ACS (ALMA Common Software) operating a set of robotic telescopes

C. Westhues[a], M. Ramolla[a], R. Lemke[a], M Haas[a], H. Drass[a] and R. Chini[a,b]

[a]Astronomisches Institut, Ruhr-Universität-Bochum, Universitätsstr. 150, 44801 Bochum, Germany

[b]Instituto de Astronomía, Universidad Católica del Norte, Avenida Angamos 0610, Antofagasta, Chile

ABSTRACT

We use the ALMA Common Software (ACS) to establish a unified middleware for robotic observations with the 40cm Optical, 80cm Infrared and 1.5m Hexapod telescopes located at OCA (Observatorio Cerro Armazones) and the ESO 1-m located at La Silla. ACS permits to hide from the observer the technical specifications, like mount-type or camera-model. Furthermore ACS provides a uniform interface to the different telescopes, allowing us to run the same planning program for each telescope. Observations are carried out for long-term monitoring campaigns to study the variability of stars and AGN. We present here the specific implementation to the different telescopes.

Keywords: Alma Common Software – ACS, observatory control, telescope control, Hexapod

1. INTRODUCTION

The robotic operation of telescopes demands a reliable software infrastructure, which ensures first and foremost a safe telescope control. We run 4 different telescopes on two sites in Chile, therefore the applied software has to fulfill a special requirement profile. It should be able to deal with various mount and camera types. Representing the distributed devices through a unified interface to the observer is desirable for observational comfort and minimization of training and efforts. A real-time software system is needed to guarantee a reliable robotic operation. Also the integration of available code should be possible. The separation of observational and engineering tasks should be achieved.

The public ACS (version 3.1 – 8.1) was chosen as middleware. Since 2004 the mount and camera controls of the Hexapod telescope (HPT) were wrapped in ACS-containers. As the observatory grew with the installation of two other telescopes (40cm Bochum-Monitoring-Telescope (BMT) and 80cm Infra-Red-Imaging-System (IRIS)) the ACS control of HPT could be generalized and applied to these two telescopes as well, although they differ in mount type (Hexapod, equatorial and Alt-Az). Also the control of the cameras (for optical, infrared and Lucky-Imaging) could be embedded in ACS containers and are controlled by a common GUI to all telescopes. Finally using the ACS framework we successfully implemented robotic features to the ESO-1m telescope (LSOM) at La Silla in 2012, a telescope originally build in 1966 for manual observation only.

The benefits from the re-usage of the ACS containers are numerous. The reapplication saves time, money and efforts in the software development. Underlying technical specifications of the different telescope can be hidden and the observer can operate on a common generalized telescope. Unlike ALMA, which uses ACS on similar telescopes in a common way, we apply it quite differently, in a range of ALMA-like real-time control over CAN-bus at HPT to a dummy ACS-MOUNT-component at LSOM.

For small observatories, cost effectiveness of operation is important. With the ACS approach we succeeded in performing long-monitoring campaigns of eclipsing and spectroscopic binaries, star-forming regions, the galactic disk, AGN and several other projects in the last years.

E-mail: christian.westhues@rub.de

1.1 ACS – ALMA Common Software

ACS, originally developed for the Atacama Large Millimeter Array (ALMA), does not aim to be a complete telescope or observatory control software. It is intended as middleware, providing an accessible interface, where diverse observatory equipment can be integrated. However, due to its flexible design concept, it is used in several other observatories. ACS is based upon open source software and incorporates state of the art technologies. The Distributed Objects are handled via CORBA.[1]

ACS can manage the time critical data flow in real-time. The middleware can implement code of C++, python and Java. The generalization of the hardware components works with the inherent component-container model. The control software of every hardware component is embedded in a container with the same interface defined in the Interface Definition Language (IDL). The common control clients access the components via their ACS-interface. Also components can easily be shifted, e.g. reassigning instruments between different telescopes.

1.2 Telescopes

In figure 1 the different telescopes are shown. The specifications of the telescopes and imaging systems are as follows:

1.2.1 HPT – Hexapod-Telescope

The HPT can use its six legs to move a 1.5m main adaptive mirror in all 6 degrees of freedom. Adapting the complex control software in an ACS control frame, the pointing and guiding accuracy needed for astronomic observations was reached.[2–4] It is connected via a fiber to the BESO-spectrograph.[5] Spectroscopic monitoring of high-mass stars is performed to study their multiplicity.

1.2.2 BMT – Bochum-Monitoring-Telescope

The BMT[6,7] is a 40cm equatorial mounted telescope. It is equipped with a SBIG STL-6303 camera. Using B and V broad-band and three narrow-band filters the telescope monitors AGN and catches the time delay in the Broad-Line reverberation photometrically. All components (mount, CD, roof, weather station, etc) are completely encapsulated in ACS. Therefore the BMT is able to operate robotically by a control daemon.

1.2.3 IRIS – Infra-Red-Imaging-System

IRIS[8] is an 80cm alt-azimuth mounted telescope. Through a tertiary mirror the beam can be directed to two Nasmyth-foci. On one side the infrared-camera is attached, where J,H and K filtered images can be taken. On the other side the BESO-spectrograph fiber can be attached. Variability studies of star-forming regions, dust-reverberation-mapping of AGN and spectroscopy are carried out autonomously, using tabulated lists as for the BMT in the ACS environment.

1.2.4 LSOM – La-Silla-One-Meter

The latest member of our ACS operated telescopes is the 1m equatorial mounted telescope on La Silla which was out of operation since recording the DENIS catalog. We were able to embed some of the old hardware and a high-speed camera for lucky imaging in the generalized framework of ACS. The mount controls bypass the ACS but keep the common mount component informed. Planning routines could therefore be reused from prior projects and we are now able to observe without interaction. We successfully use an Andor-Neo camera to perform lucky imaging of bright stars. In cooperation with other Chilean partners it is furthermore planned to put other instruments at site which will make use of our ACS mount control.

Figure 1. BMT, HPT, IRIS (Cerro Armazones)and LSOM telescope (La Silla)

2. A MIDDLEWARE FOR ROBOTIC TELESCOPE CONTROL

The general purpose of the middleware is to administer and control the diverse and distributed physical devices as seen in Figure 2. Furthermore the physical properties and functions are reorganized in logical devices. Applications can then work on the logical units without knowing the real hardware on the physical side. The multiple physical devices can be managed quite differently, discussed in more detail in Sections 2.1 - 2.3.

To build a generalized telescope control, first the common components of a telescope have to be defined (Figure 2). The ACS lends itself to administer these components as Distributed Objects (DOs). The unique features of a telescope are then reflected in exclusive DOs, e.g. the active optics of the HPT. We will focus here on the common DOs and their diverse technical implementation.

Common DOs to all the telescopes are GEO (geographical position), METEO (weather information), MOUNT, CAMERA and DOME. In the framework of ACS every component can have multiple properties, e.g. the properties of GEO are longitude, latitude, altitude and time. The components and their properties have to be defined in the Configuration Database (CDB) which is used to initialize the DOs at run-time. A component runs in a container with an interface defined in the IDL. The IDL interface can be used to publish certain properties or functions. E.g. the MOUNT-container publishes the property `mountMode` (Shutdown, Standby, Track or PolTrack) and the function `sourceEquatorial`, which will point the telescope to some equatorial coordinates.

Within the container the connection to the real component is established, which is a task for the engineering division. We use C++ (plus Real-Time-Application-Interface (RTAI) at HPT) in containers, which are directly

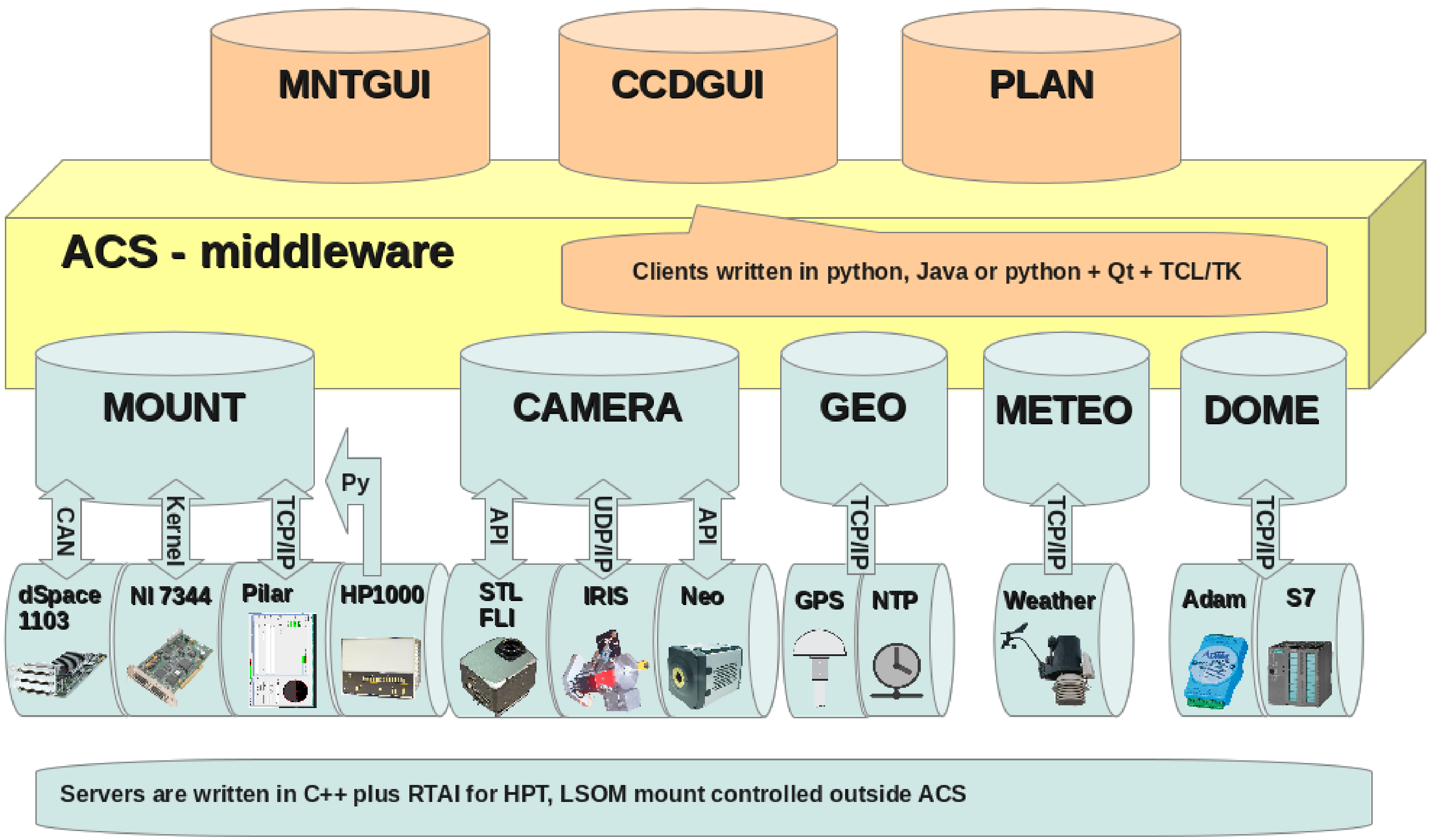

Figure 2. Unified-middleware-schema: ACS moderates between physical devices embedded in containers MOUNT, CAMERA, GEO, METEO and DOME (blue side) and the observer using MNTGUI, CCDGUI and PLAN (orange side)

related to physical devices. For example the `sourceEquatorial` function for the LSOM telescope is realized by a handler, which feeds the HP1000 over a serial connection with presets and commands. The METEO-component is connected to the weather station of the observatory and can be used by all telescopes at site simultaneously. With this concept the telescope control can be decoupled from the real components towards the common DOs, which then form a new abstract and unified telescope.

Clients operating just on top of the middleware use C++, python and Java which are extended by Qt- or Tcl/Tk-libraries for the GUIs. They are also embedded in the ACS framework. Because they do not depend on a certain telescope they can be called independent DOs. Two GUIs (CCDGUI and MNTGUI) control mount and camera manually at site or from remote. A daemon (PLAN) performs autonomous observations.

2.1 GEO, METEO and DOME

The GEO and METEO components are the most simple, because they only provide some properties. For GEO the geographical position is gathered from a GPS interface at the HPT. For IRIS the time is synchronized via NTP.

For the METEO-component at Cerro Armazones values are taken from the local weather station, while at La Silla, the ESO web page is used to monitor in particular dew and high wind, the natural enemies of astronomy.

For BMT and LSOM the weather information is used to feed some watchdogs, to keep the roof or dome closed at dangerous values. Technically this is realized with Adam-Box and Siemens S7 controls. While the Adam-Box just provides relay functions to the roof motors of BMT, the watchdog routines are implemented in the planning daemon of BMT. The daemon also has to check if the telescope is parked, before the flat-roof can be closed. For LSOM the watchdog is directly running on the Siemens S7, which has to be fed continuously with good weather information, otherwise the dome will be closed. The rotation of LSOM dome is controlled with the Adam-Box.

2.2 MOUNT

The common properties and functions of MOUNT are numerous. The primary mount controls by the manufactures are encapsulated here. This was achieved differently on each telescope, either by accessing predeveloped telescope-servers over their network interface or by handling serial connections to the physical mount devices in real-time. Functions and properties can comfortably be utilized by the MNTGUI described in Section 2.4.

At HPT the ACS operates the primary *dSpace* controller card over a CAN-Bus as described in Figure 3. The primary control runs a real-time system, which will execute its main loop every 5ms. Due to the special requirements of the Hexapod mount, the original control software failed in a sufficient guiding accuracy. The required tracking stability could only be achieved by a constant input of coordinates to the telescope via the ACS middleware. Therefore the MOUNT component is divided in a real-time part, sending every 100ms new azimuth and elevation coordinates over the CAN-bus. A non-real-time part is used to calculate the positions in advance. Therefore the HPT mount control essentially resembles the original ACS use at the ALMA site. Using the feedback of a guiding camera ensures a sufficient tracking accuracy

For BMT the primary controls of $NationalInstruments$ were implemented in a telescope-server (written in C++), which can be accessed via a client using the ACE network interface library (Figure 4). The server runs the telescope with a modified ni73xx motion driver for the NI-7344 controller card. The axis motion in Hour-Angle (HA) and Declination (Dec) are controlled in a closed loop for the relative encoders and steppers. The absolute encoders (HA, Dec) are read out via a NI-6509 card in Gray-Code with the comedi kernel module. The middleware incorporates the user-space controls of the predeveloped server-application and runs the MOUNT-container on a separate machine (connection to kernel-space), as ACS can naturally deal with a distributed system.

Figure 5 shows how IRIS telescope uses the Pilar Software, where MOUNT monitors the dataflow and gives input over the Open-Telescope-Control-Interface (OpenTCI). This is realized by TCP/IP-connection to the telescope-server, where presets, offset, foci-positions and Nasmyth-foci can be send, and the execution by the mount is verified. Also autoguiding for spectroscopy is realized by the implementation of feedback from a fiber-viewer camera. A python daemon monitors the accurate position of the star on the fiber and will trigger offsets directly to the Pilar, to prevent the loss of tracking.

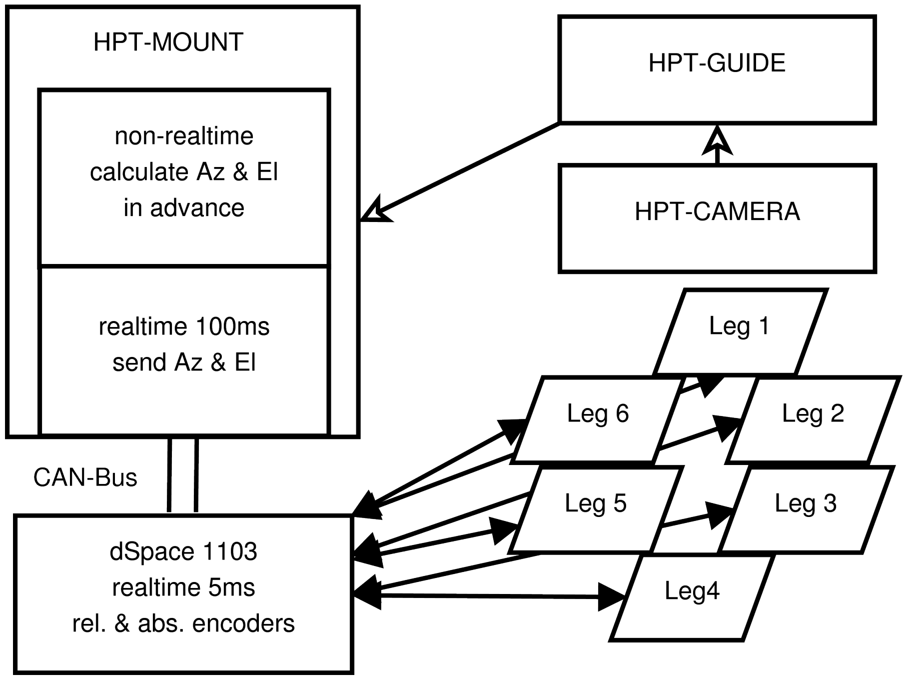

Figure 3. HPT-MOUNT-control-schema: the primary mount control is fed constantly by ACS with coordinates over a CAN bus, feedback over a guiding camera ensures tracking stability

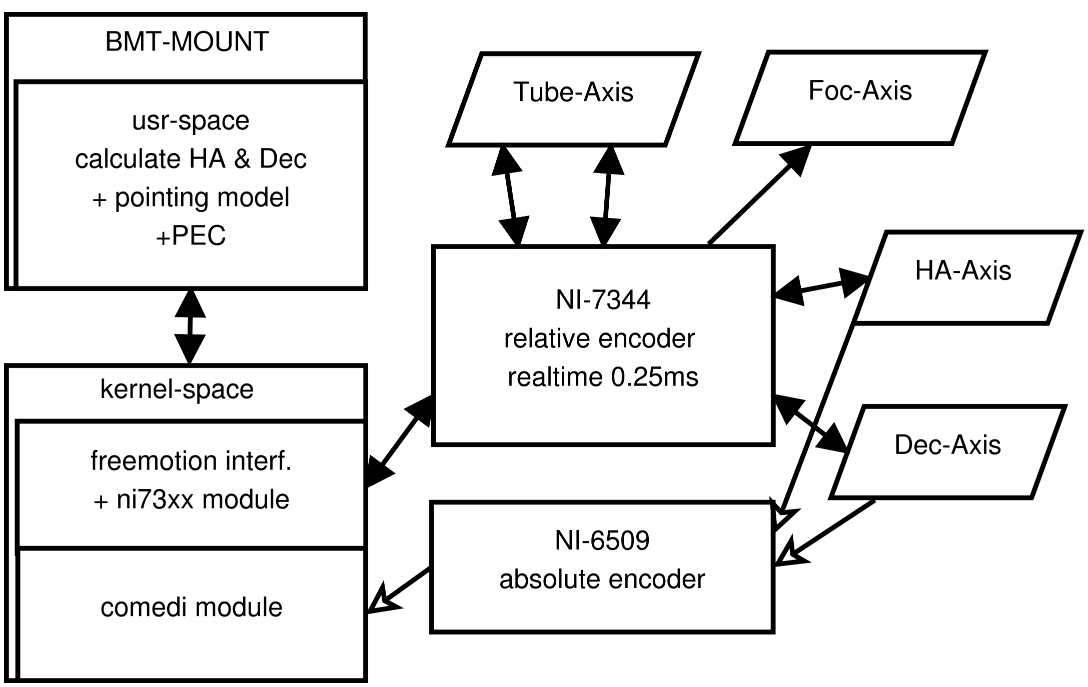

Figure 4. BMT-MOUNT-control-schema: the ACS component is running on a seperate machine and operates the NI-cards over their kernel-modules, tracking accuracy reached with encoder feedback for pointing model and Periodic-Error-Correction (PEC)

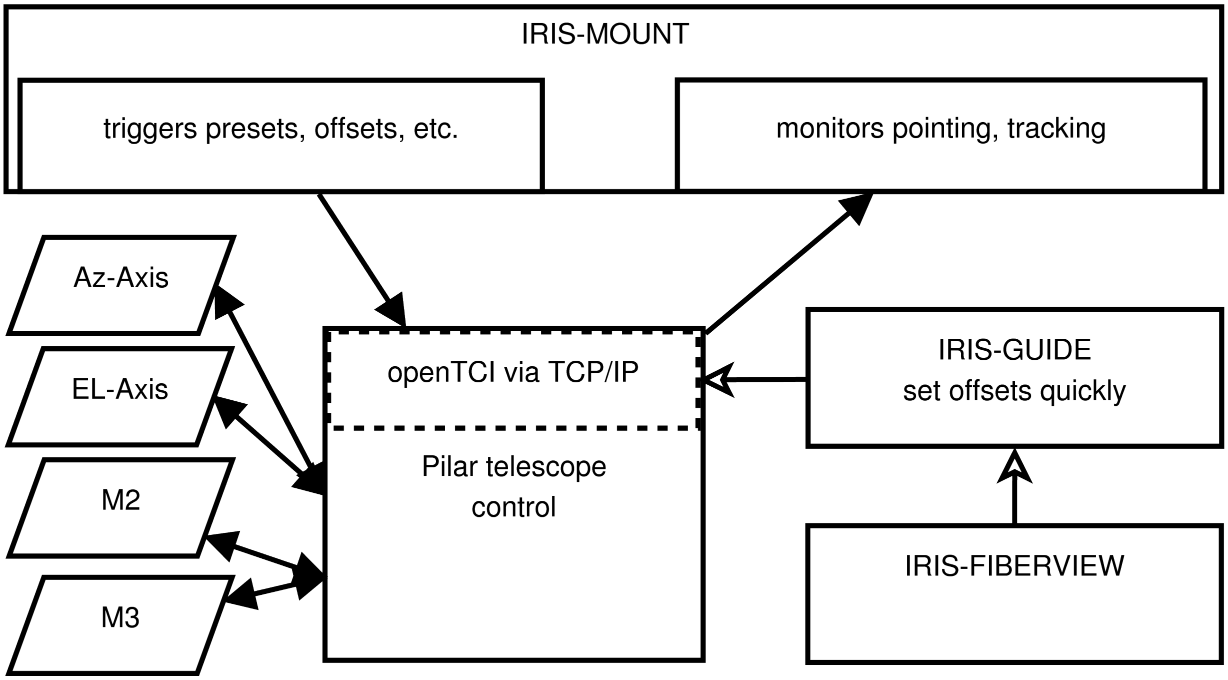

Figure 5. IRIS-MOUNT-control-schema: ACS controls and monitors the telescope over a TCP/IP connection to the openTCI interface of the primary control software, guiding offsets are bypassing the MOUNT-component due to speed

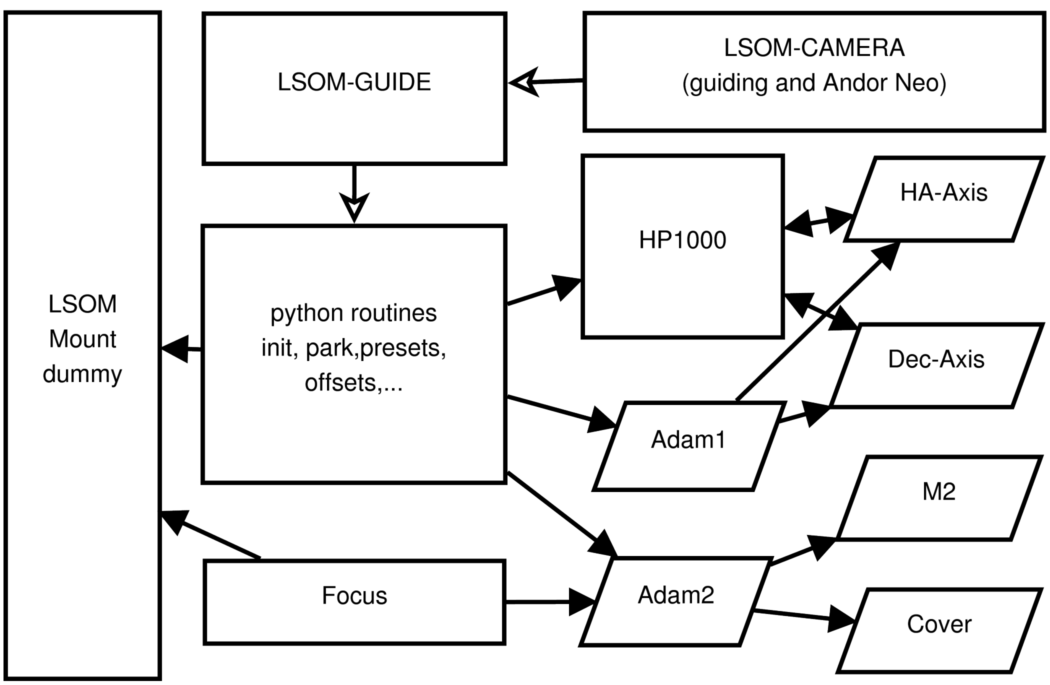

Figure 6. LSOM-MOUNT-control-schema: LSOM is controlled by a set of python routines outside the middleware which keep the dummy ACS-MOUNT just informed

At LSOM the primary telescope control (Figure 6) is run by an HP1000 which still operates a menu driven control over two terminals. To get access to the ancient control system, we emulate two terminals with their RS-232-connection in the HP1000-protocol. Presets can be set by a handler to the HP1000, where hexadecimal ESC-code sequences (to navigate through the old menus) are stored. The original handset controls were soldered to some Adam-Boxes (relay connection over TCP/IP), so that python routines can perform basic offset commands. The ACS-MOUNT-component is here used only as a dummy. The diverse control routines, to beam the old LSOM hardware to the internet-age, just keep the MOUNT informed, as the basic functions to move the real mount cannot be triggered within the ACS-middleware.

2.3 CAMERA

The various CCD cameras at USW are generalized by the CAMERA component. The properties and functions can be handled by the CCDGUI described in Section 2.5.

The HPT is operated with the BESO-spectrograph. It's camera is not yet included in the ACS middleware.

The BMT operates a SBIG-STL-6303 with 3072×2048 pixel. The CAMERA component implements the SBIG libraries and also operates the internal filterwheel with it's five filters.

The IRIS camera is operated by a code for the inbuilt DSP from the Institute for Astronomy (IfA) in Hawaii. The middleware triggers the shutterless camera and the filter-wheel over networking connections to a camera-server via UDP/IP. Due to the special needs of infrared imaging, a set of python routines were defined. These routines are used to perform image sequences on source- and sky-fields, which were optimized for the reduction pipeline.

The LSOM is running a Andor-Neo camera with 5.5 Megapixel at 12bit. To perform lucky imaging with readout rates of 100Hz, CAMERA includes the supplied libraries from Andor and was extended by the numerous unique properties of the Neo camera to develop the best readout mode for lucky imaging.

2.4 MNTGUI

The graphical interface of the MNTGUI component is shown in Figure 7. The GUI is written completely in python and imports all the properties and functions from the MOUNT component. All relevant information is displayed for equatorial and horizontal coordinate system. With the inbuilt functions the mount is parked, pointed to objects, which can be chosen from a sourcelist or turned to tracking or guiding mode. The GUI can be run locally or remotely. As this GUI is independent of the actual telescope as described in Section 2 the observer can control each mount in a homogeneous way.

2.5 CCDGUI

In the CCDGUI component (Figure 8), the attached camera can be chosen from CAMERA component and their properties are displayed or can be changed to the desired values. The GUI allows to start the implemented functions like `startExposure`. Also a DS9 image viewer is started, where the latest image will be displayed. The CCDGUI also collects the property information of MOUNT and writes all scientifically relevant information in the image header.

2.6 PLAN

The planning daemon of BMT, shown in Figure 9, uses the features of MOUNT, CAMERA, METEO, GEO and DOME to perform autonomous observations during the night according to a given list. It is written in python and the telescope status can be monitored in a GUI. Actual pointing coordinates are shown, filter wheel position and weather information are gathered from the middleware and the observer can interact in case of emergency. The nightly observing plans are optimized to make effective use of the observing time. Also plans can easily be shifted or synchronized between the telescopes in multiwavelength campaigns.

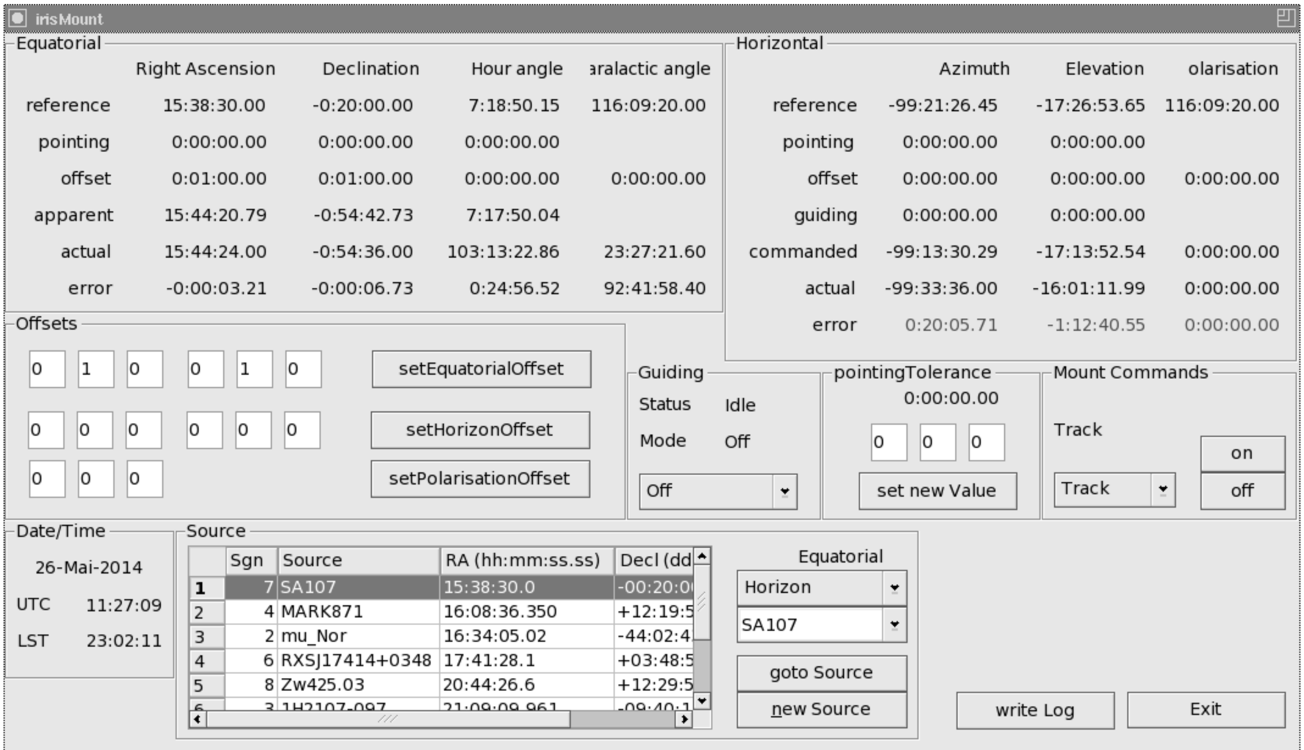

Figure 7. MNTGUI: equatorial and horizontal coordinates can be monitored in the upper part of the GUI, source presets can be selected from a list and set by the “goto Source” button, offsets can be given in both coordinate systems, also “Mount Commands” (middle right) like “Shutdown” can be triggered

3. SUMMARY

Using ACS we economically run 4 different telescopes on two sites in Chile. The ACS middleware is well suited for software development and provides a homogeneous framework for an observatory with changing staff. The observatory benefits from a stable development environment, where the re-usage of software facilitates the implementation of any new or old telescope. A homogeneous surface towards the observer is achieved, while also allowing unique features of individual telescopes to be embedded. The ’Generalized Telescope’ approach is a cornerstone for the successful long-term monitoring campaigns for multiple scientific question.

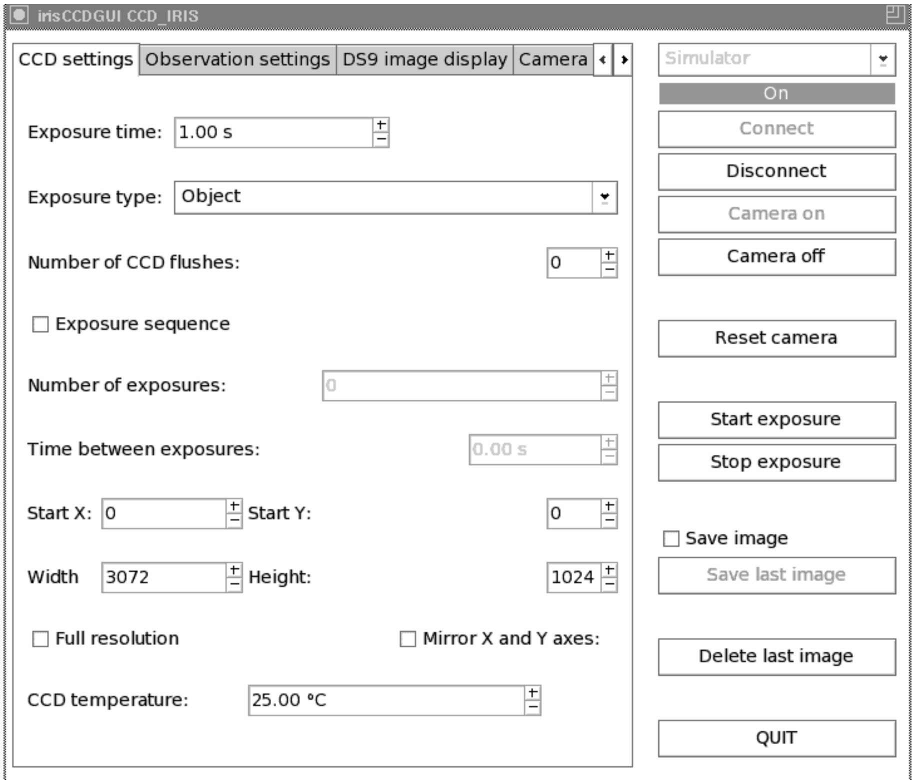

Figure 8. CCDGUI: different settings tabs can be monitored and changed, like exposure time and pixel number, unique features of the cameras are implemented in special tabs, exposures can be started and stopped with the buttons at the right

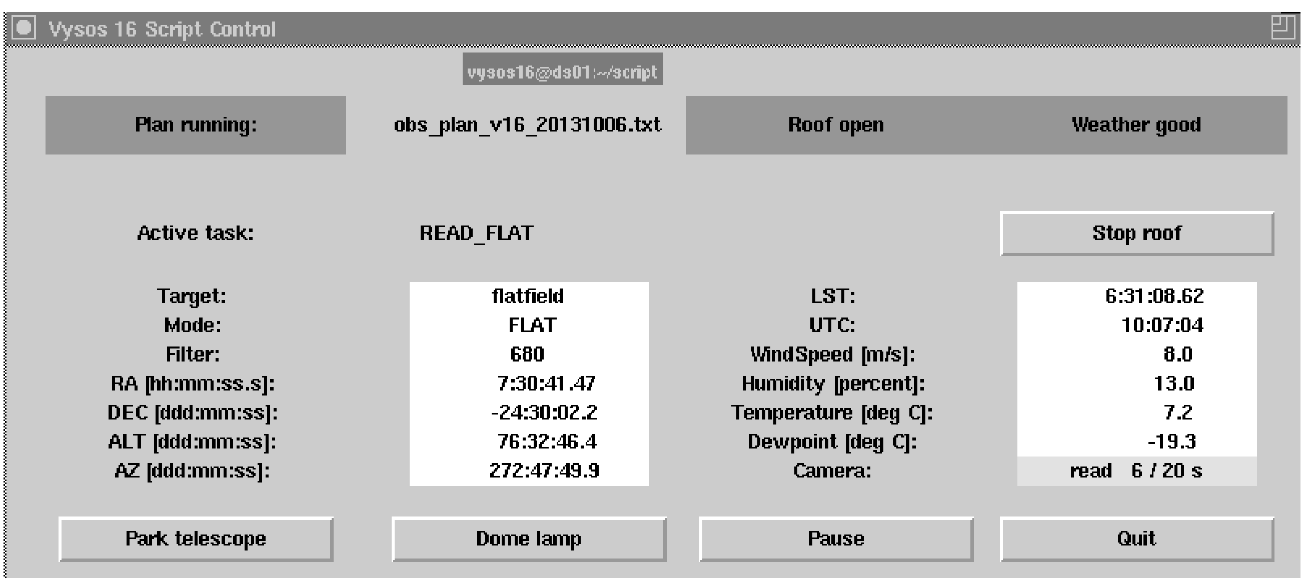

Figure 9. PLAN-daemon for BMT (formally named VYSOS 16), targets, presets, filters and exposure times are taken from a list, the execution can be controlled here, the daemon will park the telescope in case of bad weather

REFERENCES

[1] Chiozzi, G., Gustafsson, B., Jeram, B., Plesko, M., Sekoranja, M., Tkacik, G., and Zagar, K., "CORBA-based Common Software for the ALMA project," in [*Advanced Telescope and Instrumentation Control Software II*], Lewis, H., ed., *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* **4848**, 43–54 (Dec. 2002).

[2] Paegert, M., *Pointing and active optics of the hexapod telescope*, phd thesis, Ruhr-Universität-Bochum (2007).

[3] Jürges, T., *Entwurf und Realisierung einer Nachführungskorrektur für Teleskope mit mehr als zwei Bewegungsfreiheitsgraden*, diploma thesis, Universität Duisburg-Essen, Germany (2004).

[4] Drass, H., *Das Pointing des Hexapod Teleskops Analyse und Implementierung*, diploma thesis, Universität Dortmund, Germany (2009).

[5] Steiner, I., Seifert, W., Stahl, O., Lemke, R., Chini, R., and Appenzeller, I., "BESO: a high-resolution spectrograph for the Hexapod-Telescope," in [*Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*], *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* **6269** (July 2006).

[6] Ramolla, M., Drass, H., Lemke, R., Westhues, C., Pozo Nuñez, F., Barr Dominguez, A., Haas, M., Chini, R., and Murphy, M., "The 40 cm Monitoring Telescope of the Universitätssternwarte Bochum," *Astronomische Nachrichten* **334**, 1115 (Dec. 2013).

[7] Ramolla, M., *The central region of active galactic nuclei*, phd thesis, Ruhr-Universität-Bochum (2012).

[8] Hodapp, K. W., Chini, R., Reipurth, B., Murphy, M., Lemke, R., Watermann, R., Jacobson, S., Bischoff, K., Chonis, T., Dement, D., Terrien, R., Bott, K., and Provence, S., "Commissioning of the infrared imaging survey (IRIS) system," in [*Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*], **7735** (July 2010).