Implementation of the Software System for the SkyMapper Automated Survey Telescope

A. Vaccarella*, T. Preston, A. Czezowski, S. C. Keller, B. P. Schmidt, P. J. Young

Research School of Astronomy and Astrophysics
Australian National University
Mount Stromlo Observatory
Cotter Rd, Weston
ACT 2611
Australia

ABSTRACT

This paper describes the software systems implemented for the wide-field, automated survey telescope, SkyMapper. The telescope is expected to operate completely unmanned and in an environment where failures will remain unattended for several days. Failure analysis was undertaken and the control system extended to cope with subsystem failures, protecting vulnerable detectors and electronics from damage. The data acquisition and control software acquires and stores 512 MB of image data every twenty seconds. As a consequence of the short duty cycle, the preparation of the hardware subsystems for the successive images is undertaken in parallel with the imager readout. A science data pipeline will catalogue objects in the images to produce the Southern Sky Survey.

Keywords: Automated survey telescope, automated data pipeline

1. INTRODUCTION

The SkyMapper Survey Telescope is designed as a replacement for the 50 inch Great Melbourne Telescope destroyed in the bushfires at the Research School of Astronomy and Astrophysics (RSAA), Mt Stromlo Observatory, Australia, in 2003. To be located at RSAA's Siding Spring Observatory, NSW, Australia, this 1.3m telescope's primary scientific aim is to conduct the Southern Sky Survey: a multi-colour photometric survey of the southern sky[1]. The survey will create a resource analogous to that of the Sloan Digital Sky Survey[2] in the northern hemisphere, but with greater areal and temporal coverage. The telescope features a filter set optimized for stellar astrophysics, namely the determination of stellar temperatures, surface gravities and abundance. The telescope will be accessible to the wider astronomical community for research that is beyond the scope of the Southern Sky Survey via a peer-reviewed time allocation process.

Components of the DeMarco method were employed to define the software requirements for the SkyMapper system early in the project, in particular Data Flow Diagrams (DFD) and Context Diagrams. During the course of the project the requirements were amended to accommodate hardware changes as necessary and the models updated to reflect the changes. Failure analysis was also undertaken for critical subsystems and the result of the analysis fed back into the software requirements.

Reuse of in house software was necessary to reduce development time. To enable automatic operation the SkyMapper telescope will be controlled by the RSAA Telescope Automation and Remote Observing System (TAROS)[3] software that is currently nearing completion. TAROS will provide the SkyMapper observatory with autonomous control. Observation requests will be sent to TAROS via an external real time scheduler in the form of an RTML[a] Observation Block. TAROS provides the external scheduler with feedback regarding the validity and status of the observation request.

[a] http://www.uni-sw.gwdg.de/~hessman/RTML/RTML-3.0g/

Advanced Software and Control for Astronomy II, edited by Alan Bridger, Nicole M. Radziwill
Proc. of SPIE Vol. 7019, 70192R, (2008) · 0277-786X/08/$18 · doi: 10.1117/12.788450

The instrument control component of the larger TAROS system is performed by the RSAA Configurable Instrument Control and Data Acquisition (CICADA)[4] software package. CICADA is a distributed multi-process multi-threaded application using a plug-in architecture for the control of specific hardware. The integration of new hardware is greatly simplified by this architecture, as the developer of the new plug-in need only be concerned with the operation of that specific component. The complex relationships between the various subsystems and archiving of data is handled by the CICADA core, the plug-in need only provide status updates and receive commands in the appropriate format to and from the CICADA core.

The ability of the TAROS/CICADA system to operate multiple subsystems in parallel is critical for the SkyMapper system. The Southern Sky Survey, for which SkyMapper is being constructed, has a very short duty cycle and high repetition with a minimum exposure time of 5 seconds and a maximum of 50 seconds. The survey is relying on the ability of the system to prepare all the subsystems for the proceeding exposure during the detector readout and archiving. The time limit set for the detector readout and archiving is 15 seconds.

Failure Mode Effect Analysis (FMEA) of the system highlighted the complex relationships between seemingly harmless subsystem failures and the potential for costly damage to sensitive equipment. A system was required which could be easily tailored to monitor the complex relationships between several subsystems and take action to protect sensitive equipment or alert personnel where no software remedial action is possible.

2. SOFTWARE REQUIREMENTS

2.1 Software Requirements Specification

SkyMapper will run autonomously, receiving observation requests from a real time scheduler. There will not normally be any personnel in the dome during nighttime observations, though daytime site staff will attend to repairs and routine maintenance as required.

The SkyMapper wide field camera is a CCD mosaic made up of 32, 2k x 4k CCDs. These CCDs will be controlled by StarGrasp detector controllers, which are a customised version of the StarGrasp controllers as used in the PanStarrs[5] Gigapixel Camera, that will be interfaced via Ethernet to a computer running the Sun Solaris operating system.

The SkyMapper focal plane is required to be readout in 15s, which is 3.5 times faster than the time taken to readout RSAA's WFI[b] instrument. With this faster readout speed and the greater number of pixels, SkyMapper places a 15-fold increase in processing demand on the computer system compared to WFI.

The control software system will be required to transfer the raw images from the Siding Springs Observatory (SSO) to the Australian National University Supercomputer Facility (ANUSF). The transfer will take place over a gigabit fibre connection. The data will be trickled down to ANUSF during periods of low traffic over this connection.

An external scheduler developed by the SkyMapper science team will send observation commands to TAROS. The external scheduler will send TAROS an Observation Block, and TAROS will be required to send back an acknowledgement for each observation received and the status of the observation on completion.

The control software will be required to accept blocks of data containing observation parameters one at a time. The blocks of data may contain parameters for the instrument and telescope settings and also accept a single command at a time. The scheduler may send a block of data containing instrument and telescope settings that can be set concurrently. TAROS/CICADA will then be required to send the appropriate commands to the appropriate hardware concurrently. TAROS/CICADA will have the intelligence to determine what hardware can be set concurrently. An example is where the data block may contain a specified filter, rotator position, right ascension and declination. The filter can be changed at the same time that the telescope slews to the specified position and the instrument rotator is rotated to the specified position. TAROS/CICADA will send an acknowledgement to the external scheduler on receipt of the command, then command the specified hardware, and then send a status message on completion back to the scheduler.

Electro Optic Systems Pty Ltd (EOS) will construct the telescope and dome and they have provided a Windows C++ application programming interface (API) called the Control System Client API. The API will be used to interface the RSAA software control system with the EOS telescope control system (EOSTCS). The EOSTCS will provide control of

[b] http://www.mso.anu.edu.au/observing/40in/wfi/

the telescope and dome, and access to the meteorological data from the on-site Met Station. The EOSTCS will have a direct link to the Met Station, and no other links to the Met Station are permitted.

The scheduler and TAROS will require weather data for scheduling and for FITS header data. The Met Station will be connected directly to the EOS Dome Controller. No other connections with the Met Station will be available, and the meteorological data will have to be retrieved from the EOSTCS through the EOS supplied Control System Client API. The TAROS design has a Met Client which links to a Met Server via a socket to gather the necessary weather data. The control software system will be required to implement a TAROS compatible Met Server to provide the meteorological data to TAROS, which will then publish the data to the SSO LAN.

The Shack-Hartmann camera and analysis software will be controlled by third party software and initially be operated manually. This will be integrated into the rest of the autonomous system at a later stage.

The system will be required to respond to an alarm from either the thunderstorm sensor (TSS) or electromagnetic field mill (EFM) by closing the dome and parking the telescope. The communications link between both the TSS and EFM will be RS485.

The software system will be required to provide aperture tracking. Aperture tracking is where the dome shutters are positioned such that they allow an uninterrupted view from the telescope through the dome while blocking any wind that may affect the movement and stability of the telescope.

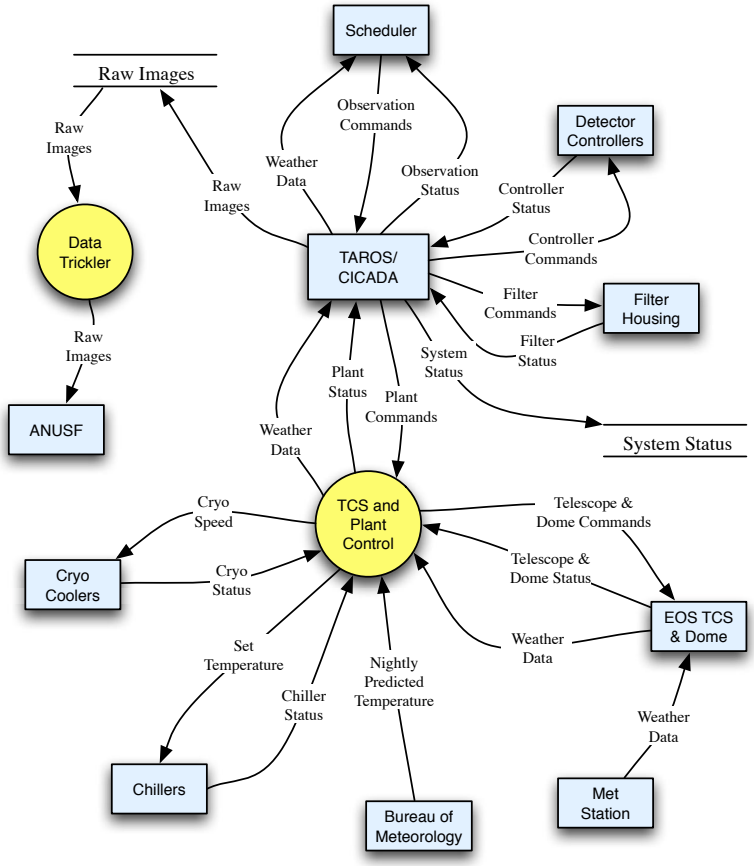

Fig. 1: Level 1 Data Flow Diagram of the SkyMapper data Acquisition and Control System

The diagram in Fig. 1 shows the data flow required between the RSAA control software and external entities of the system. There is a further system not shown in the diagram, which is the control system for the Shack-Hartmann camera. This system will be controlled by commercial off the shelf (COTS) software and will not be integrated with the rest of the control software system.

Each external entity is shown by a rectangle. The external entity can be in the form of hardware such as the cameras, external software such as the scheduler or an external web page such as BoM. A persistent data store, such as a database, is shown by the data store name bounded by two horizontal lines, for example the Raw Images store. The arrows flowing into and out of the software system represent the direction and path of the data moving through the system.

The EOSTCS, dome, and access to the Met Station are required to be controlled through Microsoft Windows applications. The third party APIs and drivers supplied for these three components are all TCS and plant.

CICADA requires a direct communication link with the telescope and dome. TAROS requires weather data from the Met Station to the TAROS Met Client.

The Met Server is required to retrieve the weather data from the EOSTCS and then supply that data to the TAROS Met client and also the Dome Thermalisation process. The EOSTCS Client receives commands from TAROS/CICADA which it then passes to the EOSTCS and returns the status of the telescope and dome, which it receives from the EOSTCS.

The software control system will be required to implement dome thermalisation. Dome thermalisation retrieves the predicted nighttime temperatures from a web site maintained by the Bureau of Meteorology (BoM) and determines the minimum nightly temperature based on that data. The dome thermalisation process sets the chiller temperature set point to that temperature minus five degrees Celsius and also retrieves the chiller's status and passes that onto TAROS/CICADA where it will be stored in the central database. The temperature in the enclosure is controlled by a number of fan cooler units installed within the enclosure with the chilled water from the chillers passing through their heater exchangers. During the night, the software control system will be required to continuously monitor the external temperature from the on-site Met Station and adjust the chiller temperature accordingly.

By passing the plant status to the CICADA telescope service, the plant status will then be included into the CICADA status database and ultimately included in TAROS Health. TAROS Health will be responsible for maintaining a log of the status of each component of the system, and also for sending alerts of major failures either via a cell phone text message or email or some other such mechanism.

3. DESIGN AND IMPLEMENTATION

3.1 Detector Controller

CICADA did not previously support the StarGrasp controllers. Integration of the StarGrasp controllers required the creation of a new CICADA Camera Controller class. The University of Hawaii Institute for Astronomy (UHIfA) provided some of the source code they used for operating the StarGrasp controllers. The supplied source code was carefully studied to gain an understanding of the operation of the controllers and a new CICADA camera controller class was constructed based on that knowledge. The StarGrasp controller receives commands over TCP/IP and sends the pixel data over UDP. UHIfA developed a library called CATP to handle the transfer of pixel data from the controller to the host computer. The CATP library contains the intelligence to request retransmission of dropped pixel data. Each CATP packet contains a header and data. The header contains information including the detector ID pixel location on the detector for the data. The use of UDP for pixel data transfer means that the data can arrive in any order, and the header data is used to keep track of which pixels have arrived and which pixels are still outstanding.

It proved much simpler to use the CATP library as a stand-alone shared library which is called by CICADA rather than attempt to roll the CATP functionality into the camera controller class.

The camera controller class allocates a block of shared memory which is large enough to store all the pixel data for the full CCD array, and which maps out the pixel data as it would appear on the array. The camera controller class then uses the CATP library to receive the pixel data from the StarGrasp controller and interrogates the header data for the location of the pixel data to determine the appropriate location of the data within the shared memory. A counter within the controller keeps track of the data as it arrives. As each complete line of data is filled, the counter triggers the CICADA Data Process, running as a separate process on a different computer, which generates the FITS file up to the end of that line. This process continues until all the pixel data has been transferred and the FITS file completed.

3.2 TAROS Monitor

To meet the requirement for monitoring the status of many subsystems and taking action upon a subsystem failure, a Java based process was added to TAROS called TAROS Monitor. The TAROS Monitor subscribes to parameters in the central database via a mechanism, the TAROS Connection Class, which allows a client to subscribe to any parameter in the database and receive regular updates from TAROS as the selected parameters change value. The central database is constantly updated with the status of each subsystem. The TAROS Monitor uses a rule-based engine to determine which parameters to subscribe to, define the safe limits for each parameter, and the actions to take when a rule condition is met.

The rules based engine used in the TAROS Monitor is JBoss Drools. A PHP interface has been created to allow users to easily and quickly define the declarative rules to be applied to the system. The rules based engine allows users to construct complex rules combining parameters from many disparate subsystems into a single easy to understand text string. Each rule also contains an associated action that is performed when a rule is triggered. In most cases the action is to send an SMS or email to designated personnel.

The rules constructed within the TAROS Monitor are also used for regular maintenance procedures. Rules are constructed to trigger when a subsystem has executed a predefined number of duty cycles. In this case an email is sent to the maintenance crew, alerting them of the need for routine maintenance.

3.3 Interfacing to EOS Telescope and Dome

Interfacing to the EOS supplied TCS, EOS dome control and the Met Server are more easily treated in a single package, Plant Control, as the three components share common data and communication links and they are required to run in a Windows environment. Unlike UNIX, on a Windows platform the overhead in running a single, multi-threaded application with the separate threads sharing common data is considerably less than if the three threads are broken up into separate processes sharing common data via a mechanism such as interprocess communication (IPC). For this reason the three requirements were grouped together into a single application which has three separate threads running concurrently within the same application.

For SkyMapper, four components needed to be written:

3.3.1 CICADA telescope plug-in running under UNIX

CICADA communicates with telescopes using a simple API that is implemented as a plug-in, with a unique plug-in for each telescope. A new plug-in was created for the SkyMapper TCS. This plug-in runs on the Solaris TAROS computer.

3.3.2 TCS client running under Windows

The telescope plug-in communicates with the TCS client running on a Windows computer. This server accepts commands from the CICADA plug-in via a socket and translates them into EOS DML commands that are understood by the EOSTCS Server, which will be running on a different machine. The TCS client has been created using the EOS Control System Client API, which allows the EOSTCS client to communicate directly with the EOSTCS server.

The TCS client is also responsible for moving the dome shutters to track with the telescope while also acting as a wind block. The telescope and dome have a different centre of rotation, so the shutter positions do not have a direct relationship with the telescope. The shutter positions are continuously recalculated to track with the telescope and commanded to move to the recalculated position.

3.3.3 Telescope GUI plug-in running under Unix

In addition to these communication components, there is also the need for a telescope GUI plug-in to be written to support the EOSTCS. This is required for both local and remote operations. EOS has a high fidelity TCS simulator that is available for our testing purposes. This has meant that the new components have been tested well in advance of delivery of the SkyMapper telescope.

3.3.4 TAROS Compatible Met Server

The Met Server retrieves weather data from the EOSTCS using the EOS Control System Client API. To do this, the Met Server has a handle on the Telescope Client and accesses the EOSTCS through that handle. The Met Server will retrieve the latest meteorological data from the EOSTCS and allows clients to connect via a socket to gain access to that data. The Met Server will also update a central data store available to both the EOSTCS Client and the dome thermalisation threads of the application.

PlantControl – Will start each of EOSTCS Client and Met Server in their own thread and instantiate the singleton object MetData. Each of the three threads will exist for the life of the application.

MetData - Will be implemented as a singleton. All three threads will access the MetData object, so the thread's access to the data will need to be synchronised. As such the code for reading and writing of the data can be bundled up in a critical section. A critical section is a lightweight mechanism for allowing only one thread at a time to execute a given piece of code under Windows without the overheads of events, mutexes or semaphores, which are also used for multithreaded synchronisation. However, critical sections do not perform an expensive control transfer to kernel mode if a thread attempts to acquire an already-held critical section. When a thread enters a critical section, the MetData object is locked. When the thread leaves the critical section the lock is released and any waiting threads can gain access to the critical section.

Met Server – Implements a ServerSocket, which allows clients to connect to the Met Server and retrieve the latest weather data from the local weather station.

LightningDetection – Runs in its own thread, and implements a ServerSocket to listen to the lightning detector hardware, while also allowing clients to subscribe to it. The LightningDetection class also subscribes to the EOSTCS Server. When a likely lightning strike is detected, the LightningDetection class will park the telescope, close the dome via the EOSTCS Server PostWait method, and also send an appropriate notification message to any subscribers.

EOSTCS Client - Instantiates a ServerSocket which listens for TCS commands coming from CICADA or TAROS and sends TCS status messages back to CICADA via the ServerSocket. The EOSTCS Client also starts the worker thread running the LightningDetection thread and subscribes to the LightningDetection's onLightning callback.

3.4 Dome Thermalisation

To meet the requirements of dome thermalisation, a Linux single board computer has been deployed with a small independent process running which communicates with the chillers via the RS485 serial interface. At the cessation of observations for the night, the application interrogates the BoM web page and determines the lowest predicted temperature for the following night, then sets the chiller temperature set point to that temperature minus 5 degrees. To determine the time for resetting the chiller temperature, the application will interrogate the ephemeral database located on the external scheduler machine to retrieve the sunrise time for the following night and use that time to set an internal timer. If the predicted night temperature on the BoM web site has not been updated by sunrise, then the timer will be set to retrieve the data as soon as possible after the web site has been updated, and the sunrise time from the ephemeral database will not be required. Alternatively, the timer can be set to trigger the retrieval of the predicted temperature at frequent regular intervals through the period from sunrise to sunset.

After sunset, and during the course of the nights observing, the dome thermalisation routine will regularly monitor the on site meteorological weather data and adjust the chiller temperature to best match the outside temperature.

3.5 Control the Filter Assembly

The WFI instrument has a simple rotating filter wheel. CICADA controls the positioning of this wheel by using a CICADA instrument plug-in with an associated plug-in GUI module. The plug-in communicates via a serial line to the filter wheel controller. This simple mechanism is not suitable for SkyMapper.

The SkyMapper filter mechanism requires the control of 9 servomotors and 2 solenoids. The servo controller consists of a Galil controller with an Ethernet interface. The Galil supplied communication library has been ported to Solaris and integrated into CICADA. A SkyMapper filter control plug-in has been created which uses the Galil drivers to interface to the Galil controller via Ethernet.

The plug-in sends a simple string command specifying the positions of the filters as key/value pairs and receives an acknowledgement for the command. The plug-in then sends a command to execute the previously set values, receives an acknowledgement and then waits for a return of status when the command is completed.

The status of the filter housing mechanism is transmitted continuously by the Galil controller in a separate thread and on a separate Ethernet port. The filter plug-in listens to this port and continuously collects the status messages in a separate thread.

The time taken to change a filter will need to be less than 20 seconds.

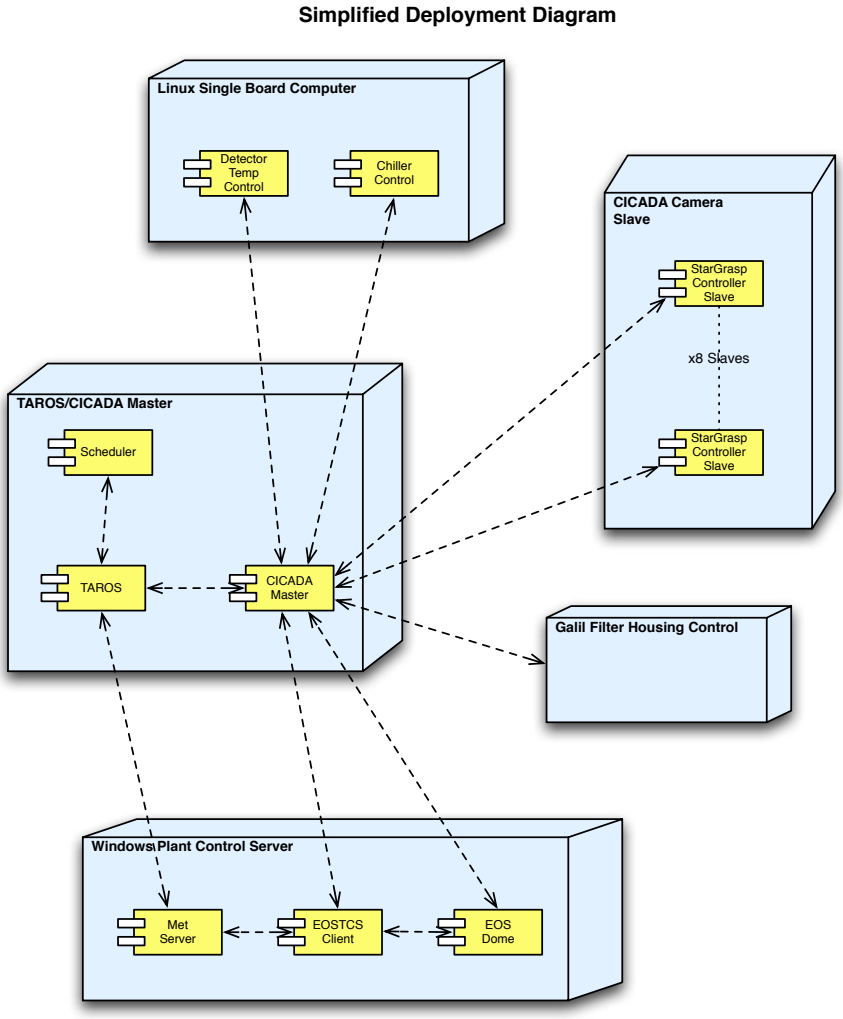

Fig. 2: Simplified Deployment Diagram of the SkyMapper Data Acquisition and Control System.

The Unified Markup Language (UML) Deployment diagram in Fig. 2 shows the layout of the software system. The TAROS/CICADA Master computer runs Solaris and the TAROS and CICADA cores are all run on that machine. The 8 instances of the CICADA camera controller's class needed to operate the 8 StarGrasp controllers run on the CICADA Camera Slave computer which also runs Solaris. The Windows specific EOS interface applications all run on a separate Windows computer. Dome thermalisation and detector temperature control are all contained on a separate single board computer running Linux Fedora Core 3.

4. PROGRESS

At this point, the SkyMapper software system has moved into Integrated System Testing (IST). The software for all the subsystems are ready for testing and so far have passed unit level testing.

5. DEVELOPMENT

The requirements for the software system have changed many times up to this point in the project and the design has been modified to reflect those changes.

The most significant change to the design, which carried with it the most significant risk for delaying the project, was the integration of the StarGrasp detector controllers. Initially the detector controllers were to consist of 4, San Diego State University (SDSU) detector controllers. The SDSU controllers have been integrated into CICADA for some time now

and the code is efficient and robust. The use of SDSU controllers in SkyMapper would have meant that no work was required for the detector controllers operation within the SkyMapper software system. For technical reasons it was decided that the StarGrasp controllers were more appropriate.

However, the development of the StarGrasp controller class progressed very quickly due to the well-defined architecture of CICADA and the support provided within the CICADA framework.

The CICADA plug-in architecture also meant that integration of SkyMapper specific components also progressed quickly and with comparative ease. The SkyMapper filter housing is like no other filter housing currently integrated into CICADA yet the API for the filter plug-in was sufficiently flexible to easily accommodate the SkyMapper filter control.

6. SCIENCE DATA PIPELINE

Over 5 years the SkyMapper telescope will undertake the Southern Sky Survey (SSS). The survey will result in a catalogue of the southern sky of better detail than any existing survey today. This is due to the speed and depth that SkyMapper is able to scan the sky, and the design of the survey itself.

For the survey, SkyMapper will generate up to 500 terabytes of image data and from that, detect and store precise information for one billion objects in the sky. The resulting dataset will be the largest publicly available in Australia, and is expected to be used by astrophysicists from around the world. This places a huge requirement on any institution aiming to host the dataset, hence the ANU Supercomputing Facility (ANUSF) was chosen due to the extensive storage and processing capacity available there.

To enable the production of the Southern Sky Survey a custom science data reduction pipeline (SDP) has been developed in association with the ANUSF to calibrate and catalogue the massive amount of image data. The SDP software suite comprises known astronomical off-the-shelf and modified software packages each tailored to a specific task. These tasks, or stages in the pipeline, are executed and managed by an overseeing control application that provides an interface from any particular multi-process scheduling platform. This allows the SDP to be moved between processing environments, for example the Portable Batch System used at ANUSF to a home built Linux based cluster. The control application is also sufficiently de-coupled from the pipeline stages so they can be added and removed with minimal code changes.

Although the SkyMapper telescope will run autonomously the SDP will require human intervention to approve certain stages. This is implemented though a GUI written as two Ruby on Rails[c] web applications, and custom Perl scripts interfacing to the Unix tool PGPLOT[d].

The pipeline processing stages can be considered in three groups: calibration, and the two observation types, 5 Second and Main. Calibration involves flattening the survey images to provide linearity across the survey lifetime. The 5 Second images are then catalogued and fed into the Main Survey stage to provide an anchor for object brightness and position to finally produce the Southern Sky Survey.

Fig. 3 details the processing stages within the SDP and maximum data rates expected. Also shown is the flow of data between the pipeline control, pipeline stages, database and image store. All elements pictured, except the observation scheduler, reside on hardware at ANUSF. The images will be stored on a robotic tape system and written to global disk when required for processing. The database is stored on a dedicated database cluster, and by design can be split amongst multiple nodes if necessary. The pipeline control application keeps track of the pipeline stages as they complete and will submit the next available stage to the PBS queue. The pipeline stages execute on one of the available clusters at ANUSF.

[c] http://www.rubyonrails.com/
[d] http://www.astro.caltech.edu/~tjp/pgplot/

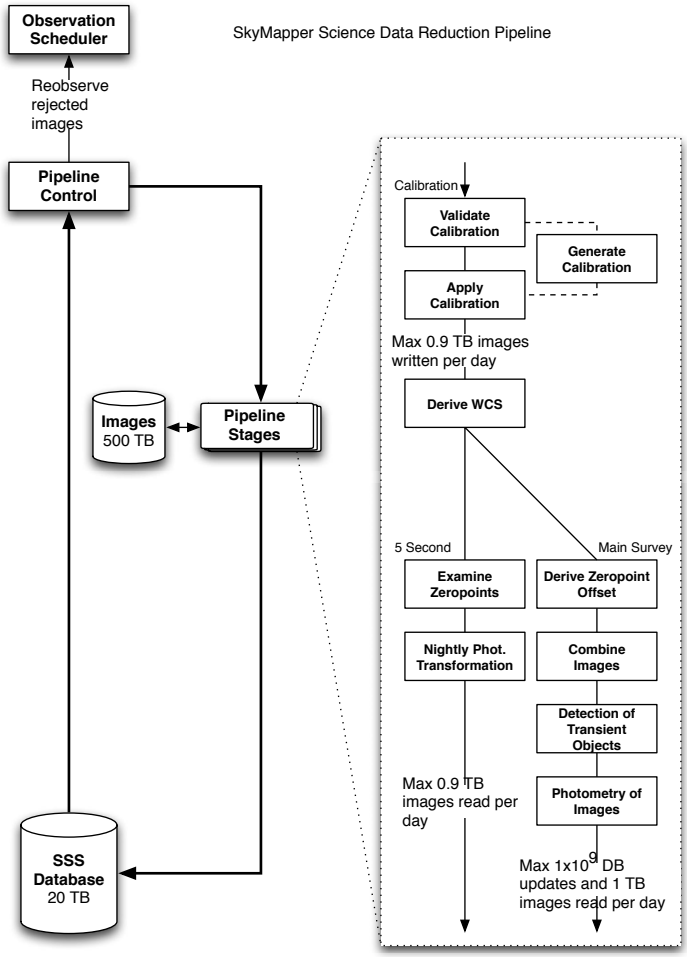

Fig. 3: Science Data Pipeline design and data rates

REFERENCES

1. Keller, S.C., Schmidt, B.P., Bessell, M.S., Conroy, P.G., Francis, P., Granlund, A., Kowald, E., Oates, A.P., Martin-Jones, T., Preston, T., Tisserand, P., Vaccarella, A., Waterson, M.F., The SkyMapper Telescope and The Southern Sky Survey, Publications of the Astronomical Society of Australia, Vol 24, Issue 1, 1-12 (2007)
2. Strauss, M.A., "Synergy between the Sloan Digital Sky Survey and large telescopes", Discoveries and Research Prospects from 6- to 10-Meter-Class Telescopes II, P Guhathakurta, ed., Proc. SPIE, 4838, 16-23 (2003).
3. Wilson, G.M., Czezowski, A., Hovey, G.R., Jarnyk, M.A., Nielsen, J.J., Roberts, W.H., Sebo, K.M., Smith, D.F., Vaccarella, A.L. and Young, P.J., Telescope Automation and Remote Observing System (TAROS), ASP Conf. Ser. 347: Astronomical Data Analysis and Systems XIV, 347, 563 (2005).
4. Young, P.J., Roberts, W.H. and Sebo, K.M., CICADA - Configurable Instrument Control and Data Acquisition, ASP Conf. Ser. 172: Astronomical Data Analysis and Systems VIII, 172, 115 (1999).
5. Kaiser, N., et al., "Pan-STARRS: A Large Synoptic Survey Telescope Array", Survey and Other Telescope Technologies and Discoveries, Proc. SPIE, 4836, 154 (2002).