STATUS OF THE LOCAL MONITOR AND CONTROL SYSTEM OF SKA DISHES

S. Riggi*, U. Becciani, A. Costa, A. Ingallinera, F. Schillirò, C. Trigilio, INAF-OACT, Catania, Italy
G. Nicotra, C. Nocita, INAF-IRA, Bologna, Italy
V. Baldini, R. Cirami, A. Marassi, INAF-OATS, Trieste, Italy

Abstract

The Square Kilometer Array (SKA) project aims at building the world's largest radio observatory to observe the radio sky with unprecedented sensitivity and collecting area. In the SKA1 phase of the project, two dish arrays are to be built, one in South Africa (SKA1-Mid) and the other in Western Australia (SKA1-Survey). Each antenna will be provided with a local monitor and control system, enabling remote operations to engineers and to the Telescope Manager system. In this paper we present the current status of the software system being designed to monitor and control the dish subsystem. An overview of the dish instrumentation is reported, along with details concerning the software architecture, functional interfaces, prototyping and the evaluated technologies.

THE SKA DISH ARRAY

The Square Kilometer Array (SKA) project [1] is being designed to carry out radio observations of the sky in the wide frequency range from 50 MHz to 20 GHz with high sensitivity and collecting area to throw light in key areas of astrophysics and cosmology [2].
To this aim three different arrays were planned to be built for SKA Phase I and deployed at two different sites: the SKA1-Mid array at the South Africa's Karoo region hosting 190 dish antennas incorporating the Meerkat precursor telescopes and observing in the 0.35-13.8 GHz domain, the SKA1-Survey array at the Western Australia's Murchison region (MRO) site hosting 60 dishes equipped with phased array feeds (PAFs), incorporating the ASKAP precursor telescopes and observing between 0.65 and 1.67 GHz, and the SKA1-Low at the MRO site hosting 250000 low-frequency antennas.
The SKA design foreseen for Phase 1 was recently updated after a re-baselining process [3] carried out by the SKA Office to constrain the project costs to the established cost ceiling. The major outcomes of such review impacting dish array were a reduction in the SKA1-Mid array to 70% of its original size (190 dishes), equipped with prioritized band 2 (0.95-1.76 GHz), 5 (4.6-13.8 GHz) and 1 (0.35-1.05 GHz) and the deferral of SKA1-Survey array. Antennas (64) from the Meerkat precursor will be integrated in the SKA1-Mid array. The achieved array layout will have a central core component and additional spiral components targeted to deliver baseline lengths of 150 km. In the rest of the paper we will focus on SKA1-Mid antennas only.

* sriggi@oact.inaf.it

DISH INSTRUMENTATION

The general design for a SKA1-Mid antenna, shown schematically in Fig. 1, currently foresees a 15-m offset Gregorian dish with a feed-down configuration equipped with wide-band single pixel feeds (SPFs) for Band 1-2-5. The feed packages are mounted on a fan indexer at the focal position, allowing for changing among the available frequency bands. Only one band will be available for observation at any given time. A RFI-shielded cabinet is present in the antenna pedestal to house digital electronics and hardware for antenna movement and monitoring and control purposes, including the computing equipment supporting the Local Monitoring and Control system. No active cooling will be available for equipments inside the cabinet, only ventilation. Four sub-elements are identified in the SKA-Mid1 dish element: the Dish Structure (DS), the Single Pixel Feed (SPF), the Receiver (Rx) and the Local Monitoring and Control (LMC). Additional details on the first three are reported in the following subsections, while the LMC design is discussed in final sections.

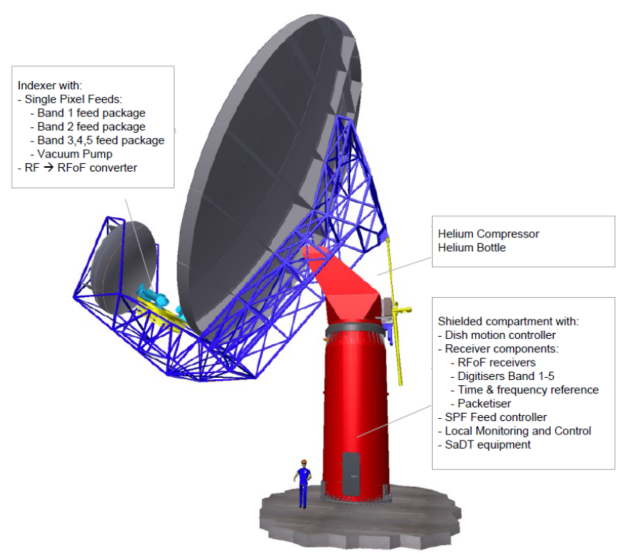

Figure 1: Schematic overview of SKA Dish design and instrumentation.

Dish Structure

The Dish Structure (DS) sub-element is responsible for the design of antenna, including mechanical structure and support, optics, reflectors, indexer as well as for the power distribution to be supplied to all sub-elements and safety systems (i.e. fire/intrusion sensors, limit switches, ...). DS

ISBN 978-3-95450-148-9

will supply the servo systems for antenna positioning in both azimuth and elevation comprising servomotors and encoders for precise position reading, connected to the Antenna Control Unit (ACU), placed inside the pedestal compartment. Three different antenna prototypes have been built within the DS consortium and tested against performance requirements derived from scientific goals. A down-selection among the antenna candidates is expected within few months.

Single-Pixel Feed

The Single Pixel Feed (SPF) sub-element will provide the feed packages of the antenna that receives the astronomical radio signals and relative control equipment. Each feed package will include orthomode transducers (OMTs) and low noise amplifiers (LNAs), prototyped for the SKA SPF bands, along with a Gifford McMahon (GM) cryogenic cooler to cool the LNAs, a cryostat assembly sharing a common helium supply system with a single helium compressor located at the antenna yoke, with helium supply lines routed to the indexer. A rotary vane vacuum pump is located on the indexer with vacuum lines to all connected cryostats. A single controller placed in the pedestal interfaces with the SPF feed packages, helium and vacuum systems using a low speed serial protocol over fibre, and allows external monitoring and control operations performed by LMC through a controller application software implemented on an ARM architecture.

Receiver

The Receiver (Rx) sub-element provides the hardware equipment to digitize the RF signal (e.g. the science data) received from each SPF band after a RF-to-optical conversion performed at the indexer. Digitization occurs inside the pedestal enclosure. A number of components are present. The digitiser packetizes and transmits the signal over a high-speed Ethernet link to the central signal processor. A Master Clock timer unit receives time and frequency reference inputs externally (from SKA SaDT element) and generates timing and frequency references where needed, including the control of the calibration noise source that is transmitted over fibre to the feed packages. As for SPF, a central controller is present and acts as a single point of monitoring and control to the LMC sub-element.

THE DISH LMC SYSTEM

The SKA M&C Hierarchy

The monitoring and control of the SKA constitutes a big challenge given the large and heterogeneous number of instrumentation to be remotely managed, including the integration of SKA Meerkat and ASKAP precursors. The overall number of monitoring points is roughly estimated of the order of 10^5 for SKA Phase 1 and larger than 10^6 for SKA2 [4], each antenna contributing with few hundreds parameters for SKA1-Mid and few thousands for SKA1-Survey antennas and PAF receivers. In this context the monitoring and control data flow provided by each SKA dish is expected to be of

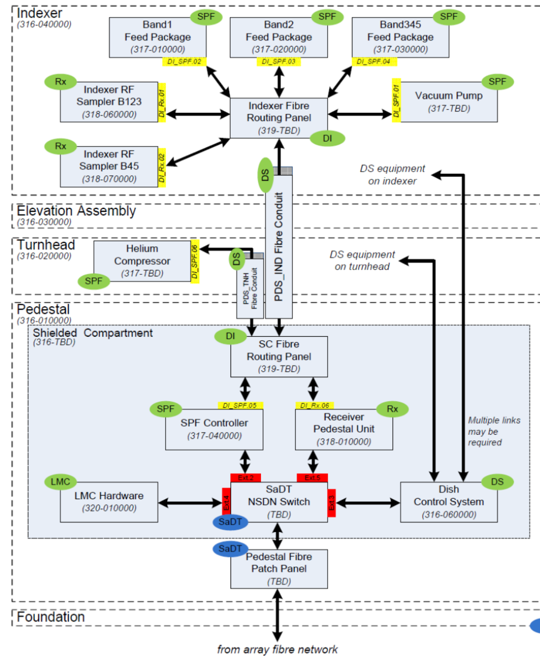

Figure 2: Schema of monitoring and control interfaces among SKA dish sub-elements.

the order of 100-200 kbps for SKA1-Mid and SKA1-Survey antennas and 1 Gbps for SKA1-Survey receiver.
To deal with complexity and ease system scalability, a hierarchical architecture of the M&C system was considered. The Telescope Manager (TM) element at the top of the hierarchy orchestrates the scientific observations, centrally coordinating all the involved telescopes and providing control information to them. Further details on TM design and organization will be given at this conference [5]. In order to perform the telescope monitoring and control TM communicates with the Local Monitoring and Control (LMCs) elements of the other SKA subsystems (one per each SKA subsystem, i.e. Dish, Central Signal Processor (CSP), Science Data Processor (DSP), . . .). These LMCs perform the direct control and monitoring tasks on their subsystems, fulfilling the following main responsibilities:

- maintain direct and local connectivity to controlled components, specially in case of TM downtimes
- receive and execute control commands (i.e. setup, configuration, calibration, scheduling, life-cycle, . . .) from TM involving dish components
- collect and aggregate logging and monitoring information (state/status, moni parameters, . . .), events and alarms from dish components and report them to TM, after a proper mapping to a SKA common model
- perform diagnostic operations, fast control (<100 ms rate) and safety actions (autonomously or when commanded) on dish components
- perform life-cycle management operations (i.e. remote software/firmware update, power up/down)

ISBN 978-3-95450-148-9

- providing drill-down and tunnelling capabilities (i.e. remote access to devices or engineering interfaces, ...) to TM and operators

The M&C interfaces among dish sub-elements are represented in the diagram of Fig. 2. LMC performs M&C operations exclusively via the sub-element controllers, through a local Ethernet link, enabled by a network switch provided by the SKA Signal and Data Transport (SaDT) element, responsible also for the external connectivity of the antenna (e.g. transport of science and M&C data).

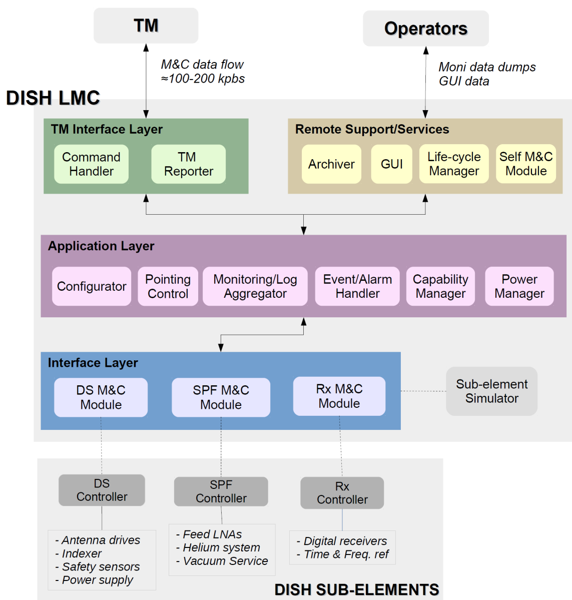

Figure 3: Logical view of Dish LMC software architecture arising from the preliminary design phase.

Preliminary Dish LMC Design

The high-level components of Dish LMC were identified during the preliminary design phase. A logical schema of the LMC architecture for SKA1-Mid antennas is reported in Fig. 3. The LMC software system can be regarded as a collection of modular packages, each providing a set of functionalities (as per SKA LMC requirements), organized in a three-level hierarchical structure. The "Interface" layer comprises a set of interface modules for the DS, SPF and Rx controllers, implementing communication access, control commands and retrieval of monitoring information. At this level a mapping from internal hardware state/status parameters to a global SKA control model (standardized for all SKA LMCs) is also performed. A suite of sub-element simulators is also planned to be developed for early testing of internal interface modules.

The "Application" layer implements built-in high-level functionalities and collective operations on sub-elements. For instance, configuration tasks, such as the configuration of the antenna for observation in a given frequency band or the setup of TM interface reporting, are performed by a *Configurator* component. Aggregation of all monitoring and log information collected from sub-elements (including pointing meta-data) and from LMC components (self M&C module) is carried out by a *Monitoring/Log Aggregator* component, which is responsible to maintain a rolled-up view of monitoring data inside the antenna and report it to TM. Similarly, events and alarms, particularly those related to safety or emergency situations (i.e. intrusion, emergency stop, ...), are handled by an *Event/Alarm Handler* component, which is responsible to execute recovery actions on failing components, filtering/reporting relevant events to TM (i.e. target lock, stow position, power-cut, ...).

Further, LMCs in SKA are expected to report a global capability information to TM, expressing the ability to deliver certain functionalities, i.e. for the dish the ability to operate the telescope in a band given the internal (health) status of all its constituents. Such functionality is provided by the *Capability Manager* component in the LMC architecture.

Pointing operations, performed by the *Pointing Control* component, requires support for receiving and executing pointing commands from TM interface, including the computing of local pointing correction to be applied before commanding the DS controller servo system.

On top of the hierarchy, a "TM Interface" layer is responsible to execute high-level commands from TM (defined in a standardized TM-LMC interface model), making use of service functionalities implemented in the middle layer. This layer was introduced since TM should not be aware of LMC internal implementation nor should access directly dish sub-element components. Moreover, it is desiderable to decouple as much as possible TM interface from LMC internal design, so that a change on the interface side does not severely affect LMC.

An archiving component, comprising a data archive plus archiving services, is also present. It is infact expected that LMC should not permanently store monitoring and log data locally. These data will be delivered to TM which will make them available globally. Only a limited-duration circular queue data archive, covering a 12 h time-window at maximum, is therefore foreseen and LMC shall provide services for creating, managing and downloading the archive.

Finally a set of high-level services are to be delivered to both LMC and engineering operators, such as a monitoring and control GUI, tunnelling services to access sub-element engineering GUIs and utilities to perform life-cycle operations (i.e. software/firmware update).

Implementation Guidelines

Object-oriented design will be extensively adopted for the implementation of the LMC control system, C++ and python being the likely adopted programming languages. LMC components are mostly network-distributed "online" components, designed to be configurable at run-time, exchanging data/events/logs each other over a software channel with different patterns, exposing commands callable by other components. "Offline" components, in turn, provide a utility or support function in the context of online components.

All the required communication and M&C features (alarm,

ISBN 978-3-95450-148-9

logging, archiving systems, GUI tools, error and exception handling, security, . . .) will be provided by a M&C framework and its underlying communication middleware. To minimize efforts and risks LMC will therefore rely on an existing off-the-shelf M&C framework rather than implementing base functionalities from scratch.

Key Technologies

A considerable effort is being made within the SKA Telescope Manager consortium to promote a standardization of adopted technologies, best practise, conventions (i.e. common model for definition of states and modes, . . .) and re-utilization of experiences and modules throughout the entire SKA control system. As a consequence the choice of a suitable COTS M&C framework, affecting the development of all SKA elements, resulted from a collaborative effort of representative members from all elements' LMCs and TM groups, as well as from the support of selected experts of the framework candidates and reviews of external accelerator/observatory facilities. A set of about 150 requirements (including performance requirements whenever available for LMCs) has been weighted against all the examined frameworks to drive the best choice for SKA needs, among which:

- Strong alignment with architectural concepts and required LMC functionalities;
- Reliability and scalability to SKA system size;
- Support for industrial standards and custom module development;
- Maturity and long-term support for both framework and middleware;
- Support for hierarchy creation among components, component configuration and command implementation;
- Availability of experiences both within LMCs and SKA precursors;
- Integration and re-use of precursors;
- User-base and strong community support

The following candidates were taken under consideration for evaluation against technical requirements: EPICS v3 [6] and v4 [7], TANGO Control System [8], Alma Common Software (ACS) [9], Meerkat CAM [10]. At last TANGO got the highest scores and was finally selected for prototyping stage after a careful evaluation of pros and cons considering also modern M&C trends (see for instance [11]).

The protocol for communication between LMC and subelement controllers is still to be defined. A natural choice would be extending TANGO layer down to low-level, but other alternatives for the middleware and data serialization have been evaluated against LMC requirements: ØMQ [12] with MessagePack [13] or Google Protocol Buffer [14] as serializer, ZeroC ICE [15], KaTCP [16]. Sample prototype modules have been developed and benchmark tests carried out on different network links. ØMQ and ICE received higher overall rates in the scoring process. KaTCP and ØMQ were found easier to be integrated with the TANGO layer. TANGO, ICE and ZeroMQ middlewares latency scaling curves were found comparable and allow to fullfil the most stringent performance requirement. KaTCP significantly deviated from the other middlewares, particularly above few MB transferred. The reason of such discrepancy requires further investigation.

Other technologies are currently under investigation, particularly those related to GUI development and continuous integration. Outcomes of the evaluation will be reported in a future project update.

SUMMARY

The SKA project has recently entered pre-construction phase II. LMC is expected to report a detailed design review of the dish control system. This stage will include definition of commands, monitoring points, state/status for each dish sub-element, consolidation of internal and external interfaces, refinement of the architecture and downselection of technologies to be employed. Development of some of the software components is also foreseen.

ACKNOWLEDGMENT

The author thanks the System Engineering (SE) team of Dish Consortium for their efforts and support in the dish interface modelling.

REFERENCES

[1] http://www.skatelescope.org

[2] "Science with the Square Kilometre Array", New Astronomy Reviews 48 (2004) 979-1606.

[3] A. McPherson, "Report and Options for Re-Baselining OF SKA-1", SKA-TEL-SKO-0000229 (2015).

[4] Y. Wadadekar et al., "Design Concept Description", WP2-005.065.020-TD-002 (2011).

[5] L.R. Brederode et al., "SKA Telescope Manager Project Status Report", MOA3O01, *These Proceedings*, ICALEPCS'15, Melbourne, Australia (2015).

[6] www.aps.anl.gov/epics

[7] http://epics-pvdata.sourceforge.net

[8] www.tango-controls.org

[9] www.eso.org/~almamgr/AlmaAcs

[10] L. van den Heever, "Meerkat Control and Monitoring - Design Concepts and Status", MOCOAAB06, ICALEPCS'13, San Francisco, CA, USA (2013).

[11] A. Götz et al, 'TANGO - CAN ZMQ REPLACE CORBA¿, TUCOCB07, ICALEPCS'13, San Francisco, CA, USA (2013).

[12] http://zeromq.org

[13] http://msgpack.org

[14] http://developers.google.com/protocol-buffers

[15] http://zeroc.com

[16] http://pythonhosted.org/katcp

ISBN 978-3-95450-148-9