Monitor and Control System: Reference Design Concept

020.50.25.00.00-0002-DSN-A-MONITOR_CONTROL_REF_DESIGN_CONCEPT

Status: **RELEASED**

PREPARED BY	ORGANIZATION	DATE
R. Hiriart	Computing & Software Div., NRAO	2019-07-19

APPROVALS		ORGANIZATION	DATE
R. Selina, Project Engineer	R. Selina 2019.07.29 15:49:01 -06'00'	Electronics Division, NRAO	2019-07-29
M. McKinnon, Project Director	Mark McKinnon Digitally signed by Mark McKinnon Date: 2019.07.29 16:21:36 -06'00'	Asst. Director, NM-Operations, NRAO	2019-07-29

RELEASED BY (Name and Signature)		ORGANIZATION	DATE
M. McKinnon, Project Director	Mark McKinnon Digitally signed by Mark McKinnon Date: 2019.07.29 16:21:52 -06'00'	Asst. Director, NM-Operations, NRAO	2019-07-29

Change Record

Version	Date	Author	Affected Section(s)	Reason
01	2018-05-30	Hiriart	All	Initial Draft.
02	2018-10-19	Hiriart	All	Update for internal review.
03	2019-07-19	Hiriart	All	Update for reference design release.
04	2019-07-23	Selina	All	Minor copyediting.
A	2019-07-29	Lear	All	Prepared PDF for signatures and release.

Table of Contents

1 Introduction

1.1 Purpose

This document describes general concepts and design choices for the ngVLA Monitor and Control (M&C) System as a preliminary step in deriving its system requirements and specifying detailed design.

1.2 Scope

The scope of this document is to analyze how ngVLA requirements shape the design of the M&C system, focusing on design aspects that affect the interfaces between the telescope electronic systems and computing. It considers in particular the operational needs for maintenance and troubleshooting.

The general architecture of the software and computing systems, including the main data flows involved in performing observations and computing derived data products, are described in [AD05], along with other M&C aspects. The supporting design of the electronic interface boards between the computing systems and the telescope electronics is described in [AD01].

2 Related Documents

Ref. No.	Document Title	Rev/Doc. No.
AD01	Monitor & Control System: Hardware Reference Design	020.30.45.00.00-0004-DSN
AD02	Operations Concept	020.10.05.00.00-0002-PLA
AD03	Preliminary System Requirements	020.10.15.10.00-0003-SPE
AD04	ngVLA: System Electromagnetic Compatibility and Radio Frequency Interference Mitigation Requirements	020.10.15.10.00-0002-REQ
AD05	Computing & Software: Reference Design Architecture	020.50.00.00.01-0002-REP

3 General Requirements

Besides the usual functionality expected from a modern control and monitoring system, the main requirement affecting this design is the budget constraint on telescope operations. This document analyzes how this requirement affects the relative priority of system attributes and what strategies can be introduced in the design to achieve the required performance levels. The system must achieve high reliability, maintainability, and usability to decrease operational and maintenance costs. Given the high number of antennas, scalability is also a concern.

Maintenance operations usually are either preventive or corrective. Preventive maintenance seeks to retain the system in an operational or available state by preventing failures. It affects reliability directly. By contrast, corrective maintenance includes all actions to return the system from a failed to an operating or available state. System reliability determines the load of corrective maintenance activities. Thus, the system should support and facilitate scheduling of optimal preventive maintenance activities to increase reliability and decrease the amount of required corrective maintenance and associated down time. At the same time, making the system more maintainable should decrease the cost of both preventive and corrective maintenance.

Array operations costs are correlated with the required number of operators. Their activities can be roughly classified as array coordination, observation execution, array supervision, and failure recovery. Costs can be decreased by automating these activities as much as possible and designing a highly usable system.

The array availability must also be considered. Availability considers both the probability of failures and the time that it takes to recover from them. Given the scale of the ngVLA telescope, there is a relatively high probability of devices failing, compared to currently operating telescopes. The system must be designed to tolerate failures during observations. After recovery, devices and antennas should be reintegrated into the array as efficiently as possible. This is facilitated by designing the system as a composition of decoupled components that can be operated autonomously with a minimum of centralized interactions.

In general, the ngVLA M&C concept is guided by two principles, derived from lessons learned from the EVLA and ALMA:

- The system should be based on autonomous and decoupled components controlling smart devices.
- The system should be organized and managed as a hierarchically connected system.

The first principle worked well for the EVLA, where the M&C system was structured by Module Interface Boards (MIBs) and MIB-like boards and computers which were highly decoupled, connected to the rest of the system by their Ethernet interfaces. They were designed as autonomous components. Once they were powered up, they were functionally operative and ready to receive command messages and send monitoring data. In case of failures, a MIB and its controlled electronics can be powered down, recovered or replaced, and powered up again without affecting the rest of the system.

ALMA was structured as a hierarchy of control components, forming a Distributed Control System (DCS). In general, this architecture handles complexity well, is scalable, improves reliability, and is well suited for a geographically dispersed system. A DCS allows one to distribute the system load on multiple machines and enables the installation of redundancy strategies. It also helps to scale networking loads, as higher level components can receive commands of a higher level of abstraction, which are translated to multiple commands directed to the lower level components. Monitoring can be scaled similarly.

Integral in this architecture was the use of a database to manage the current and past system configurations, tracking which hardware devices (identified by S/N) were installed in the system at any given time. Tracking this information is fundamental for the application of automated diagnostics and preventive maintenance algorithms. It is also necessary in order to develop tools that facilitate the task of gathering all the necessary information needed to effectively troubleshoot problems.

4 Control Architecture

Implementing the principles from Section 3, the ngVLA control system will be structured in five layers, as shown in Figure 1. The bottom Hardware Device layer represents the hardware devices that compose the telescope. The electronics devices will be packaged as Line Replaceable Units, identified by a unique serial number.

The Hardware Controller Layer corresponds to controller boards (analogous to MIB boards in the VLA, or AMBSI boards for ALMA), which provide a standardized Ethernet interface to its connected hardware devices. They translate Ethernet messages to the low level interfaces used by hardware devices: SPI, I2C, and GPIO. This layer also includes the CSP Local Control System and other central electronic systems (e.g., local oscillator and timing). In this case, the Hardware Controller will not necessarily be Controller boards, but they could consist of computers that implement the same interface.

The Ethernet messages received by the Controller boards can be command messages (usually referred as SET messages), or monitoring messages (GET messages) and they need to specify the target device and the specific value inside the device that is being modified or requested (the command or monitor *point*). For SET messages, the command can optionally carry an application timestamp in the future, specifying when the command should be applied. If not present, the command should be applied as soon as possible. The controller sends a response message in return for both the GET and SET messages. It responds with the value and read timestamp for GET messages, and with the application timestamp and a status code for SET messages.

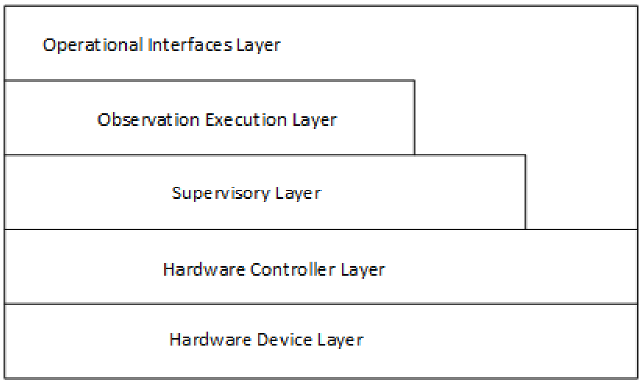

Figure 1 - Control architecture layers. The boundary between the Computing and Software IPT and other hardware IPTs has been defined as the interface between the Supervisory Layer and the H/W Controller Layer.

Each LRU is controlled by a single Controller board, which can be queried for the corresponding serial numbers. The system automatically discovers the serial numbers of each LRU and keeps track of their corresponding type and the system slots where they have been installed. This is necessary in order to associate data streams with specific hardware devices. The Supervisor should be able to detect when an LRU has been replaced and reconfigure itself, detecting and propagating the new serial numbers.

The Supervisory Layer provides higher-level system functions, integrating one or more controller boards. For example, the Antenna Supervisor would accept a high-level command to tune the frontend, which could then be translated into several commands sent to the controller boards that are involved in this operation. The Supervisory Layer incorporates logic to react to events detected in the lower layers, and supports maintenance operations without requiring interactions with a centralized control. The Supervisory Layer supports both reliability, by detecting and reacting to faults before they become failures, and maintainability, by providing smart interfaces for error reporting, diagnostics, and maintenance operations.

Each LRU should be autonomous and come up in an operational state after power up. The initial initialization routine will be executed by the Controller boards, and will include the connection to the network. Each LRU has a defined type that identifies its function in the system, and a role that identifies where it is installed in the system. As an example, each antenna has several IRD modules, each one connected to receive different polarizations and frequency bands.

As soon as the LRU reaches the operational state, it will send a multicast message containing identifying information such as its serial number(s), type, role, and status. This message will be received by the Supervisor, which will configure itself accordingly. The initialization routine should also include a built-in diagnostic, which could also be invoked on demand.

The Observation Execution layer orchestrates the execution of astronomical observations, following the operations defined in the telescope observing modes. This is the layer that supports the allocation of sets of antennas into sub-arrays and implements the required observing modes.

The Operational Interfaces Layer incorporates user interfaces in the operator consoles. The components belonging to this layer interact not only with the Observation Execution layer, but with the Supervisory and Hardware Controller layers as well. The ability to bypass layers is important to support effective troubleshooting. Usually, the lower layers are accessed by means of console applications (a.k.a. administrative or service ports).

Regarding the allocation of real-time requirements, these are divided in hard real-time requirements and soft real-time requirements, the distinction being how critical it is if a task misses its defined deadlines. Any deadline that cannot be missed without placing humans and/or equipment in danger should be regarded as a hard deadline. The allocation of real time requirements in either the Supervisory Layer or the Hardware Controller Layer depends on the scope of the operations involved. If a real-time operation involves more than one device, then its deadline constraints should be imposed in the Supervisory Layer. Otherwise, they should be allocated in the respective Controllers and devices.

So far, no real-time operation has been recognized that would imply real-time requirements in the Supervisory Layer. At the present stage of the project, it may be too early to make a strong decision on this area, but if this assumption holds, it can be stated that the Supervisory and above layers will deal only with soft deadlines, where the outcome of missing them will mostly result in intervals of flagged science data, and hard real-time requirements will be implemented in the Hardware Controller Layer and below. This is consistent with the general principle of constructing the system with smart devices.

Another aspect that will be affected by real-time requirements is the protocol between the Supervisory Layer and the Hardware Controller Layer. Currently, because of its ubiquity in the industry, it seems clear that an Ethernet-based solution is highly preferred. However, it is important to recognize that Ethernet is not a real-time control communication protocol, and it is affected by unpredictable latencies.

Several modifications of the Ethernet protocol have been proposed by the control system industry to address this issue. These are defined in the IEC 61784 standard, organized in profiles depending on their performance indicators (which include delivery time, throughput, and time synchronization accuracy). These protocols modify the protocol stack at different levels, and are classified as “Top of TCP/IP”, “Top of Ethernet”, or “Modified Ethernet”, depending on where they introduce modifications. Examples are Modbus/TCP, Ethernet/IP, TCNet, Profinet CBA, EtherCAT, and Profinet IO.

The decision as to whether these modified protocols are required is tied to the decision of requiring real-time capabilities in the Supervisory Layer. This decision is deferred to the Conceptual Design phase of the project, as it depends on detailed information about the hardware interfaces.

If normal, unmodified Ethernet can be used in the Supervisory/Controller interface, and an interesting possibility would be to select protocols such as HTTP/REST, UPnP, CoAP, MQTT and XMPP for control, monitoring, and discovery. GraphQL, an optimization of REST, is also gaining good traction. These protocols have become popular choices on constructing the "Internet of Things." There are clear advantages on selecting well-known and used protocols: the existence of good quality implementations and a wide pool of available developers. The task of securing the interface is also facilitated, as security is a core requirement for the Internet of Things. As they have been designed to work on public networks, they could be a good choice for controlling and monitoring the long haul antenna stations.

Regarding safety requirements, LRUs (which are composed of Hardware Devices and their Controllers) should be designed so they deal with any safety critical condition on their own, without requiring the participation of higher-level functions in the monitoring and control system. Otherwise, LRUs are no longer autonomous and pose an operational risk.

5 Monitoring Architecture

The system will periodically sample the current values of a collection of attributes for each device, and will transmit them through a monitoring bus to the subscribed clients. In general, the source of monitoring data will be the hardware controllers, and clients will be distributed in all upper layers in the system. The monitoring bus incorporates publish/subscribe interfaces and allows the definition of multiple channels, allowing the receivers to select the monitoring data of interest, and control filtering options.

Although monitoring data originating from controllers will normally be listened to by clients in the supervisory and upper layers, the system does permit controllers to receive data from other controllers, if necessary.

Higher-level components in the control hierarchy could accumulate, translate, or summarize the monitoring data received from its lower-level components. For example, the antenna supervisory component could send an antenna status event, which would incorporate the status of the antenna devices whose malfunction would trigger data flagging commands. This translation of monitoring events allows one to distribute the processing of the monitoring event to the components that are better capable of interpreting them.

The system should provide interfaces to modify dynamically the parameters that control the capture and processing of monitoring data (the monitoring sampling frequency, the offset and slope to translate from hardware units to system units, etc.). Upon startup, their default values will be loaded from files stored in flash memory in the controllers, but they may be overridden by the supervisor with values read from the Telescope Configuration database. All the configurations defined in the Telescope Configuration database will be kept under version control.

The sampling period for specific monitor points could be shortened for a period of time when a given error or warning condition is detected, with the intention of collecting additional data to aid in debugging. This type of automatic collection of debugging information would be triggered by the Supervisor. This would be similar to an "oscilloscope function" with an automatic trigger. Given the amount of data that this function can potentially generate, the high-rate data will not be sent through the network in the normal monitoring stream, but will be stored in local files in the Supervisor.

The Supervisor will be the place where functions to aid in troubleshooting operations will be implemented. This part of the system will need to be constructed with modifiability in mind, as it is usually the case that the incorporation of logic to debug urgent problems needs to be incorporated as soon as possible. One strategy to achieve the required modifiability could be to introduce this logic at configuration, and not during implementation. This can be achieved by the use of scripted logic, which would be executed when a certain configured condition is detected. These scripts would conform to a safe API, which expose only the functions necessary for this troubleshooting operations.

Given the number or antennas, devices, and monitoring points that the system is required to monitor, scalability is a concern. Scalability needs to be considered both for the data acquisition, which ends with the monitoring data saved in a database, and for the use of this database, which should respond to queries in a reasonable amount of time. The monitoring system can be scaled horizontally by deploying multiple data collection processes, and the application of clustering at the database level.

6 Error Handling

The system must detect and react adequately upon the occurrence of error conditions. The error handling functions will incorporate strategies to improve the system availability, constraining the impact of faults to be as limited as possible. Error handling also plays an important role in improving maintainability, by collecting error information, processing it, and presenting it with the goal of facilitating the task of troubleshooting problems as much as possible.

The system will incorporate one or more messaging channels to propagate detected errors. At the lowest level, error conditions will be detected in the Controller layer. These will be received by their Supervisor, which will apply rules to interpret the received messages. The Supervisor will take reactive actions and will propagate the messages to the upper layers. The messages sent by the Supervisor can aggregate several received messages, perform error analysis, and add additional information. The structure of the error messages will be designed to preserve the original messages while adding information. This is necessary in order to prevent cascading error messages, which can make the identification of the root problem by the operator more difficult.

This pattern of reception, interpretation, encapsulation, and forwarding will be repeated in each layer of the Monitor & Control architecture as error events propagate upwards. Ideally, a single error message should appear in the operator console, clearly specifying the problem and suggesting corrective actions.

In practice, the logic of the error handling system will be based initially on artifacts such as Root Cause Analysis, Fault Mode and Effect Analysis, and other system engineering specifications. As engineering and operations teams develop an understanding of the system, the error handling system will need to incorporate this knowledge in a timely manner. Therefore, it is important to design this part of the system in such a way that it can be easily modified and extended by different groups. After some time, maintenance teams will understand the reasons behind system failures, and operators will become experts on how to recover the system. It is important to define ways for this knowledge to be captured and integrated.

After errors are analyzed by the M&C hierarchy, they will be consumed by a centralized, system-wide error analyzer. This system will apply more extensive algorithms, which can involve not only the instantaneous information collected during the occurrence of the problem, but also data from the monitoring database. The tasks of analyzing the occurrence of an error condition, deducing the root of the problem, and suggesting the course of action is closely related to the task of analyzing monitoring data to predict possible problems, and suggest preventive maintenance actions.

Those two functions could be allocated in the same component, or two different deployments of the same component. This approach is known as Condition-Based Maintenance, which manages maintenance actions based on the condition or state of the elements of a system, applying algorithms for diagnostics (to assess the current condition) and prognostics (to predict the future condition) to increase the effectiveness and decrease the cost of maintenance operations.

7 Time Synchronization

Both the application of commands at the correct time and the timestamping of monitoring data depend on the Supervisory and Controller computers to be synchronized in time. The time synchronization mechanism designed into the system will depend on the precision required in these operations, which are derived from the telescope system requirements.

There are several alternatives: The EVLA relies on the internal clocks of the MIBs and single-board computers, and uses the Network Time Protocol to keep them synchronized. ALMA, on the other hand, locks the real time machines to a timing event coming from a Maser, and sets the absolute time from a GPS. These different schemes (and possibly others) will need to be analyzed and compared against the timing requirements.

For now, the working assumption is that following a synchronization mechanism similar to the EVLA should suffice.

8 Security

The M&C security design will be based on system policies that are yet to be specified. These policies should identify possible vulnerabilities, their probability of occurrence, their impact and the cost of the associated countermeasures. A wide range of possible internal and external threats could affect the M&C system, e.g., intercepting control and monitoring command streams, hacking into the supervisor and controller machines, issuing commands to intentionally corrupt operational databases, etc. The appropriate security level for different parts of the system must be specified before introducing protective measures into the design.

Regarding the antenna deployment, the components belonging to the Supervisor, Controller, and Device layers will be located inside an antenna, while the upper layers will be deployed in the Central Electronics Building. Logical access to antenna networks will be limited via private addressing and filtered to include a designated set of control points. Connections from control points will require an authentication handshake and should support tiered authorization levels for monitor versus control commands.

Monitor & Control communication channels from the Central Building to the Antennas can be uniquely encrypted based on authorization if required. It is less clear whether the communications inside an antenna should be encrypted as well, as accessing these channels would require breaking into the antenna and presumably an alarm system should alert for the intrusion and trigger protective measures. At this time, there is no requirement to encrypt the individual antenna data streams.

Any subsystem which could pose risks to the antenna or human life will have a hardware failsafe to protect against malicious commands.

9 Development and Simulation

The general functionality of an M&C system has much in common with automatic control systems in other industries (e.g., energy, gas, oil, and manufacturing). The project will analyze the suitability of incorporating a third-party framework.

For example, SKA recently selected Tango Controls (http://www.tango-controls.org). ALMA developed the ALMA Common Software (ACS, http://www.eso.org/projects/alma/develop/acs) system, and other telescopes and research facilities have incorporated the Experimental Physics and Industrial Control System (EPICS, https://epics.anl.gov). The use of these frameworks, along with code-generation strategies (it is usually the case that a set of statements must be repeated for each control and monitoring point) can significantly reduce software implementation and maintenance costs.

The project will also develop simulation components early in the development phase in order to facilitate testing and integration activities. Simulation and mocking components will be needed at different levels to test the integration between software components and individual electronic devices, and to test system-level functions.

10 Appendix

10.1 Abbreviations and Acronyms

Acronym	Description
ACS	ALMA Common Software
ALMA	Atacama Large Millimeter-submillimeter Array
CSP	Central Signal Processor
DBE	Digital Back End
DCS	Distributed Control System
DTS	Data Transmission System
EPICS	Experimental Physics and Industrial Control System
EVLA	Expanded Very Large Array (Jansky Very Large Array)
FE	Front End System
GPIO	General Purpose Input Output
GPS	Global Positioning System
IF	Intermediate Frequency
IRD	Integrated Downconverter/Digitizer Module
LO	Local Oscillator
LRU	Line Replaceable Unit
M&C	Monitor and Control
MIB	Module Interface Board
MS	Measurement Set
ngVLA	Next Generation VLA
OPT	Observation Preparation Tool
PBT	Proposal Builder Tool
PHT	Proposal Handling Tool
PST	Proposal Submission Tool
SKA	Square Kilometre Array
TAC	Telescope Allocation Committee
UI	User Interface
VLA	Jansky Very Large Array
WVR	Water Vapor Radiometer