ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-TUBR02

DESIGN PATTERNS FOR THE SKA CONTROL SYSTEM

S. Vrcic, SKA Observatory, Manchester, UK

Abstract

Square Kilometre Array Observatory (SKAO) comprises two radio-telescopes: the SKA Low Frequency Telescope, located in the Murchison Region, Western Australia, with the observing range 50 – 350 MHz; the SKA Mid Frequency Telescope, located in the Karoo Region, South Africa, with the observing range 350 MHz – 15 GHz. The SKA Global Headquarters is in the Jodrell Bank Observatory, near Manchester, UK. The SKA Low and SKA Mid Telescopes use different receptors (log-periodic antennas vs offset-Gregorian dishes), otherwise the telescopes share the same design concept, the design of many components, in particular software, is common to both Telescopes. Work on construction, which officially started on the 1. July 2021, was preceded by the extensive design effort which culminated in successful completion of the Critical Design Review (CDR). This paper describes the design patterns defined so far for the implementation of the Telescope Control System (and applies for both Telescopes).

INTRODUCTION

The Square Kilometre Array Observatory (SKAO) is an International Organisation that comprises two radio-telescopes and spans three continents:

- The Global Headquarters (HQ) is located at the Jodrell Bank Observatory near Manchester, UK.
- The SKA Low Telescope, located in Murchison Region in Western Australia operates in the frequency range 50 – 350 MHz.
- The SKA Mid Telescope, located in the Karoo region in South Africa, operates in the frequency range 350 MHz to 15 GHz. The SKA Mid observing range is divided in 6 bands; each receiver can collect and process data for one band at a time; instantaneous bandwidth varies from band to band.

The SKAO was instigated by an idea to build a radio-telescope with a collecting area of 1 km^2. The science goals are described on the SKAO website [1].

Instead of building a single gigantic dish, radio-astronomers use a technique called interferometry. A Radio-Interferometer superimposes electromagnetic waves collected by multiple receivers to amplify the signal of interest and eliminate signals generated by the ground-based sources and by other man-made equipment (including the telescope itself). The signals generated by sources not of interest are generally referred to as radio-interference (RFI). The SKA Telescopes are interferometers.

The SKA Low Telescope comprises 131,072 log periodic antennas, organised in 512 stations. The stations are placed so that the Telescope consist of the densely populated core and three spiral arms with receding density. Signal collected by the antennas is digitized, the beams are formed using input from the antennas the belong to the same station, the beams formed by different stations are aligned in time, and correlated (input from each pair of stations is complex cross-multiplied) to extract information of interest and eliminate RFI. Integration in time and/or frequency may be performed to reduce the amount of output data. Up to this point, data processing is performed in real-time. The output of correlation is then captured, data sets formed and stored for further processing.

The SKA Mid Telescope comprises 197 offset-Gregorian dishes, each with the diameter of 15 meters. To cover the required frequency range, each dish is equipped with several receivers, but only one can be placed at the focus at any given time (consequently, before starting a new experiment a receiver may need to be replaced). The signal captured by the single pixel feed is digitized and processed in the same manner as in the Low Telescope.

After many years of preparation, the construction of the SKA Telescopes officially started on 1. July 2021 (artist impression of the SKA Telescopes shown in Fig. 1).

During the pre-construction phase design were developed and reviewed for each part of the Telescopes, including the Control System. The general principles for the design of the SKA Telescope Control System were established relatively early in the design process. Following the Critical Design Review (CDR), while preparing for transition to construction, work on the detailed design and development of the Control System software started, although with the limited resources. The lessons learned over that period are being applied as we update documentation, improve the development environment, and refine the design, including the design patterns.

This paper provides an overview of the design approach and patterns used in the SKA Control System, with the goal to collect input from the colleagues working on similar projects.

Figure 1: Artist impression of the SKA Telescopes.

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-TUBR02

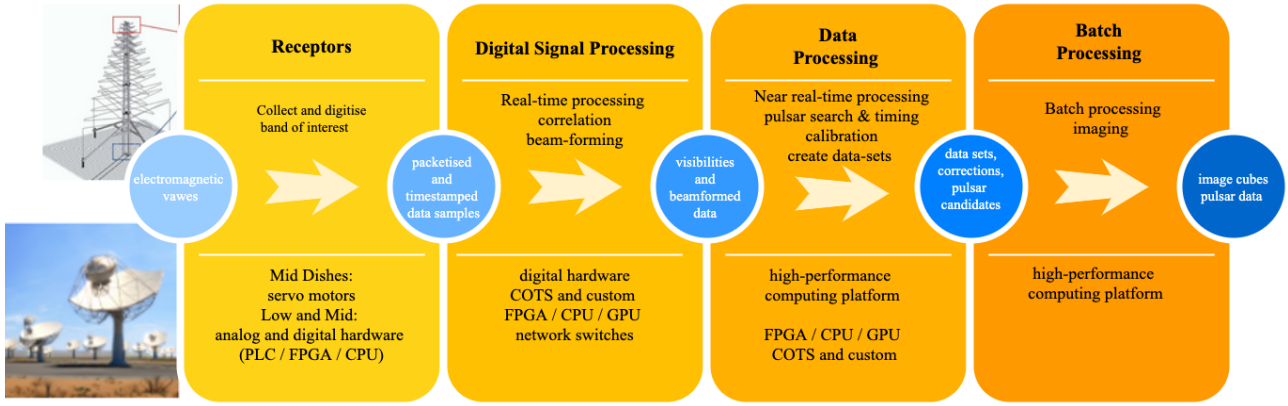

Figure 2: SKA Telescope functional overview and the technologies used.

SKA TELESCOPE CONTROL SYSTEM REQUIREMENTS

Figure 2 provides a high-level overview of the functionality required and technologies used to implement the SKA telescopes. With exception of the antennae, similar technology is used in both telescopes. Analog hardware is used in receivers; once the collected signal is digitized, digital hardware and computers are used for signal processing and to packetize, transmit, receive, and store astronomical data as it is passed from one sub-system to the other. The SKA Telescopes use computing hardware equipped with Field Programmable Gate Arrays (FPGAs), Graphical Processing Units (GPUs) and Central Processing Units (CPUs). The log-periodic antennae in the SKA Low Telescope do not have moving parts. The dishes in the SKA Mid Telescope are equipped with servo motors that move the dish to point the dish focus towards the desired source and rotate the indexer (the tray where receivers are placed) to place one of the receivers in the focus.

The requirements for the Telescope Control System can be classified as: requirements related to monitoring and control of the equipment and requirements related to monitoring and control of the experiments (in the context of SKA referred to as observations).

Equipment Monitoring and Control

Requirements related to monitoring and control of the equipment and software are standard requirements for the similar systems used in industry and scientific facilities. The following list is a brief overview of the required functionality:

- Monitoring (report status, periodically and on request, report state and mode changes, errors, and faults).
- Archiving (periodically report and store the status of all the components and systems of interest).
- Provide interfaces that allow operations to change configurable parameters and trigger state/mode changes. Both Human Machine Interfaces (HMI) and Machine to Machine Interfaces (MMI) are required.
- Support for hardware / software / firmware updates.
- Support for debugging, testing and maintenance.

The above listed requirements apply for all the levels of the monitor and control hierarchy: the telescope, each subsystem, and individual components.

The components and subsystems differ in complexity, as does the number of attributes, configurable parameters and commands exposed by different subsystems and components. For each component and subsystem, it must be determined what is the information of interest and how it should be represented; this will be an on-going activity during the detail design, development, integration, and commissioning.

Graphical User Interfaces (GUIs) that display overall status of each telescope, status of the major subsystems and interfaces and the list of active alarms will be provided. In addition, more detailed 'engineering' views will be provided to be used during the development, integration, testing and commissioning; and will continue to be used in operations by the maintenance staff. The plan is to arrange the views of the telescope hierarchically so that a click on a faulty component opens a more detailed view of the component.

Figure 2 shows the telescope functions and equipment directly involved in capturing and processing astronomical data. Figure 2 does not show functionality and equipment provided by infrastructure, such as: facilities, roads, lending strips, power, water, networking, IT equipment, and masers and other equipment that provides very precise time required for telescope operations. Most of the infrastructure comes with own COTS monitor and control systems, and User Interfaces (operator and maintainer consoles). Also not shown are Data Archives: Observing Data Archive, Scientific Data Archive, and Engineering Data Archive. The Telescope Control System will communicate with the COTS systems to collect information of interest (overall status, significant events and alarms), but will not duplicate all information. Also not shown are Data Archives: Observing Data Archive, Scientific Data Archive, and Engineering Data Archive.

Observation Monitoring and Control

Observing Modes Both SKA telescopes support multiple observing modes: imaging, pulsar search, pulsar timing, and VLBI beam-forming, and several sub-modes for each observing mode. In addition, to support exploration of the transients, both telescopes are able to capture and, on re-

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-TUBR02

quest, offload a limited amount of captured raw data for each receptor. The Telescope Control System API (Application Programming Interface) shall allow users to configure a telescope for the desired observing mode and take advantage of the telescope versatility; in other words, the Control System shall not restrict users to a limited number of pre-defined configuration sets.

Concurrency and Commensality The SKA Telescopes provide processing resources for concurrent processing in all 4 observing modes; thus, allowing search for pulsars and precise timing of the known pulsars to be executed concurrently with surveys and imaging observations. The Mid Telescope, due to a huge instantaneous bandwidth (5 GHz), provides a limited concurrency for Band 5 observations.

Subarrays Not all science observations require all the receptors, some require only the receptors in the core while others are mostly interested in long baselines and therefore use only a small number of receptors in the core. To provide for efficient use of observing time, both telescopes are required to allow operations to subdivide the array (receptors and processing resources) and operate each subarray as an independent telescope, in terms of the observing mode, and the start and stop time of the observations. The requirement to support subarraying provides for more efficient use of resources but increases complexity of the Control System (and other subsystems).

Observation Monitor and Control – Key Functions The following is a brief overview of the Telescope Control System functions related to observation monitor and control:

- Assign resources (receivers and processing resources) to a subarray.
- Configure subarray for the observing modes. A number of parameters can be specified for each observing mode; a complete sub-array configuration is passed in the form of ASCII (American Standard Code for Information Interchange) encoded JSON (JavaScript Object Notation) script.
- Point the Mid Telescope dishes, track sources.
- Provide regular updates for delay and phase tracking corrections (to be applied during signal processing).
- During the observation, pass the updates received from the Science Data Processor (SDP) to the components used by the subarray, these include parameters for pointing and steering beams (station beams in the Low Telescope; beams for pulsar search, pulsar timing and VLBI in both telescopes), antennae weights to be applied in beamforming, RFI extinction related parameters and more.
- Start and stop tracking and data processing.
- Store data products (data sets).
- Release resources from a subarray.
- Derive and report overall status of the sub-arrays.
- Keep track of resource status and allocation, report status changes periodically and as requested.

Non-Functional Requirements (NFRs)

The key non-functional requirements for the Telescope Control System are:

- **Resiliency** – ability to function in presence of errors and faults, and to recover from errors and faults. Resiliency is extremely important for the Control System, which should be able to report status and accept commands even when much of the telescope equipment is only party functional, non-responsive or failed. Control System must be able to continue functioning when some of the components are unresponsive or provide unexpected response, and to resume normal operations when previously non-responsive or failed components recover.
- **Reliability** – Control System must be reliable, it should be the last component to fail. In all circumstances, the Control System should be able to reliably monitor and report status of other components, execute requests and commands and raise alarms when needed.
- **Performance** – Control System shall execute requests and detect and report status changes, errors, and alarms in timely manner. (The exact meaning of 'timely manner' is quantified in the actual requirements).

The key Control System related design decisions are to a great extent influenced by the concerns related to resiliency, reliability, and performance. Development of the SKA Telescope Control System is still in the early stages, we still have to prove that the design choices result in implementation that meets the requirements.

SKA TELESCOPE CONTROL SYSTEM DESIGN

This paper summarizes key design decisions and provides an overview of the design patterns for the implementation of the SKA Control System.

Distributed Control Logic

Each SKA Telescope consists of several major subsystems. Receptors (dishes in the SKA Mid Telescope, antennas/stations in the SKA Low Telescope), Central Signal Processor (CSP) and Science Data Processor (SDP) capture and process astronomical data. Other sub-systems provide infrastructure, communication networks, very precise timing, and preparation and execution of observations and telescope monitor and control.

Given the number of receptors and the signal and data processing capacity required, each sub-system consists of hundreds of hardware and software components. To handle complexity and size of the system, the sub-systems are further decomposed in smaller sub-systems and products.

Logic required for monitor and control is embedded in each sub-system and hierarchically organized. Each 'level' performs aggregation and reports overall status of the subsystem or component. Configuration and commands are passed from top to bottom, at each 'level' the higher-level parameters are translated into detailed configuration of the

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-TUBR02

subordinate components. This approach reduces complexity of individual software components and enables scalability and modifiability – important quality attributes as the SKA Observatory is expected to be used for at least 50 years; during that time refresh of the digital hardware and computing platforms is expected, as well as expansion of the array and addition of new observing and/or processing modes.

Physical vs Functional View

In an interferometer, the functional view does not always directly map to the physical view. The requirement to operate each sub-array as an independent telescope further complicates 'mapping' of the functionality to physical equipment. SKA Control System provides two views:

- Physical (equipment and components).
- Functional (subarrays, resources, and capabilities).

SKA Base Classes

A set of the SKA Base Classes has been developed to be used as the base for development of the monitor and control software. All components and sub-systems implement the same set of state and mode indicators, and commands to trigger state transitions. Each sub-system is required to implement a Master Controller which performs aggregation of the status and coordinates execution of commands. The Maser Controller:

- Monitors status of all components and based on the collected information derives overall status of the sub-system.
- Implements a set of commands that can be used to trigger state transitions, update software (and firmware), set logging level and more.

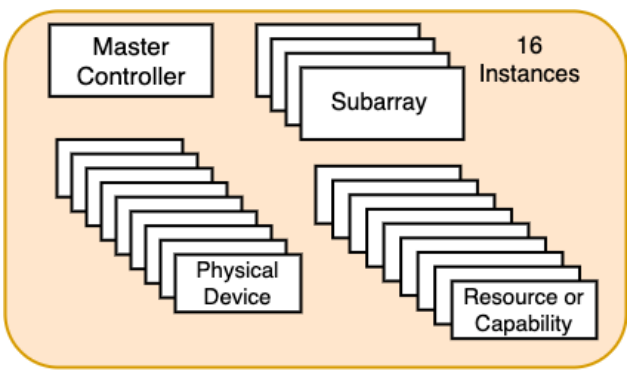

Figure 3: Design Pattern for implementation of monitor and control for sub-systems that process astronomical data.

Figure 3 shows the design pattern for monitor and control of sub-systems on the signal chain. Each sub-system implements:

- Master Controller which reports overall status of equipment and components, implements commands to trigger state transitions, upgrades, and other housekeeping tasks.
- Monitor and control for all physical devices.
- Multiple instances of Subarray (presently 16), which provides monitor and control functionality related to configuration and execution of observations.
- Monitor and control for resources and capabilities, i.e. entities that implement individual functions (in most cases implemented as FPGA bit streams and software pipelines).

TANGO Controls Framework

Early in the design phase a decision was made to use TANGO Controls [2] as the base for implementation of the SKA Telescope Control System. TANGO Controls is an open-source toolkit that can be used for controlling any kind of software and hardware.

From the design point of view the key concepts are:

- TANGO Device – a software component (instantiation of a TANGO Device Class). Each TANGO Device models a device, software component, or sub-system.
- Tango Device Server – provides an execution environment for one or more TANGO Devices.
- TANGO Data Base – all TANGO Devices register with the TANGO DB which can be used to obtain the addresses of registered devices, and then access devices directly.

TANGO uses CORBA (Common Object Request Broker Architecture) [3] for request-reply communication and ZeroMQ [4] for publish-subscribe.

TANGO supports development in C++, Java and Python. SKA Control System is developed in Python.

Decoupling TANGO API from the System Logic

As already mentioned, the SKA Control System is hierarchically organized and provides the same API at all levels of monitor and control hierarchy. This implies 'deep adoption of TANGO", i.e. TANGO API is provided for all sub-systems and components.

Unwanted consequences of this approach are:

- The framework (TANGO) becomes entangled into every aspect of the control system logic; the Control System becomes overly dependent on TANGO.
- Confusion regarding the significance of the status indicators, i.e. how to distinguish the TANGO Device (software model) from the device (sub-system or software construct) being modeled.

The SKA solution for those issues is to decouple the TANGO API (layer) from the 'system logic', so that the TANGO Device provides only the API, while the logic required for monitoring, status aggregation, error and fault detection, execution of commands, etc. is handled by a non-TANGO class, the so-called Device Manager.

Loose Coupling of Components

In the case of the SKA Telescopes, due to large number of components:

- The FMECA (Failure Modes, Effects and Criticality Analysis) shows that, once the full system is deployed, at any given time some of the components will be out

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-TUBR02

of service or undergoing maintenance. This means that Control System must be able to correctly function in presence of errors and failures. The Control System must be able to handle situations where one or more components become unresponsive, and to resume normal operations once the components become available. Also the Control System should not expect individual components to complete state transitions in a particular order, each component may transition to ON, STANDBY or OFF individually, or may become unresponsive.

- Execution of commands at a telescope or sub-system level will take significant time, in some cases tens of seconds. This is particularly true for transitions that increase or decrease power consumption (ON/STANDBY/OFF), where delay is intentionally introduced to control inrush power. Reconfiguration of a subarray observing modes also may require significant time, as the commands propagate through the hierarchy of components, and the FPGAs get reprogrammed.

Although different in nature, the above-described aspects of the telescope behaviour may be addressed using the same technique, or more precisely by using techniques that promote loose coupling of components.

The SKA Control System uses the following techniques:

- Asynchronous communications. The commands that require I/O, must to be forwarded to subordinate components or sub-systems and/or require communication over the network, or access to hardware are implemented as non-blocking, meaning that the originator of the command is not blocked waiting for the command to be executed. The server (TANGO Device) immediately acknowledges receipt of the command and returns command to the caller; during the command execution, the device on request can report the progress, once the command is completed the originator is notified via an event (status change).
- Input queue for commands – When received, a command is added to the end of the input queue; the worker thread executes commands one-by-one. The input queue is implemented to handle exceptional situations, when the system is under stress due to unexpected events; the system will be tuned so that, under normal circumstances, the queue is empty 90% of time. Not all devices have to implement the input queue.
- The subarray observing mode configuration is passed in a form of ASCII encoded JSON script, which contains the complete configuration for a single unit of observing. Use of data-centric self-contained messages and simple common types in data model support loose coupling.

SUMMARY

The following is the summary of the key design decisions and patterns chosen for implementation of the SKA Telescope Control System:

- Hierarchical organization.
- Distributed control of process logic.
- Decouple physical and functional view.
- Set of SKA Base Classes.
- Standard set of state/mode indicators, and commands that trigger state/mode transitions.
- TANGO Controls framework.
- TANGO API provided at all levels of hierarchy.
- Decouple TANGO API from “system logic”.
- Use JSON to pass configuration messages.
- Use asynchronous communications.

Development of the SKA Telescopes is still in early stages, and the team still has to prove that the design choices made so far are adequate for the challenging requirements.

ACKNOWLEDGEMENTS

Design for the SKA Telescope Control System is result of many years of work of many SKA team members, this paper a brief overview of the status.

REFERENCES

[1] SKA Observatory, https://skatelescope.org

[2] Tango Controls, https://www.tango-controls.org/

[3] Common Object Request Broker Architecture™, https://www.corba.org/

[4] ZeroMQ messaging library, https://zeromq.org