON has no information about the nature of the image nor does it know where the photons arriving on the focal plane originated. It provides meta-data to be stored with the image that gives its internal state, but information about telescope pointing, focal plane geometry, the nature of the image (spectra, sky dark, etc.) must be provided by the controlling Instrument or Observation control system.

The image and meta-data captured by the MONSOON software is passed to a data handling system (DHS) to be archived or stored. Normally, the DHS is provided as a shared library that is loaded at runtime into the MONSOON PAN software. The MONSOON program does provide a simple DHS library that writes a FITS image to the local disk.

3. MONSOON Hardware

The MONSOON system was proposed and designed to handle massive mosaic focal planes. It replaces the existing Arcon, Wildfire and SDSUII array controllers that cannot handle the requirements of these massive focal planes. The hardware design uses COTS technology wherever possible. The major subsystems are also designed to be easily extensible. Figure 1 below diagrams the MONSOON subsystems.

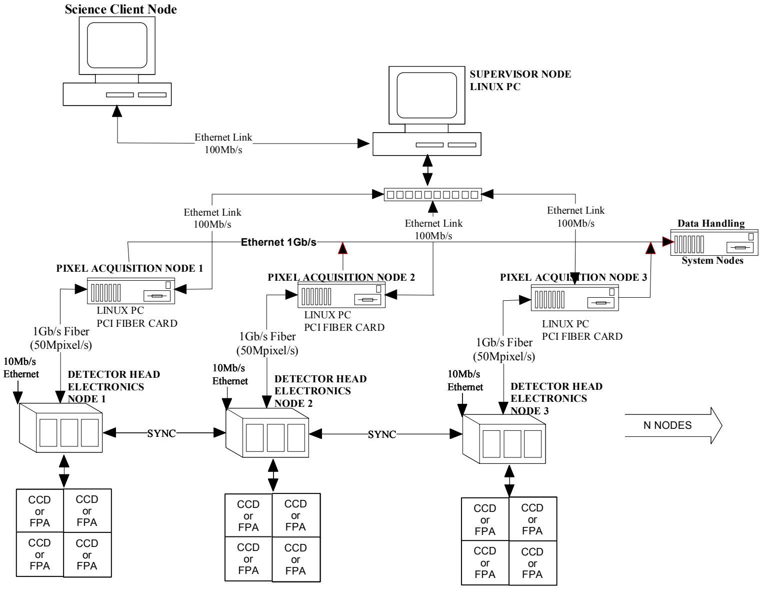

Figure 1. MONSOON Hardware Components

The basic building block of the MONSOON hardware system is the PAN/DHE pair. The Pixel Acquisition Node or PAN is a console-less, generic, commercial PC box running LINUX. The Detector Head Electronics is a commercial electronics enclosure containing the MONSOON compact PCI backplane carrying the circuit boards needed to interface to the section of the focal plane being controlled by the local PAN/DHE pair. Except for a synchronizing hardware clock, each PAN/ DHE pair only communicates with the MONSOON supervisor Layer (or MSL) and does not share information with other PANs. Communication between the PAN and DHE is over an industry standard bi-directional fiber interface capable of handling 100 to 240 Mbytes/s. In the MONSOON hardware implementation, data is sent for archiving over a gigabit ethernet connection to a Data Handling system (DHS) that determines its final disposition and processing. The MONSOON system does not include a local hardware display capability

The coordinating entity for the various PAN/DHE pairs is the MONSOON Supervisor Layer. This program also runs on a commercial PC running LINUX. The MSL handles the coordination of the PAN/DHE pairs, communications with other systems and the user and connection security. The MSL does not see or participate in the image data transfers, though it may send meta-data to the DHS for inclusion with the archived image data. (Note: that the supervisor level can run on one of the PANs or in an Instrument control system computer.)

4. MONSOON Data flows and Interfaces

Figure 2 below gives a diagram of how the MONSOON system interacts with other observatory systems. In particular it should be noted that the local status and DHS interfaces and the Science Client system are not part of the MONSOON program. These systems interact with MONSOON using, the GPX interface (NOAO ICD 4.0) for the command and response stream, the DHS Interface (NOAO ICD 1.0) for the image and meta-data stream and the Local Status Interface for the system status data stream.

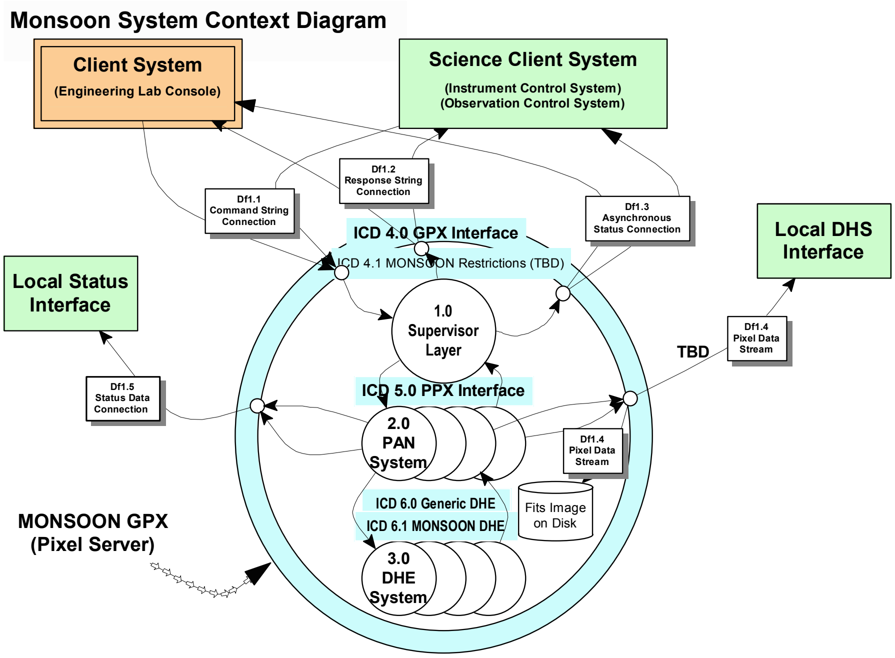

Figure 2. MONSOON Software Context

4.1. MONSOON GPX Interface

The MONSOON **GPX** interface generally follows NOAO Interface Control Document (ICD) 4.0 that describes the interface between software systems and combines the functionality of the detector controller and data pre-processor. It presents a common interface to the upper level observatory systems regardless of the underlying detector technologies or focal plane arrangement being used for observing. The **GPX** handles detector set-up, taking and controlling exposures and if one is provided, control of a local shutter. The **GPX** also allows for the archiving of small amounts of image data on a local disk in extraordinary circumstances.

5. MONSOON Software Design Philosophy

In addition to the science requirements imposed on the MONSOON system by its astronomical uses, the MONSOON software design was guided by a number of requirements that were more constraints on the process and scope than they were requirements for doing the science. The software was to be Open-Source and would make maximum use of facilities provided by the OS(LINUX>. It uses processes to perform identifiable sub-tasks and uses shared libraries to isolate and limit the scope of software functions.

A key requirement of the MONSOON design was flexibility and code reuse. Since we knew that the MONSOON system would be used in a wide variety of systems we wanted to be sure that we could add functionality quickly. This key feature was tested almost immediately. Soon after the completion of the design we were presented with the requirements for the orthogonal transfer array (OTA) device control. These requirements added a level of complexity to the device control that the original design did not include. A preliminary evaluation of the requirements and MONSOON design has shown that with only a modest additional effort the design can handle the increased complexity[7].

Another key feature of the design was that the software be usable in a wide variety of applications at different institutions. It was our goal to provide a software system that could be easily adopted by any observatory. In order to avoid trying to impose any control philosophy on other institutions the MONSOON software does not include certain facilities. First, while there is a MONSOON engineering interface provided, most user interface tasks are delegated to the observatory staff. The MSL and PAN interfaces are software interfaces, not GUI's. Some support for GUI interfaces is provided, however. Second, a number of systems are unique to specific observatories (EPICS, DRAMA, the MPG/WIYN Router, etc); the interface to these systems is also left to the observatory staff. The nature of the MONSOON system allows multiple connections to the system and can accommodate connections from multiple clients.

6. MONSOON system Layering

Both the MONSOON software and hardware are built in isolated layers. The interactions between the layers are defined by a series of ICD's that describe the interfaces. In general the ICD's come in two types: the generic interfaces (ICD 1.0, 4.0, 5.0, 6.0) give a set of generic functions that are used to interact with the layer, The hardware or system specific ICD's (ICD 4.1, 6.1, 7.1) give details about the specific implementation used by the MONSOON hardware or software system. Easily modified shared libraries implement the various ICD's and conceal the details of the hardware or system interface. This layering is part of the reason the MONSOON system remains flexible. Figure 3 is an example of how the ICD's and layering can be used to implement an SDSU-II or other version of the MONSOON system.

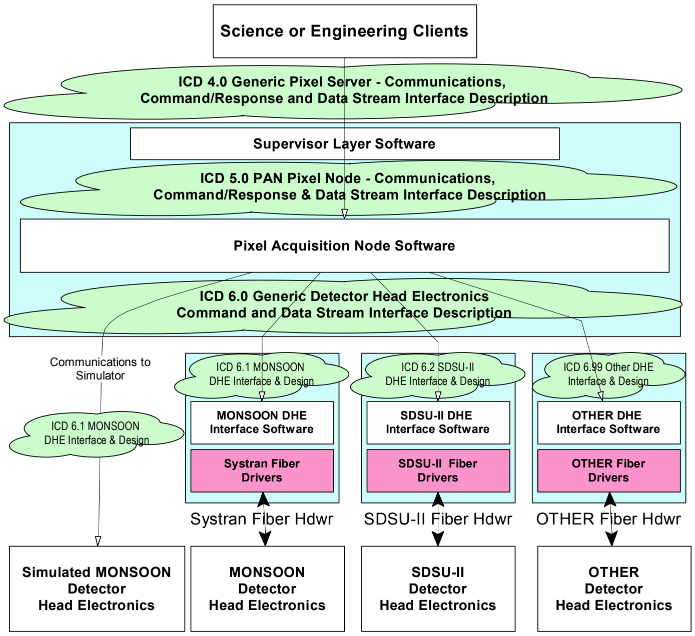
Science or Engineering Clients
ICD 4.0 Generic Pixel Server - Communications, Command/Response and Data Stream Interface Description
Supervisor Layer Software
ICD 5.0 PAN Pixel Node - Communications, Command/Response & Data Stream Interface Description
Pixel Acquisition Node Software
ICD 6.0 Generic Detector Head Electronics Command and Data Stream Interface Description
Communications to Simulator
ICD 6.1 MONSOON DHE Interface & Design
ICD 6.1 MONSOON DHE Interface & Design
ICD 6.2 SDSU-II DHE Interface & Design
ICD 6.99 Other DHE Interface & Design
MONSOON DHE Interface Software
SDSU-II DHE Interface Software
OTHER DHE Interface Software
Systran Fiber Drivers
SDSU-II Fiber Drivers
OTHER Fiber Drivers
Systran Fiber Hdwr
SDSU-II Fiber Hdwr
OTHER Fiber Hdwr
Simulated MONSOON Detector Head Electronics
MONSOON Detector Head Electronics
SDSU-II Detector Head Electronics
OTHER Detector Head Electronics

Figure 3. Monsoon Software Layers

In addition to allowing the flexibility to accommodate changing hardware, the layering also allows the effect of hardware changes to be confined to a small subset of the shared libraries used in MONSOON. The specifics of the fiber interface hardware are contained in the communications hardware library (libcomHdw). Likewise the DHE hardware specifics are contained in the DHE hardware library (libdheHdw). No information about the hardware specifics needs to be propagated above the generic DHE library (libdheUtil)

7. MONSOON Supervisor Layer Structure

MONSOON systems that contain multiple PAN's will require a MONSOON Supervisor Layer or MSL[8]. This system will become the common interface to a multi-PAN MONSOON system deployed for science operations. The MSL is a command and control layer, no pixel data will flow through it. The MSL may provide some pixel data flow control if required by the DHS library. It provides access to the MONSOON system through the GPX interface to client software and provides a single point of access to the focal plane. It provides multiple client connections and access security to the science clients to prevent unauthorized manipulation of the focal plane.

The MSL is the system level that will handle sending commands to multiple PAN's and gathering the responses from multiple PANs and summarizing them for the science client. It is planned that the MSL will provide for error monitoring & recovery for the PANs and reporting of such to the science client.

The MSL may be configured to run on a separate machine, on one of the PAN's or on the Instrument or OCS computer. Communications to the PAN's from the MSL will be over the general observatory ethernet link.

8. MONSOON PAN Process structure

The largest and most complex component of the MONSOON system is in the PAN software. In order to keep the flexibility needed to accommodate multiple detectors and focal plane arrangements the PAN software is constructed as a set of cooperating independent processes that isolate the various functions of the PAN. Communications to the PAN software is through a modified GPX interface, called PPX (NOAO ICD 5.0), a set of commands and responses that allow an engineering interface or MSL to control the PAN system.

The PAN software consists of four permanent process and two temporary processes. "panDaemon", "panCapture", "panProcAlg" and "panSaver" are permanent processes started by a shell script and required for the PAN to take data. "extTrigger" and "fSaver" are temporary helper processes started by panCapture and panSaver respectively. The processes are internally constrained to start up in the correct order. "panDaemon" starts first and reads the configuration records (files at the present time) for the system[6]. It creates the shared memory buffers needed by the processes and fills the Attribute-Value and command tables from the configuration records. The other processes wait for the shared memory spaces to be created and initialised and then start their own initialisation procedures. This includes mapping the shared memory into the local address space and setting up any local resources the process controls.

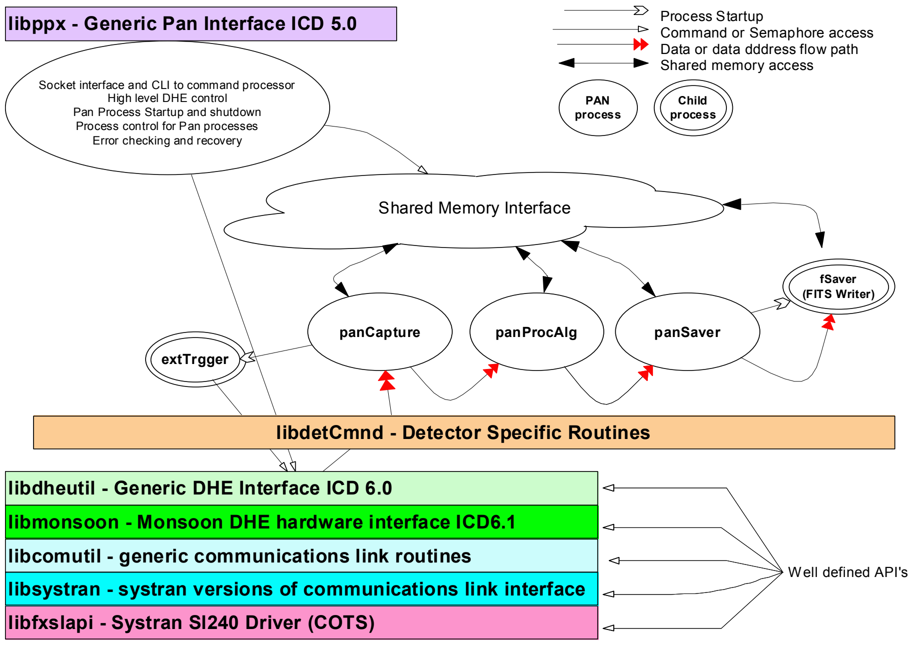
libppx - Generic Pan Interface ICD 5.0
Process Startup
Command or Semaphore access
Data or data dddress flow path
Shared memory access
PAN process
Child process
Socket interface and CLI to command processor
High level DHE control
Pan Process Startup and shutdown
Process control for Pan processes
Error checking and recovery
Shared Memory Interface
fSaver (FITS Writer)
panCapture
panProcAlg
panSaver
extTrgger
libdetCmnd - Detector Specific Routines
libdheutil - Generic DHE Interface ICD 6.0
libmonsoon - Monsoon DHE hardware interface ICD6.1
libcomutil - generic communications link routines
libsystran - systran versions of communications link interface
libfxslapi - Systran SI240 Driver (COTS)
Well defined API's

Figure 4. PAN Process Structure

The PAN processes are set up as a chain of producer/consumer processes which interact through the shared memory spaces, through semaphores and through image buffer queues. Facilities for additional communications through peer-to-peer socket connections were considered in the original design but have not been implemented at this time, as we have not needed that level of communication thus far.

Figure 4 diagrams the inter-process communications used by the PAN software and shows that three techniques are used to pass data between the processes. Shared memory segments are the most frequently used method for transferring information between the PAN Processes. Whenever several processes must simultaneously use data the shared memory technique is used. Queues are used for passing data buffers between processes. These FIFO style queues allow the order information inherent in the data capture process to be maintained across processes. Each queue is mirrored by a semaphore that allows the processes to avoid polling to determine if there is something to do.

Processes wait on semaphores that are given by the producer and taken by the consumer. E.G. At the start of an exposure the "panCapture" process waits for data to arrive from the DHE. When it does, "panCapture" puts the buffer on the "fullRawBufferQ" queue and gives the rawBufReady semaphore. "panProcAlg" has been waiting on the rawBufReady semaphore and is now able to take the buffer off the queue, process it and pass it on in the same way to "panSaver".

8.1. Shared Library Usage in MONSOON

An important design feature of the PAN processes and the MONSOON design in general is the extensive use of shared libraries to implement functionality not provided by the OS. Each process loads and uses a set of shared libraries that implement shared memory, queue, semaphore and socket operations, Attribute-Value setting and query, hardware interaction operations and the PPX and GPX interfaces. The use of shared libraries insures that all processes on the PAN get the same version of the libraries and have a consistent view of their interactions and functionality. These libraries are common to all MONSOON systems and have been thoroughly tested before release. The queue and socket libraries have been in use at NOAO for over ten years.

Shared libraries are also used to isolate functionality unique to a specific Focal plane or detector type. When the system is started it loads the library designed for the system being started. Four classes of system specific libraries exist and as many more as are needed can be added to the system.

The system specific libraries are needed when a system is built which does not fit into one of the standard pre defined systems. We expect that all MONSOON systems using the NOAO DHE hardware and the Systran fibre communications boards will use the same libdheHdw and libcomHdw libraries. Likewise it is possible that the generic detector library will be used for most CCD systems and the DHS FITS library will be sufficient for most testing. A carefully thought out "makefile" system for these system specific libraries allows a new system to change only those files where differences from a base system occur. It is our intension to extend this system so that a system can be built using inheritance; this means a system could be built by starting with the generic routines, modifying those by using routines from system A and modifying those by using routines from system B.

8.1.1. The DHE hardware libraries

The DHE hardware libraries (called libdheHdw), handle hardware differences between DHE designs. This library is called through a set of generic routines contained in libdheUtil and understands the details of the hardware protocol used by a particular DHE type. At this time two versions of the DHE hardware library exist. One supports a simulated DHE within the PAN processes. The second library supports the MONSOON DHE version currently in use. We have already used this multiple library facility to implement an expansion of the PAN/DHE communications protocol required to support the OTA devices being developed for QUOTA and ODI.

A third version of the libraries to support the SDSU II DHE hardware has been designed but not implemented.

8.1.2. The Communications Hardware Libraries

The communications hardware libraries (called libcomHdw), handle differences between different interconnection hardware. These routines are also called through a set of generic routines (libcomUtil). The libcomHdw library understands the details of the communications protocol and the underlying communications hardware. We have implemented two versions of this library as well. A simulation version of the library essentially provides a loop back version of the communications link. The production version uses the Systran hardware and software to communicate with the MONSOON DHE. The library libsystran makes calls to and understands the COTS device driver and hardware interface library provided with the Systran SL100/SL240 Fiber Extreme communications link. This link uses an industry standard fibre channel hardware interface.

Two other communications hardware libraries are planned. The first will support the SDSU II style of fiber interface. The second will use a private point-to-point ethernet connection to communicate with a DHE.

8.1.3. The Data Handling System Libraries

One of the classes of libraries included in the configurable library set was added to accommodate use of MONSOON at a wide variety of institutions. The Data Handling System or DHS libraries (called libdhsUtil), provides the data archiving facilities for the MONSOON system. Since there are at the various observatories a number of DHS system already in place, we decided MONSOON should be able to interface with each of them. To accomplish this a DHS API[5] was developed which treats the DHS system as just another device for data storage. The API provides for a set of routines, which are written by the local observatory staff, to implement open, close, configure and write functionality to the local DHS system. In this way it is possible to handle different Data archiving methods without modifying the base panSaver process.

Two DHS libraries have been written, libdhsNULL implements a /dev/null style of data saving. That is, it immediately discards the data and returns success. This library has been useful in providing a method of testing that does not fill the limited PAN disk space and helps determine PAN data throughput rates. The second, libdhsFITS, uses the data to write standard multi-extension FITS file onto the local PAN disk. This library not only provides a template for future DHS libraries but also provides a method for continuing observations in the event of a failure of the local DHS hardware or ethernet.

NOAO's Data Products Division is currently building a DHS library for NEWFIRM science operations.

8.1.4. The Detector Hardware Libraries

The fourth configuration library class is the detector hardware library (called libdetCmnds). This library will be the place most of the changes to a MONSOON will be made since this library handles the differences between different detector types and focal plane arrangements. Differences between IR and CCD data capture and pre-processing will be handled here. Also additional functionality for future detector types can be added within this library.

Three detector libraries have been developed and are the basis of the hardware and software testing and detector development and testing being carried out at NOAO. The first, the generic detector library was developed to be suitable for hardware board testing and general software development. This library will eventually be expanded into a CCD version that handles CCD mosaics.

The second detector library is an Aladdin III library. This library implements the control of Aladdin III detectors such as those used in NIRI and GNIRS and was used to take first light data in 2003. The third library is an Orion II detector library. This library is being used to take data in the evaluation of the Orion II InSb detectors destined for NEWFIRM. Detector libraries for a generic CCD, the OTA testing LAB, and the NEWFIRM 2x2 Orion II focal plane are currently under development and testing.

8.2. Additional Configuration Flexibility

In addition to the flexibility in handling hardware differences provided by the detector hardware shared libraries there is another level of configuration flexibility built into the MONSOON system. In the case that a focal plane or detector type is so different that the standard method of data capture, processing and archiving will not work, it is possible to extend or amend the system by adding or deleting processes to deal with the variation. A wave front sensor application for MONSOON might contain only a data capture and processing algorithm without an archive process. The ODI application may contain two data capture processes, one that handles the details of the guiding, centroid calculation and charge shifting and one which handles the Science data capture and processing.

This ability to change the processing, capture and archiving methods is included in the base design and may be added in such a way as to allow on-the-fly reconfiguration of the process set during a run. (Note: this facility was discussed and included in the design but has not been and may not be implemented.)

8.3. System Start-up

The key to the MONSOON system is the automated system start-up. Included in the hardware design and provided for but not implemented in the current software, is the ability for a PAN to read a serial identifier chip in the

MONSOON DHE hardware. This chip can be implemented in the detector specific configuration and protection hardware or in the DHE itself. The identifier is then used to determine the appropriate configuration records and start the PAN software with the correct configuration. The current start-up script uses a command line argument to determine the correct configuration.

The start-up script determines from the configuration record the required configuration files. It calls the focal plane configuration script that customizes the configuration directory. It then reconfigures the shared library load path and library directory to the correct settings for the current focal plane, hardware and site. The various processes are then started and focal plane set-up can begin. Using the gpxSetMode, and gpxSetMemCfg commands the system is brought to readiness for data taking.

9. Building the software

The MONSOON software development effort has made extensive use of automated software tools for building the system components. We have used CVS for version control and GNU gcc/gmake to build the systems. A set of near identical makefiles is used to build the hardware libraries, the Utility libraries and the Application processes. In the early development we frequently saw several developers working on the same sections of code. By using these techniques we have been able to minimize the problems seen in multi-developer efforts.

An additional help in the code development is the inclusion of an automated system for generating the API associated with the MONSOON libraries. TeX and LateX are used to generate the API document directly from embedded comments in the library source code.

```
/***************************************************************************
 *
 * __doc__ \section {The queUtil <<VERSION>> Library}
 * __doc__ \subsection {queUtil.h}
 * __doc__ \begin{description}
 * __doc__  \item[\sc use:] \emph{\#include ``queUtil.h''}
 * __doc__  \item[\sc description:]  this file contains all common code
 * __doc__    required by the functions needed to build the static
 * __doc__    and dynamic queUtil libraries. These libraries
 * __doc__    abstract the queue, dequeue and stack interface to the system.
 * __doc__  \item[\sc argument(s):]  not applicable
 * __doc__  \item[\sc return(s):] not applicable
 * __doc__  \item[\sc last modified:] Monday, 4 November 2002
 * __doc__ \end{description}
 *
 **************************************************************************/
```

Figure 5. API Documentation Fragment

An installation system for distributing source code using tar, gzip, etc., has been developed and eventually we expect to have an Open Source version of the source code available under CVS. Additionally a system for binary updates on multiple PAN's has been implemented using rsync and ssh. We now keep eight PANs on two continents updated from the main source distribution whenever required.

10. Handling Legacy systems

From the beginning it was hoped that the MONSOON system software would be used to upgrade existing systems that use older array controllers. Proposals ranging from complete hardware replacement to software upgrades have been studied. The authors believe that the MONSOON software could be retrofitted to control older hardware in as little as two man-weeks. The changes required for such a change over would be concentrated in the libdheHdw and libcomHdw libraries. Additional work would be needed in the libdetCmnds library if the detector type did not match one of the existing MONSOON detectors. The creation of a detector specific library also takes about two man-weeks. However given the current man power availability and the number of MONSOON systems being developed at NOAO over the next year it is unlikely that any older systems will be converted in the near future.

ACKNOWLEDGEMENTS

We would like to acknowledge the members of the MONSOON team for their efforts refining the software requirements and of course in testing the MONSOON software. We especially commend the tolerance of the hardware engineers in testing the concepts used in the final software while trying to develop and debug the MONSOON hardware using a hastily written PAN/DHE software program developed in less than a month at the start of the MONSOON development and then ignored as far as possible by the software developers. We also ask their understanding as we move into real operations with the MONSOON software and they struggle to learn another method of interacting with the detectors.

REFERENCES

1. Nick C. Buchholz, Phil N. Daly, Barry M. Starr – 2002 - *NOAO Interface Control Document 4.0 - Generic Pixel Server Communications, Command/Response and Data Stream Interface Description.*
2. Nick C. Buchholz, Phil N. Daly, 2004, *The Generic Pixel Server Dictionary,* Proc. SPIE Vol. 5496, Advanced Software, Control and Communications Systems for astronomy, Hilton Lewis, Gianni Raffi, Eds. (this volume)
3. MONSOON Project Team – 2003 - *MONSOON Image Acquisition System (Pixel Server) - Functional and Performance Requirements Document (FPRD)* – Initial Draft.
4. N. C. Buchholz, P. N. Daly – 2003 – *MONSOON Software PDR - Requirements, Software Architecture & Design* – power point presentation.
5. N. C. Buchholz, G. Chisholm, P. N. Daly, P. Ruckle – 2004 - *Interface Control Document 1.0 - Data Handling System Interface - Status and Data Stream Transfers* - Draft
6. P. N. Daly, N. C. Buchholz and P. Moore– 2004 – *Automated Software Configuration in the MONSOON system* - Proc. SPIE Vol. 5496, Advanced Software, Control and Communications Systems for astronomy, Hilton Lewis, Gianni Raffi, Eds. (this volume)
7. D. Sawyer, P. Moore, G. Rahmer, and N.C. Buchholz – 2004 *Orthogonal Transfer Array Control Solutions Using the Monsoon Image Acquisition System* - Proc. SPIE Vol. 5499, Optical and Infrared Detectors for Astronomy, James Beletic, James D. Garnett Eds.
8. P. N. Daly and N. C. Buchholz– 2004 – *The Monsoon Implementation of the Generic Pixel Server* - Proc. SPIE Vol. 5496, Advanced Software, Control and Communications Systems for astronomy, Hilton Lewis, Gianni Raffi, Eds. (this volume)