SPIRou High-Level Instrument Control Command
SPIROU-4900-CFHT-RP-00590-V1.0

Prepared by: Tom Vermeulen	**Date**: Sept. 17, 2012

Keywords	Control/Command, Software, Hardware, Guider, Cassegrain, Spectrograph, DetCom, QSO, Agent, Status Server, Director, Phase 1, Phase 2, TCS

Summary	This document presents a preliminary design for the SPIRou high-level software that will be used for data acquisition, instrument control, and instrument monitoring.

Version	Date	Comments
0.9	Sept 12, 2012	Preliminary release.
1.0	Sept 17, 2012	Incorporated changes based on CFHT internal release comments.

Table of Contents

List of Figures

1 Purpose

The purpose of this document is to provide a preliminary design for the SPIRou high-level software that will be used for data acquisition, instrument control and instrument monitoring.

2 Scope

This document is focused primarily on the areas of software which are the responsibility of CFHT to either modify or design and create. Areas of the software which are under the responsibility of other groups are covered within other preliminary design documents. The CFHT software infrastructure which will be used for SPIRou is based on the New Environment for Observing (NEO) infrastructure already in place for MegaCam, WIRCam, and ESPaDOnS.

The document defines a high-level design for aspects of the software related to the instrument and telescope control. The impacts of changes to other areas of software are identified and will be addressed in more detail within future design documentation.

3 Overview of Observing at CFHT

This section provides a high-level overview for how observing is currently performed at CFHT.

3.1 Queued Service Observing (QSO)

Almost all the instruments in use at CFHT are used in a Queued Service Observing (QSO) mode. This means that Principle Investigators (PIs) that are granted observing time by the Time Allocation Committee (TAC) submit their observing programs via a web-based Phase 2 submission tool. CFHT matches the constraints of the PI programs against the sky conditions and observes on behalf of the PIs the highly ranked programs that best match the sky conditions while maintaining an effective balance between funding agencies. Once observations are complete, the data is reduced and distributed to the PIs. The QSO model of observing is also the model of observing which will be used with SPIRou. Figure 1 illustrates how this model works. The bulk of this document will focus on the software within the box labeled “Perform Observation”.

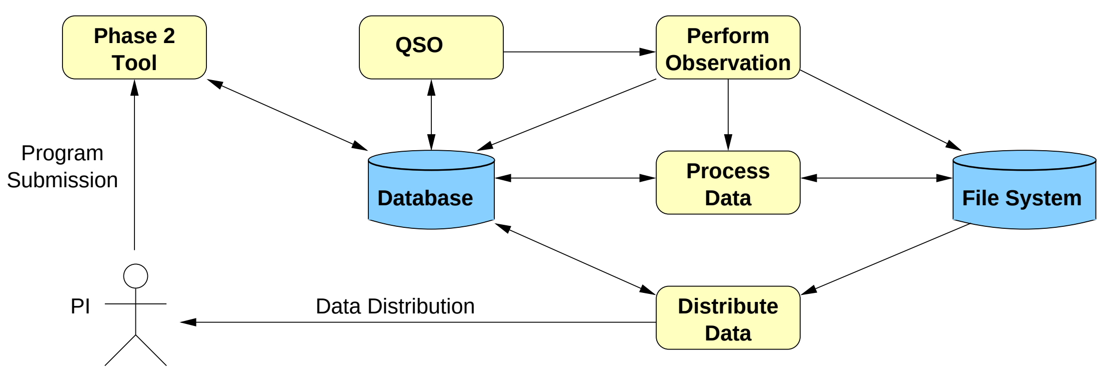

Figure 1: QSO Observing Model

3.2 Remote Observing

CFHT is operated remotely from Waimea at night with no on-site night time staff at the summit. In addition, it is not unusual to have three to four consecutive days without CFHT staff present at the summit. SPIRou must be designed to accommodate this mode of operation and require no on-site intervention as part of normal operations. Figure 2 shows a picture of the remote observing room in Waimea.

Figure 2: Picture of the Waimea Remote Observing Room

Toward the left of the screens shown in figure 2 is a four-headed display that is used to display the graphical user interfaces (GUIs) used to control the instrument and perform the observations. This four-headed display will also be used to display the GUIs used for observing with SPIRou. Figure 3 shows a screenshot of the GUIs used for observing with WIRCam.

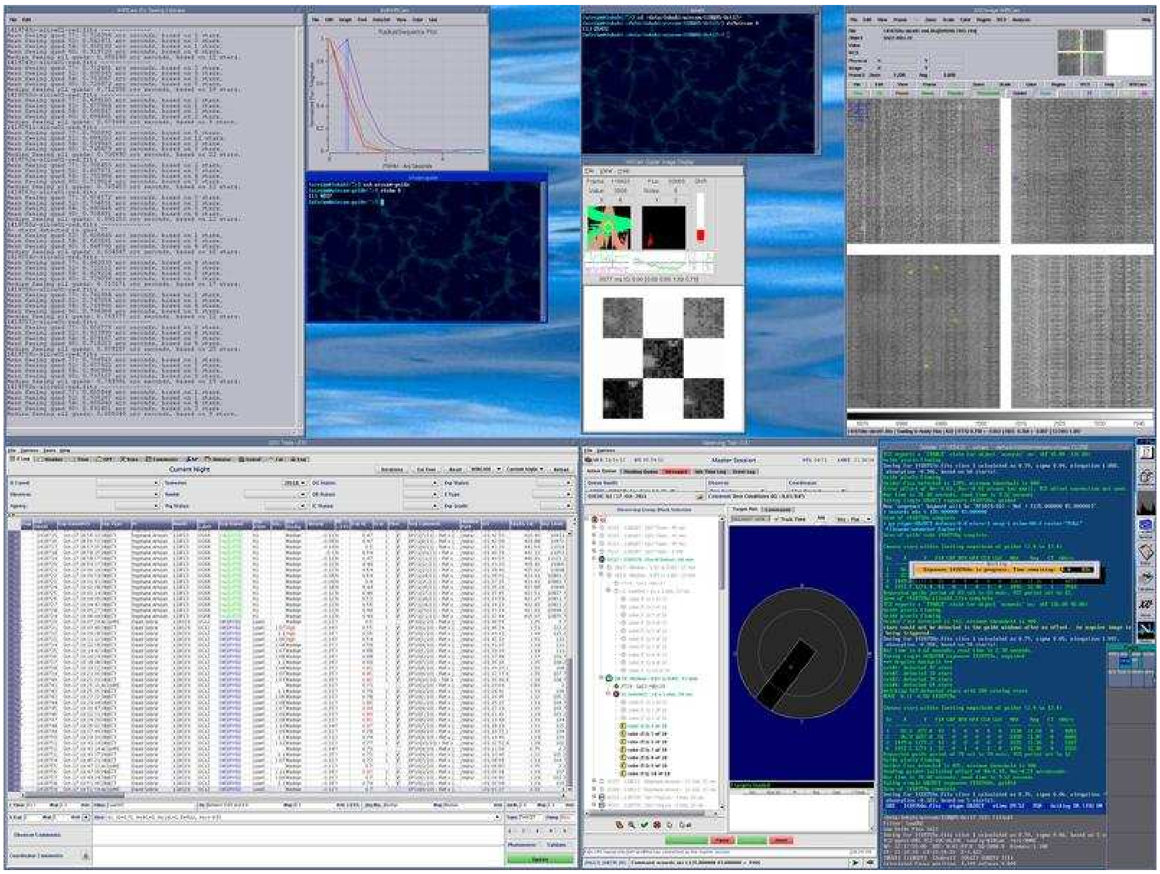

Figure 3: Screenshot of Observing Software for WIRCam

4 Software Design

4.1 Software Architecture

Figure 4 illustrates the proposed SPIRou software architecture spanning the full life-cycle from PI proposal submission to data distribution. Of particular note is the subsection illustrating instrument and telescope control. Boxes shaded in green indicate software designed and developed by groups outside of CFHT and boxes shaded in yellow indicate software designed and developed within CFHT. Boxes shaded in blue indicate software that is designed and developed within a close collaborative effort between both CFHT and outside organizations.

The following sections outline the basic details regarding each of the overall components and a high-level overview of the changes that must be made to accommodate SPIRou.

4.1.1 QSO PI Utilities

These are the web-based software components PIs interact with. At this point, CFHT has yet to go through a full analysis of the impacts and changes necessary to support SPIRou. The design process for these changes will move forward within CFHT once SPIRou progresses beyond the PDR phase. We expect the observing modes of the instrument and constraints of the system to be well defined following the PDR. Once this is well defined, it will become easier to assess the changes required within the following software utilities.

4.1.1.1 Phase 1 Proposal Submission Tool

Changes will be necessary to support instrument specific parameters associated with SPIRou PI proposal submissions.

4.1.1.2 Phase 2 Program Submission Tool

The Phase 2 tool must be modified to support SPIRou programs and the observing modes and constraints specific to the instrument.

4.1.2 QSO High-Level Control

4.1.2.1 QSO Database

The internal structure of the database as well as data within the database will be modified in order to support SPIRou. These changes will largely be driven by the science requirements of the PIs and the various operating modes supported with the instrument.

4.1.2.2 QSO Observing Tool

The QSO Observing Tool must generate commands specific to what is required by the data acquisition and instrument control aspect of SPIRou. Changes will be necessary to make sure these commands make their way to the instrument session.

4.1.2.3 Command Server

The Command Server is responsible for receiving commands from the QSO Observing Tool and interacting with the director session in order to handle the execution of the commands. No changes to the Command Server for SPIRou should be required for SPIRou.

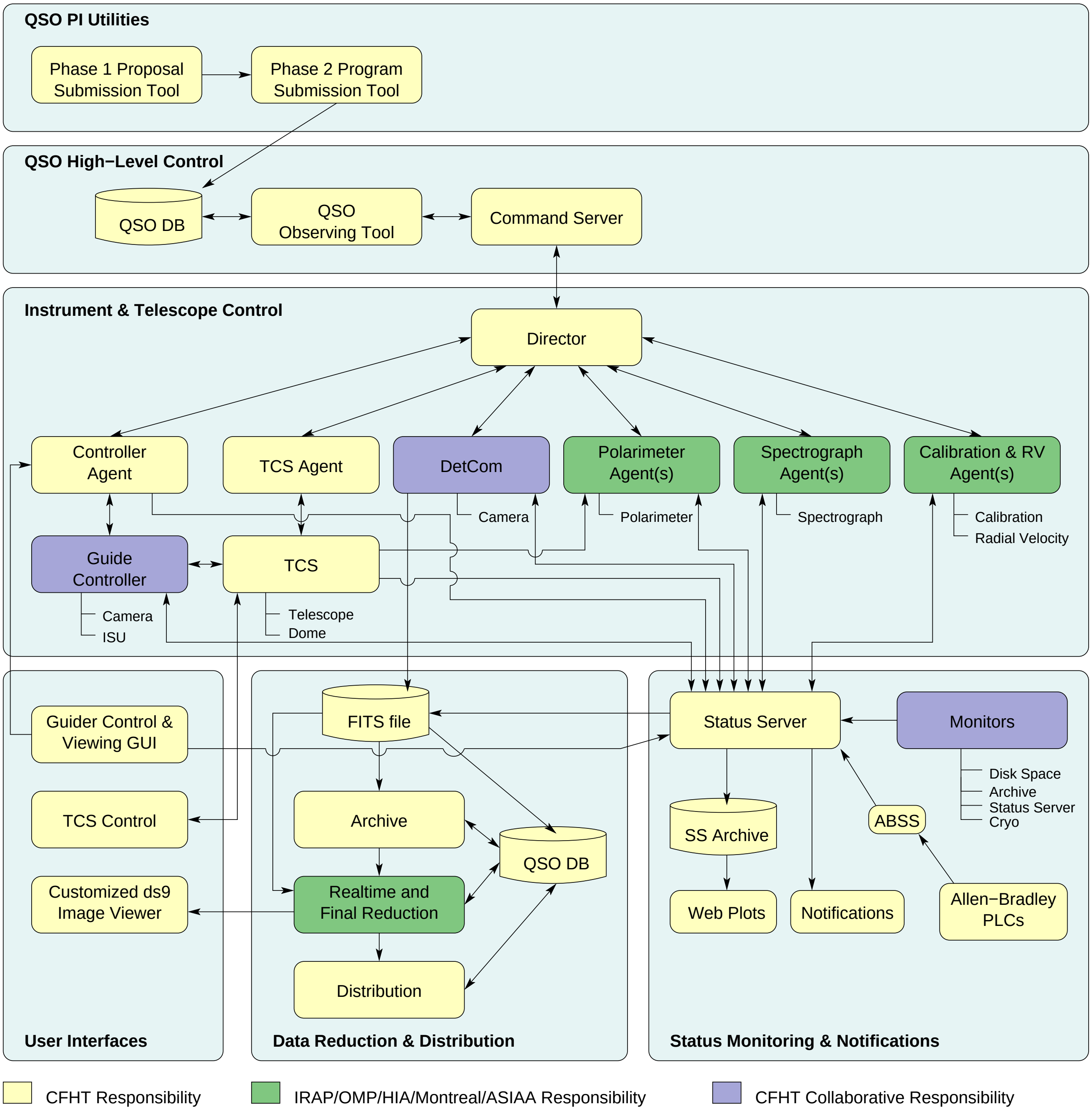

Figure 4: Overall Software Architecture for SPIRou

4.1.3 Instrument and Telescope Control

This area of the software is where much of the effort for SPIRou lies. The specific design details for each of the components designed outside of CFHT should be available within the preliminary design documentation for each subsystem, so they won't be covered in this section. Instead, this provides a general overview of each of the components.

4.1.3.1 Director

The central component within the instrument control and data acquisition layer is a program called “director”. Running under director are “agents”, possibly on other computers, which do instrument and camera control. Each command is passed on to the appropriate agent, the agent carries out the command, and director returns a completion pass-or-fail indication. This process is single threaded. Figure 5 shows a screenshot of director while observing with WIRCam. More details regarding director are available at http://software.cfht.hawaii.edu/director.

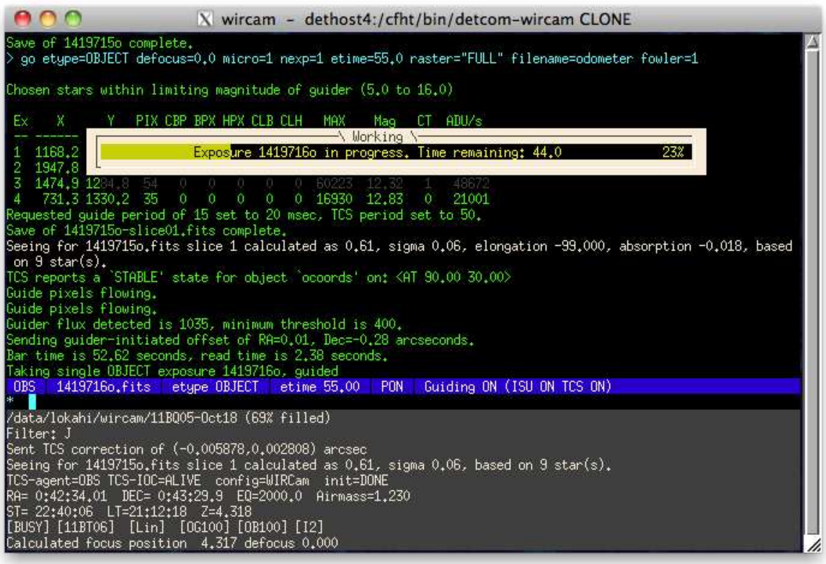

Figure 5: Screenshot of Director

At this point, no changes to the director software are anticipated for SPIRou.

4.1.3.2 Controller Agent

This is a new agent that must be implemented for SPIRou. The role of the controller agent is to take single line instrument configuration and exposure commands and ensure operations are executed efficiently. The controller agent communicates with other agents, via director, in order to efficiently configure the instrument prior to exposure initiation. Section 4.2 illustrates how this works using the ESPaDOnS controller agent as an example.

4.1.3.3 TCS Agent

The TCS agent is responsible for handling commands used to define the target position of an object on the sky. Upon receipt of a positioning command, the TCS agent will ensure that the telescope and dome slit are properly positioned with sufficient accuracy so the requested target from the PI can be placed within the hole of the pierced mirror by the acquisition and guiding subsystem.

At this point, no changes to the TCS agent software are anticipated for SPIRou.

4.1.3.4 DetCom Agent

DetCom is a camera control agent used to control the science detector(s), create the image FITS file, and initiate FITS header creation. This agent is typically running on a dedicated Linux host separate from the host that is running director. Due to the single threaded nature of director, while a science exposure is being taken, director is blocked from handling any other commands. This is important in order to prevent any changes to the instrument configuration or telescope position during an exposure.

SPIRou will require the development of a new DetCom agent in order to control and retrieve pixel data from a Teledyne H4RG detector. Figure 6 illustrates a block diagram with the primary components associated with the SPIRou DetCom agent.

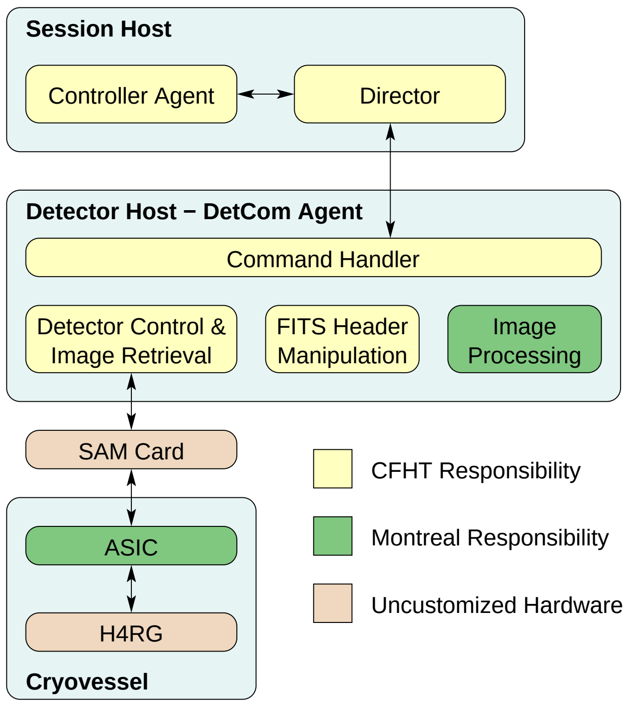

Figure 6: SPIRou DetCom Architecture

4.1.3.5 Polarimeter, Spectrograph, Calibration, and Radial Velocity Agents

Individual director agents will be created to control and monitor the hardware within the polarimeter, spectrograph, calibration box, and radial velocity box. The design of both the hardware and software for these sections can be found within the preliminary design documents produced by IRAP/OMP, OHP, and HIA.

4.1.3.6 Guide Controller

Target acquisition, pointing correction, and guiding is handled by the guide controller. The guide controller is a process running on the guide host which communicates directly with the controller agent on the session host via a socket connection. Figure 7 illustrates the guiding architecture for SPIRou. More details can be found in the SPIROU-3620-CFHT-TN-00243_High-level Tip/Tilt Guiding Software PDR document.

4.1.4 User Interfaces

SPIRou will require both newly created user interfaces as well as modifications to existing user interfaces. The interfaces discussed in this section are only the interfaces used for observing with SPIRou. Changes will also be necessary to the user interfaces within the QSO suite of utilities (Phase 1, Phase 2, and QSO tools).

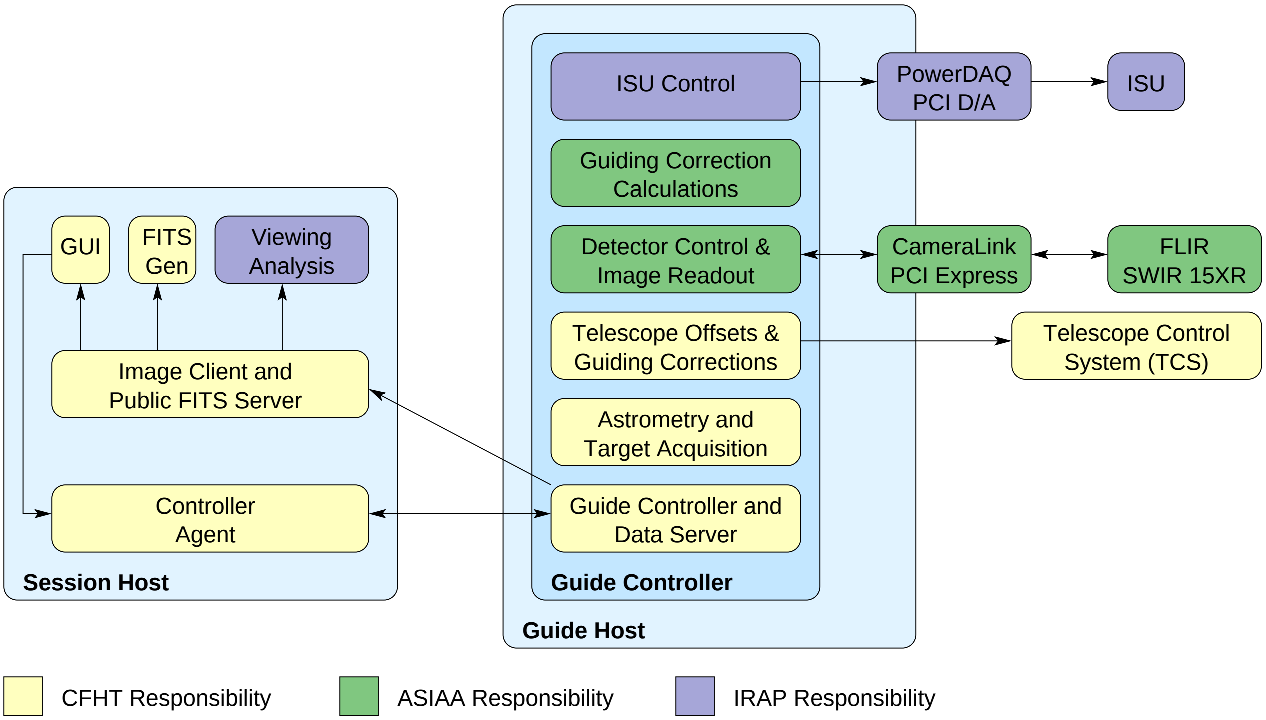

Figure 7: SPIRou Guiding Architecture

4.1.4.1 Guider Control and Viewing GUI

This interface is used to visually display what is happening with the guider as well as to provide overall target visualization. With ESPaDOnS target acquisition and guiding is a manual process performed by the observer. A screenshot of the GUI used by the observer for ESPaDOnS is shown in figure 8. Obviously, the GUI developed for SPIRou will be different. However, there is a base of software available at CFHT to build upon for image rendering and target selection.

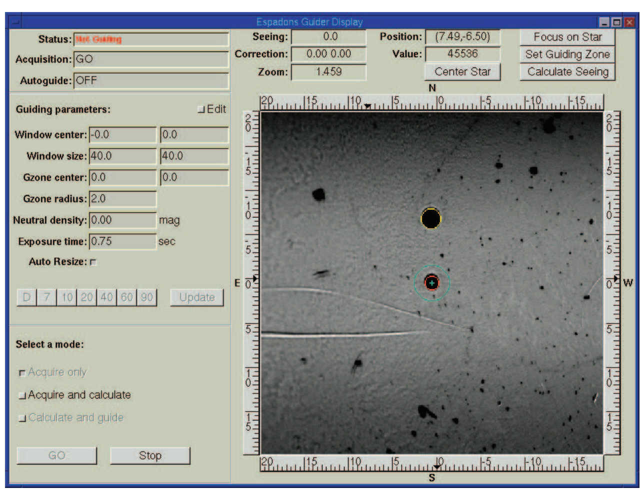

Figure 8: ESPaDOnS Guider Screenshot

In order to optimize the efficiency of the instrument the plan for SPIRou is to automatically identify the target specified by the PI, place it within the fiber hole, and initiate guiding. The GUI will allow the user the ability to manually select the target in case the automated process isn't able to definitively identify the target.

4.1.4.2 TCS Control

A number of interfaces are available for the observer to manually control the positioning of the telescope, dome, and other subsystems within the observatory. Some changes to the TCS user interfaces will likely be required to support SPIRou. Once SPIRou moves beyond the PDR phase, a more detailed investigation will be performed within CFHT to identify the SPIRou specific impact to the TCS control interfaces.

4.1.4.3 Customized SAOImage DS9 Image Viewer

Each of the existing instruments at CFHT utilize the SAOImage DS9 image viewer to display science content for the observer. In each case, CFHT has developed TCL customizations to more closely tailor the image viewer to work closely with the science content for each instrument. This makes it much easier for the observer to evaluate whether the acquired data is within the constraints specified by the PI.

CFHT will most likely create a customized DS9 to display the real-time reduced SPIRou spectral images much like what is currently done with ESPaDOnS.

4.1.5 Data Reduction and Distribution

All instruments operated under QSO mode at CFHT include a data reduction pipeline and distribution system to deliver the scientific data to both the PI and the Canadian Astronomy Data Centre (CADC).

4.1.5.1 Data Archive

CFHT has both an online archive of several terabytes as well as a data archived on tape. SPIRou data will be handled within this archive much like the data is handled for other instruments with some minor changes. As we look at making changes it will be important to understand the approximate data volumes expected with SPIRou.

4.1.5.2 Real-time and Final Reduction

This is addressed within a separate preliminary design document.

4.1.5.3 Distribution

Changes to the distribution system will be necessary at both CFHT and CADC in order to accommodate SPIRou data.

4.1.6 Status Monitoring and Notifications

Within CFHT an entire infrastructure has been set up surrounding status information. In particular, status information is used on an ongoing basis to identify the health of each instrument and the overall observatory as well as to assess the environmental conditions during which observations were taken. Both instrumental and environmental status data is included as part of the metadata within the FITS files distributed to the PIs. This information is also available within the science archive. In addition, status information is used to generate alerts whenever conditions deteriorate beyond established notification thresholds. Finally, if fault conditions occur, historical data can be used for root cause analysis.

4.1.6.1 Status Server

The Status Server is used to store medium term status information. Values such as exposure time, current filter, telescope position, and weather data are saved there. FITS headers are generated by the various director agents and written to the Status Server. Temperature and telescope position based focus calculation coefficients are saved in the Status Server. Any status that should be monitored for error conditions, e.g., detector temperature or cryovessel vacuum levels, can have an alarm trigger equation associated with it to cause email and phone text messages on error. Figure 9 shows a screenshot of the "gss" graphical Status Server browser which illustrates the hierarchical nature of the Status Server. A more detailed overview of the Status Server is available at http://software.cfht.hawaii.edu/statserv/Overview. The Status Server C API is available at http://software.cfht.hawaii.edi/statserv/StatusServerAPI.

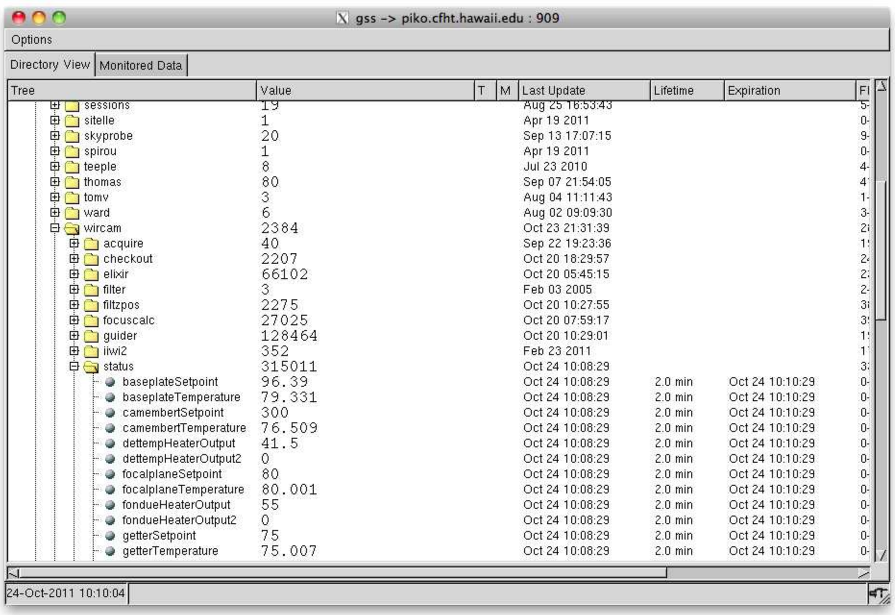

Figure 9: Screenshot of the Status Server Browser (gss)

No software changes to the Status Server are anticipated for SPIRou, although a number of additional status items for SPIRou will be stored within the Status Server.

4.1.6.2 Monitors

A number of monitoring processes are currently available to monitor existing instruments, external conditions, and observatory infrastructure. Some of this monitoring has been added within agents running under director. In other cases, standalone process are used to periodically sample hardware status and update the Status Server.

4.1.6.3 Allen-Bradley PLCs / ABSS / Data Logger

Allen-Bradley PLCs have become the CFHT system of choice for generalized hardware monitoring and control. A standalone process known as "abss" is used to map memory registers in the PLC to hierarchical locations in the Status Server. This mapping is handled via a generalized configuration file. In the future this functionality will be replaced by a newly developed "Data Logger" application.

4.1.6.4 Status Server Archive and Web Plots

Historical information from the Status Server is stored within a status archive. This archive is used to generate on-the-fly plots of key instrument and observatory information. Examples of web pages with both status data and dynamic plots can be found within the left menu links at http://statserv.cfht.hawaii.edu Each instrument has a series of status pages containing both real-time information and dynamically updating plots of historical data. A series of web pages dedicated to SPIRou will also need to be developed.

4.1.6.5 Notifications

Hundreds of business rules have already been defined for existing instruments and observatory infrastructure at CFHT. These rules are designed to compare existing conditions with pre-defined notification thresholds in order to identify potential issues. If conditions deteriorate beyond established thresholds, warning emails, alert text messages, pop-up messages, or pre-defined sound messages are automatically generated. These notifications are used to help ensure that the instruments and infrastructure within CFHT is operating within established specifications.

With SPIRou it will be important to have notification rules defined for many of the temperature and pressure sensor values captured from the instrument. This will help CFHT ensure that the instrument continues to operate within pre-established limits.

4.2 Instrument and Telescope Control Sequence Example

As illustrated in figure 4, commands are sent from the QSO Observing Tool to director in order to control the instrument, move the telescope, and take science exposures. In order to better understand how this works, this section begins with a QSO command sequence and traces what happens as the commands are processed through the instrument control software components. The example is a command sequence from ESPaDOnS.

4.2.1 QSO Command Sequence for ESPaDOnS

The following sequence of commands captures a single intensity exposure of 300 seconds in the "staronly" observing mode at a target location of 10:00:00.00 +20:00:00.0 (RA Dec) using a normal detector readout speed. This example was chosen because this sequence may be similar to command sequences that will be used with SPIRou.

```
@header _runid "11BZ54"
@header _piname "Sherlock Holmes"
@header observer "QSO Team"
@header object "Target Name"
tcoords 10:00:00.00 +20:00:00.0 2000.0 0.0 0.0 "Target Name"
gcoords select
@nheader reset
@nheader 10001.2 QOBSERVE "John Doe" "Name of Q Observer"
@nheader 10001.3 QCOORD "Jane Doe" "Name of Q Coordinator"
@nheader 10001.1 QRUNID "11BQ34" "Q RunID"
@nheader 10001.4 QID 551005000000002225510
@nheader 10001.5 QOBSEQID 550014000000002270205
@nheader 10001.7 QOBITER $$QOBITER(550014000000002270205)
@nheader 10001.9 REL_DATE "2013-02-28T00:00:00" "UTC Release Date"
@nheader 10001.6 QICSEQID 550512000000002293359
go etype=OBJECT rmode=normal obsm=staronly stokes=I enum=1/1 etime=300.0
 raster="FULL" filename=odometer
```

4.2.2 Command Sequence Diagram

Figure 10 illustrates a flow diagram for the messages that occur between the software components shown in figure 4 to support the command sequence example from section 4.2.1.

Before reviewing the sequence diagram there is an important point to be made. Each instrument at CFHT must be designed not only to be robust but also to operate in an efficient manner. Efficiency for an instrument can be defined using the following equation.

$$\text{Efficiency} = \frac{\text{Open Shutter Time}}{\text{Available Observing Time}} \tag{1}$$

For SPIRou to be as efficient as possible, any time that is spent not capturing photons on the science detector is considered overhead. Examples of overhead time are telescope movement, dome movement, instrument configuration, and detector readout. Obviously all these operations are important. However, the key is to minimize them in an effort to improve efficiency and maximize the science output of the instrument. Efficiency must be considered at each step of the design process. From an instrument configuration standpoint this means that multiple actions must be initiated in parallel. Since many groups are planning on using Galil controllers, these controllers must be used in such a way that no single Galil action should prevent another action from being initiated. For example, blocking Galil commands such as "AM" can not be used. Fortunately, the actual communication with the Galil should be handled by a CFHT provided Galil control library that is designed to prevent this. More details about the Galil control library can be found in section 4.3.

The left-hand column of figure 10 contains a numerical identifier at various steps within the sequence. Each of these steps is explained in more detail below.

1. QSO sends individual header commands to director. Director receives these commands and forwards them on to the DetCom agent. These header commands are used to populate several mandatory FITS headers including the PI name, Run ID, observer, and target name.

2. QSO sends a "tcoords" command to director. This command is used to send the requested telescope coordinates to the telescope control system (TCS). Director receives this command and because there is a "tcoords" script in the ˜/.director/bin directory it ends up forwarding this command on to both the controller agent and the TCS agent. The controller agent sends a request to the guider control process to stop any guiding that may be in progress. The TCS agent starts moving the telescope and dome in order to acquire the requested target.

3. QSO sends a "gcoords select" command to director. This command is used to indicate that guiding is requested. Director receives this command and sends it on to the controller agent. The controller agent sends a request to the guide control process to initialize the guider.

4. QSO sends individual nheader commands to director. Director receives these commands and forwards them on to the DetCom agent. These commands are used to set optional QSO specific FITS headers.

5. QSO sends a "go" command containing a sequence of arguments to director. This command is sent to the controller agent by director. The "go" command defines both the required instrument configuration as well as the science exposure parameters. The controller agent initiates instrument configuration requests based on the contents of the "go" command.

6. The controller agent sends configuration requests to the cassegrain agent. The cassegrain agent takes each request, validates the syntax, and initiates any necessary Galil controller operations via Galil library API calls. Each request returns a response to the controller agent almost instantly. This entire process is non-blocking and Galil operations, such as motor movements, are performed in parallel. Please read section 4.3 for more details on how Galil control agents work.

7. The controller agent sends configuration requests to both the spectrograph agent and the exposure meter agent. The spectrograph agent handles Galil operations in the same manner as the cassegrain agent.

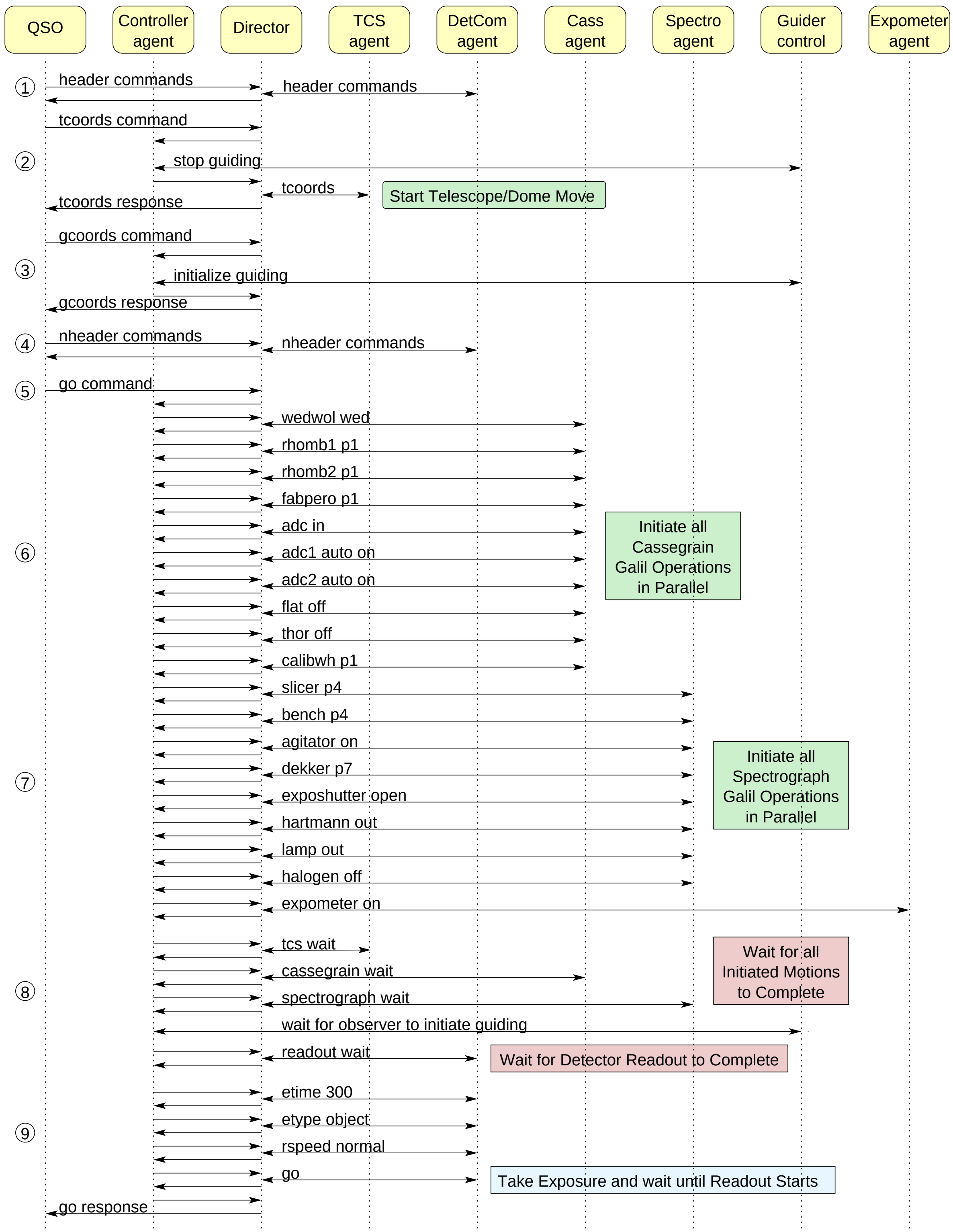

Figure 10: Command Sequence Diagram for ESPaDOnS

8. Before starting the science exposure it is necessary to make sure that all previously initiated movements have

completed. This is accomplished by sending a "wait" command to the TCS, cassegrain, and spectrograph agents. Each "wait" command will block until all previously initiated operations for that agent have completed. If a previously initiated action failed, the wait will return with a failure. If there are no actions initiated, or if the actions have all completed successfully, the wait will return immediately. In addition, in the case of ESPaDOnS, it is necessary to wait until the observer has initiated guiding since this is a manual process. Finally, moving the telescope and configuring the instrument can be done during the previous readout of the science detector. A readout of the ESPaDOnS CCD takes 38 seconds using a normal readout speed. Since the readout of the CCD for ESPaDOnS is long, it is possible to hide much of the overhead within the readout of a previous image. The current design for SPIRou does not include a shutter. As a result, it will not be possible to reconfigure the instrument or move the telescope during a previous readout with SPIRou.

9. Once all the appropriate waits have completed in the previous step, it is time to take an exposure. This is accomplished by sending exposure configuration parameters to DetCom and initiating the exposure. As was mentioned earlier, while the exposure is in progress, it is not possible to send any commands to director with the exception of a "stop" or "abort". Once the exposure is finished and the readout begins, QSO will receive a response to the "go" request. At this point, the next sequence of commands can be processed by director.

Each command request results in a pass-fail response. If a response indicates a failure, the command sequence will immediately stop and the failure must be addressed by the observer. Any failure must be accompanied with detailed diagnostic messages produced by the agents or relevant software components.

4.3 Galil Control

Since several working groups have plans to write software to communicate with a Galil Controller, this is a short section about the CFHT Galil control library.

Galil Controllers have become a hardware standard for instruments used at CFHT. CFHT has developed an extensive set of libraries and code to interface with a Galil controller. We are currently working on implementing a C library that abstracts most of the complexity of working with a Galil. As a result, much of the detail of interacting with a Galil controller is now contained within a config file. Using the new library it should no longer be necessary to send individual commands to the Galil. Instead, the communication with the Galil should be handled via library API calls. In addition, the Galil communication library will handle updates to the Status Server. Figure 11 illustrates at a very high level how agent software will communicate with a Galil controller using the library.

As previously mentioned, a configuration file must be created containing the details of the connected hardware devices. This file contains information about motors, encoders, inputs, outputs, homing directions, limit switches, analog inputs, analog outputs, servo tuning parameters, preset positions, backlash correction, and timeout thresholds to name just a few of the configuration categories. At agent startup, the software performs the following API call to read the configuration file.

```
motorInit()
```

If this call is successful, the agent will wait for commanded input to arrive from director. As mentioned earlier, for efficiency purposes, Galil operations are initiated in parallel. A series of director commands could result in a sequence of API calls, for example, to move several motors (one call per axis).

```
motorMove()
motorMove()
   ..
motorMove()
```

Each routine returns a pass-fail response from the Galil library. The library provides an optional call-back function parameter that is invoked whenever the requested movement has completed.

Once all motor initiation commands have been processed, the agent receives a "wait" command from director. This "wait" command could result in a sequence of blocking API calls.

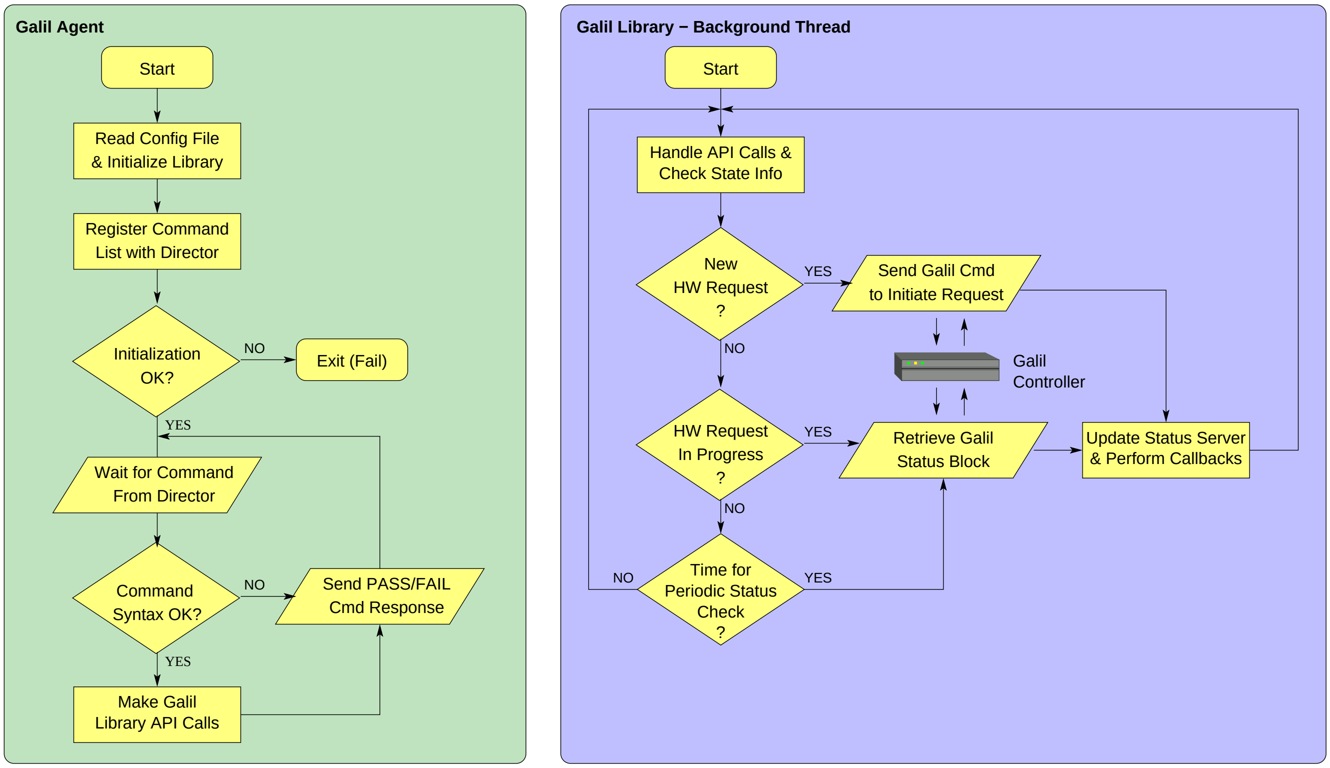

Figure 11: Agent Interface with Galil Library

```
motorAxisWait()
motorAxisWait()
  ..
motorAxisWait()
```

This is a very high-level example, but hopefully it illustrates how an agent would utilize the Galil library.

CFHT discourages the use of custom programs that are stored on the flash memory of the Galil. If this is deemed to be necessary, CFHT would like to be informed prior to implementation. Instead, we prefer that the operation of the Galil is handled via high-level software calls to the Galil library API.

5 Available CFHT Instrument Control Software

The previous section provided details about a CFHT library for Galil communication. CFHT has written software to interface with a number of different hardware subsystems. Wherever possible we would like to use hardware that we have experience with and available software for. There are two important reasons why. First, the software interfaces have already been written and debugged. Second, we can utilize a pool of available spares across multiple instruments. Below is a list of hardware that meets this criteria. For any of these systems CFHT will provide source code and libraries that can be used and, if necessary, modified.

- Galil Controller
- Allen-Bradley SLC 500 PLC
- LakeShore Temperature Controller

- Pfeiffer Vacuum Gauge
- Baytech Remote Power Control
- Perle RS-232/RS-485 to Ethernet
- Axis video cameras and video servers

More details about hardware and software standards can also be found in the Instrument Design Specification (IDS).

6 Reference Documents

This section contains links to other CFHT reference documents that are not found in the SPIRou DocuShare system. SPIRou specific information can be found in the SPIRou DocuShare repository at http://ged.obs-mip.fr/omp/dsweb/HomePage.

- **Instrument Design Specification (IDS)** - http://software.cfht.hawaii.edu/InstrumentDesignSpecs.pdf
- **Director** - http://software.cfht.hawaii.edu/director
- **Status Server** - http://software.cfht.hawaii.edu/statserv
- **DetCom** - http://software.cfht.hawaii.edu/detcom
- **Client-Server Socket Library** - http://software.cfht.hawaii.edu/sockio
- **Data Acquisition** - http://software.cfht.hawaii.edu/DataAcquisition
- **SPIRou Tip/Tilt Guiding and Viewing Software Responsibilities** - http://software.cfht.hawaii.edu/spirou/GuidingSoftwareResponsibilities
- **SPIRou High-Level Instrument Control Software Overview and Examples** - http://software.cfht.hawaii.edu/spirou/SpirouSoftwareOverview
- **Motor Control Library (preliminary draft)** - http://software.cfht.hawaii.edu/MotorControlLibrary.pdf

7 Acronyms

Acronym	Description
ABSS	Allen-Bradley PLC to Status Server software process to map data from the PLC to the Status Server
API	Application Programming Interface
ASIC	Application-specific Integrated Circuit
CADC	Canadian Astronomy Data Centre
CCD	Charge Coupled Device
Dec	Declination
DB	Database
DetCom	Detector Communication Agent
ESPaDOnS	Echelle SpectroPolarimetric Device for the Observation of Stars at CFHT
FITS	Flexible Image Transport System
GSS	Graphical Browser for the Status Server
GUI	Graphical User Interface
H2RG	Teledyne Hawaii-2RG Infrared Detector
H4RG	Teledyne Hawaii-4RG Infrared Detector
IDS	Instrument Design Specification
ISU	Image Stabilization Unit
NEO	New Environment for Observing
OAP	Observatory Automation Project
QSO	Queued Service Observations
PI	Principal Investigator
PLC	Programmable Logic Controller
PDR	Preliminary Design Review
RA	Right Ascension
SAM	Teledyne Sidecar ASIC Module
SPIRou	SpectroPolarimetre Infra-Rouge (Near Infrared Spectropolarimeter)
TAC	Time Allocation Committee
TBD	To Be Determined
TCL	Tool Command Language
TCS	Telescope Control System
WIRCam	Wide-Field Infrared Camera At CFHT