PAST, PRESENT AND FUTURE OF THE ASKAP MONITORING AND CONTROL SYSTEM

Malte Marquarding, CSIRO Astronomy and Space Science, Epping, Australia

Abstract

The Australian Square Kilometre Array Pathfinder (ASKAP) is CSIRO's latest radio interferometer located in the Mid West region of Western Australia. It is in the final phase of construction with Early Science Program starting in the middle of 2016. This fully remotely operated telescope is one of the pathfinder telescopes of the Square Kilometre Array (SKA) building knowledge about infrastructure, technologies and scalability. This paper presents another incremental status update on previous reports. It focuses on software lessons learned and modifications arising from the initial six antenna test system. ASKAP's software consists of two major components, the Telescope Operating System (TOS) and the Central Processor (CP). This paper addresses the TOS, highlighting the monitoring and control aspects of the system.

PROJECT STATUS

The full ASKAP telescope consists of 36 dishes of 12m in diameter hosting phased array feed (PAF) detectors. The hardware is located in a specially designed central building which addresses self-generated radio interference. An initial set of six PAFs, the Boolardy Engineering Test Array (BETA) has been installed and commissioned. The evaluation of this system lead to an improved design with new 36 beam PAFs and an updated digital back end including RF over fibre. This is called ASKAP design enhancement (ADE) and resulted in most of the hardware being moved from the antenna dish pedestals to the central site building connecting to the dishes through fibre optic links. 12 ADE antennas with end-to-end integration of TOS and CP will form the Early Science Program platform. 25% of the total time will be assigned to science project during the time the rest of the telescope will be commissioned. It will produce full data products with only the reduced number of antennas separating it from full ASKAP operation.

Currently the project has evolved to encompass the following active test and production platforms.

- A Parkes 12m dish hosting prototype a PAF. This system is dedicated to engineering tests and new PAF developments such as improvements to the current system as well as next-generation SKA prototyping. Several prototypes have been tested. The software to operate this facility has been stable and remained largely unchanged for several years.
- The BETA system at MRO which host Mark 1 phased array feeds. It is used for beam forming research and small science projects run through the ASKAP Commissioning and Early Science (ACES) team. It has produced several reviewed astrophysics publications and has proven to be an invaluable tool for control system integration, deployment procedures and data management.
- A Mark 2 PAF prototype system at MRO. This single antennas system is dedicated to engineering testing. It is also used for testing hardware modification to the existing Mark 2 set up. Most recently feedback from ACES has prioritised the use of on-dish calibration devices (ODC) to simplify complicated calibration procedures. One such system is being commission on this antenna.
- A four antenna array hosting Mark 2 phased array feeds. It forms the base of the full production array. Commissioning of these systems is taking place at the moment encompassing interferometer phase closure, moving to 36 electronically formed beams and visibility streaming.

For more information about ASKAP visit the project's home page [1].

PROGRESSING THE ASKAP MONITORING AND CONTROL SOFTWARE

Architecture and Framework Update

As described above ASKAP project went through three major stages. A prototype phased array feed on a non ASKAP antenna co-located with the Parkes 64m telescope which still used for research into beam weight generation and stability. Beam weights are used to create beams according to constraints such as maximising the signal-to-noise for a given configuration. This engineering test system was also used to prototype the software technologies and formed the base line of the current system. As described in our previous update [2], the ASKAP software architecture is supported by two key technologies, EPICS and ZeroC ICE. EPICS handles the control and monitoring aspects through software input/output controllers (IOC), whereas ICE defines the interface between software components. Ice interfaces provide the main service contracts between the Central Processor and the Telescope Operating System (see Fig. 1).

We have been using the EPICS framework since the beginning of the project and have now added several helper modules to abstract common usage in ASKAP. For engineering and operator displays of monitoring and control ASKAP has now fully adopted Control Systems Studio (cs-studio). We have joined the project and ASKAP has become one of the products of the upstream source. As part of the collaboration we are in the final stages of setting up cloud-based continuous integration with open access to all collaboration partners. In addition to working with the cs-studio core pack-

ISBN 978-3-95450-148-9

age we have chosen the operator interface BOY and created plugins for streamlining visualisation of and access to large number of repeating records. Alarms are handled through BEAST which provides a backend for alarm management and metadata storage associated with alarm handling such as acknowledgment, guidance and broadcast. It provides full cs-studio integration for client visualisation. The overall connection between EPICS integration, user interfaces and alarm handling has proven to be major benefit to operating the instrument and has justified the choice of technologies.

We have added and refined the use of following software packages since the last project update:

- ASYN [3] wrappers for EPICS driver development with customisations
- autosave with analysis tools for extracting changes in realtime values.
- Exclusive use of cs-studio [4] and BOY for GUI development for engineering
- We have moved to the Control System Studio (cs-studio) alarm system BEAST.

ASKAP Design Enhancements

The ASKAP Design Enhancements (ADE) Program introduced several enhancements to the EPICS based Telescope Monitoring and Control System. Firstly a software layer called common library has been added between the hardware and EPICS IOCS. It provides a cleaner and implicitly documented interface definition. ASKAP consists of over 4000 instances of individually addressable hardware components each with hundreds of monitoring and control points. All devices need to be efficiently configured, controlled and monitored as a single entity. To assist with this we developed Composite IOCs and Summary Record to create a scalable control and monitoring point hierarchy on top of the flat structure of EPICS.

Composite IOCs represent logical partitioning of the telescope into higher level groups that can be controlled and monitoring as a single entity. Composites exist for each antenna and a single composite for the entire telescope array (see Fig. 1). Configuration and control is fanned out from the array composite IOC through to the antenna composite and down to antenna subsystems and ultimately each hardware component (e.g. FPGA). Similarly monitoring information is aggregated up from the lower level devices using summary records. The EPICS database for the composite IOCs are largely automatically generated from a single point definition file and handles such things as control of heterogenous subsystems, sequenced control and masking out of antennas or subsystems not in use.

Summary records are automatically generated and provide aggregation of a set of monitoring points and will propagate alarm status. Hardware that is disabled, not present or not in use is masked out dynamically at runtime. Summaries exist at all levels in the monitoring hierarchy and include summaries of summaries. E.g. at the top level Array Composite IOC a single summary point can represent the alarm state of all temperatures in the system.

Table 1: ASKAP Monitoring and Control System Specifications Update

	BETA (6 Ant)	**ASKAP (36 Ant)**
Total Number of I/O Points	30,000	800,000
Total Number of Archived I/O Points	25,000	800,000
Monitoring data archival rate	130 GB/year	1.5 TB/year
Monitoring data archival rate max	1 kHz	1 kHz
Highest "soft" control loop rate	1 Hz	1 Hz
Estimated number soft-IOCs	35	230
Estimated number soft-IOC Linux computers	8	11
Science "raw" data output rate	400 MB/s	3 GB/s

With the introduction of the common library abstraction and automation has also been introduced to the generation of EPICS records for the individual IOCs. Record metadata such as monitor archiving policies, alarms and cs-studio BOY elements can be generated through the build process. This simplifies repetitive elements across a large number of antennas, subsystems, cards or points. A summary of the current control system specification is described in Table 1. This presents an incremental update to the previous specifications.

Development Tools Revisited

The complexity and interaction between the different development groups in hardware and software was not adequately supported by Redmine. We have adopted JIRA [5] as the issue management system which allows for sub-projects with issues freely movable between these. It also provides the tools for better planning of milestones and releases. This underpins the TOS move towards agile development techniques. For continuous integration a minor move has been undertaken from hudson to jenkins [6]. This tools is also used to create release deployment artefacts. For our TOS target platform debian linux jenkins automatically generates packages for code release. The CP code base has been moved into a separate repository as target platforms and developer audiences are very different. Connection is maintained via the Ice service contract interfaces.

ISBN 978-3-95450-148-9

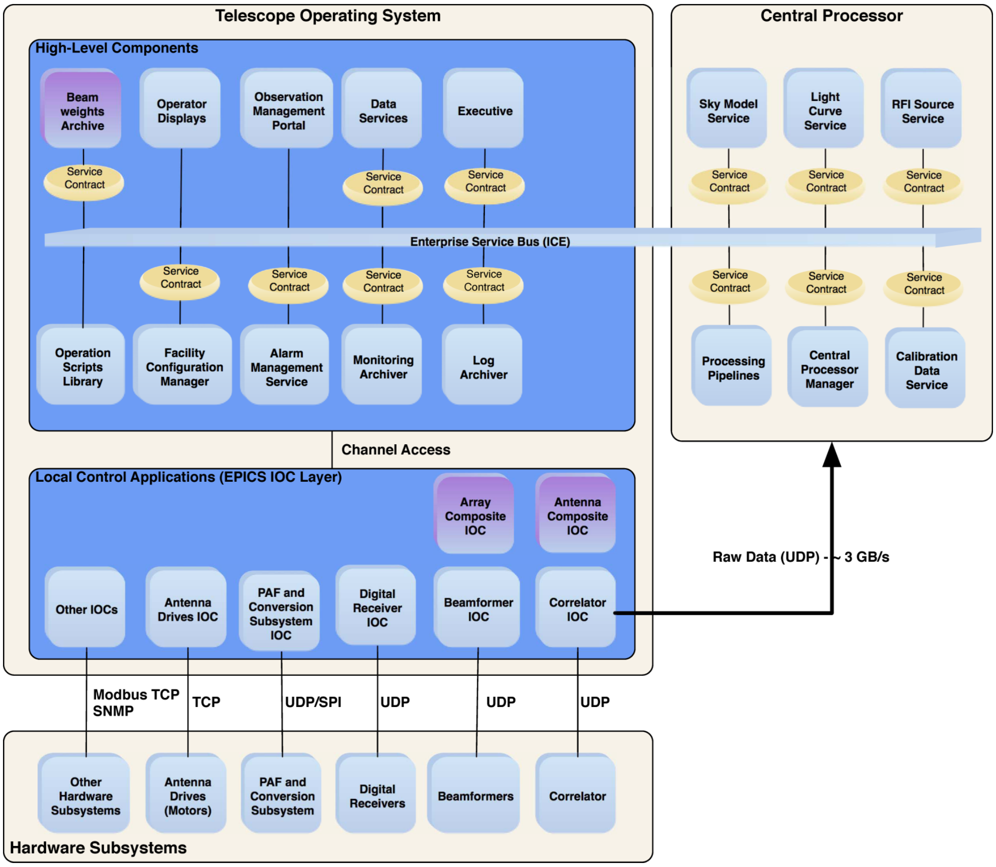

Figure 1: Logical view of the ASKAP software as implemented in latest version of TOS. Purple components are new additions.

CURRENT AND FUTURE CHALLENGES

Improvements and Upgrades

- TOS and CP integration and commissioning will be a major effort as these two software components are hosted at different physical sites with different host organisations. Another related task is to handle configuration management across all of ASKAP from firmware revisions to operational parameters.
- The current drives motor IOC was developed before the introduction of composites and lacks the benefits of aggregation and fan out over large number of similar devices. A re-design is required to scale to the full ASKAP array.
- We are in the process of moving to cs-studio version 4 which addresses issues encountered during current version. cs-studio is changing the implementation of BOY which will affect our automation and customisation.

Beam Weights

As phased array feeds are cutting edge technology in astronomy the understanding of dealing with these instruments is still a rapidly evolving process with open constraints. The TOS needs to cope with frequent changes to requirements and operations while maintaining reliability. We have updated the TOS architecture to add a beam weights component to deal with this (see Fig. 1). Data rates for capture and the beam weights processing can be comparable to the data rates for the standard (visibility) data. To make it valuable a rich set of metadata needs to be maintained as well. This can have deeper impacts on the architecture and design which still have to be evaluated. New auxiliary devices such as ODCs are coming online to assists in the beam forming methods and operations. This is the exciting new technology field in radio astronomy which will keep scientist and engineers busy for a long time.

REFERENCES

[1] ASKAP website http://www.atnf.csiro.au/projects/askap

[2] J.C. Guzman, Status of the ASKAP Monitoring and Control System, ICALEPCS'11, Grenoble, October 2011 http://accelconf.web.cern.ch/AccelConf/icalepcs2011/papers/frcaust04.pdf

[3] ASYN EPICS driver framework website http://www.aps.anl.gov/epics/modules/soft/asyn

[4] Control System Studio website http://controlsystemstudio.org

[5] JIRA software development tool website https://www.atlassian.com/software/jira

[6] Jenkins Continuous Integration https://jenkins-ci.org