

EVLA Monitor and Control System

Hichem Ben Frej, Barry Clark, Wayne Koski, Rich Moeser,
George Peck, James Robnett, Bruce Rowen, Kevin Ryan,
Bill Sahr, Ken Sowinski, Pete Whiteis

Revision History

Date	Description
2001-Jul-16	Initial release
2001-Sep-25	Revision of text
2001-Oct-16	Minor cleanup & revisions
2001-Nov-20	Links to requirements documents added
2002-May31	Major revision. Entire chapter reorganized & updated
2003-Aug-22	Major revision. All sections reviewed, some deleted, new sections added.
2003-Sep-24	Minor revisions based on feedback from the EVLA Advisory Committee
2004-Dec-06	Revisions based on developments over the past year
2006-Apr-19	Update

Table of Contents

EVLA Monitor and Control System.....	1
10.1 Introduction.....	1
10.2 EVLA Monitor and Control Software	2
10.3 Antenna Monitor & Control Subsystem (AMCS).....	4
10.4 Correlator Monitor and Control Subsystem (CMCS).....	13
10.5 Operational Interface Subsystem (OIS).....	15
10.6 EVLA Monitor and Control Network (MCN).....	22
10.7 Transition Planning.....	25

10.1 Introduction

This chapter of the Project Book has been organized by subsystems, with allowances for topics that do not fit within that scheme. Four subsystems have been identified – Antenna Monitor and Control (AMCS), Correlator Monitor and Control (CMCS), the Operational Interface System (OIS), and the EVLA Monitor and Control Network (MCN). The sections covering the four subsystems all have a similar structure:

- 10.a Subsystem Name
 - 10.a.b Subsystem General Description
 - 10.a.c Subsystem Requirements
 - 10.a.c.1 Subsystem Hardware Requirements
 - 10.a.c.2 Subsystem Software Requirements
 - 10.a.d Subsystem Design
 - 10.a.d.1 Subsystem Hardware Design
 - 10.a.d.2 Subsystem Software Design

10.2 EVLA Monitor and Control Software

In June of 2004 an initial high level design for all EVLA Software was completed. The results of this effort are presented in the document entitled “EVLA High Level Software Design”, which can be found on the Computing Working Documents web page, <http://www.aoc.nrao.edu/evla/techdocs/computer/workdocs/index.shtml>. It is document # 33. The diagram given below shows the components of the EVLA software system and the major data flows, as identified by that high-level design document. A follow-up effort to add detail and depth to the high level design is now underway.

That portion of the software identified as being in the “Online Domain” are the components of the software that fall within the EVLA Monitor and Control system, i.e., from some point within the Observation Scheduler into the Data Capture and Format component.

10.2.1 EVLA Software Requirements

The scientific community within NRAO and Array Operations at NRAO has produced a series of requirements documents that represent significant input from the users of the EVLA. The document set is located on the Computing Working Documents web page (<http://www.aoc.nrao.edu/evla/techdocs/computer/workdocs/index.shtml>), and consists of the following:

- EVLA-SW-001, EVLA e2e Science Software Requirements, April 15, 2003 (#26)
- EVLA-SW-002, EVLA Data Post-Processing Software Requirements, (Draft), July 3, 2003 (#28)
- EVLA-SW-003, EVLA Array Operations Software Requirements, June 6, 2003 (#27)
- EVLA-SW-004, EVLA Engineering Software Requirements, August 8, 2003 (#29)
- EVLA-SW-005, EVLA Science Requirements for the Real-Time Software, November 22, 2004 (#38)

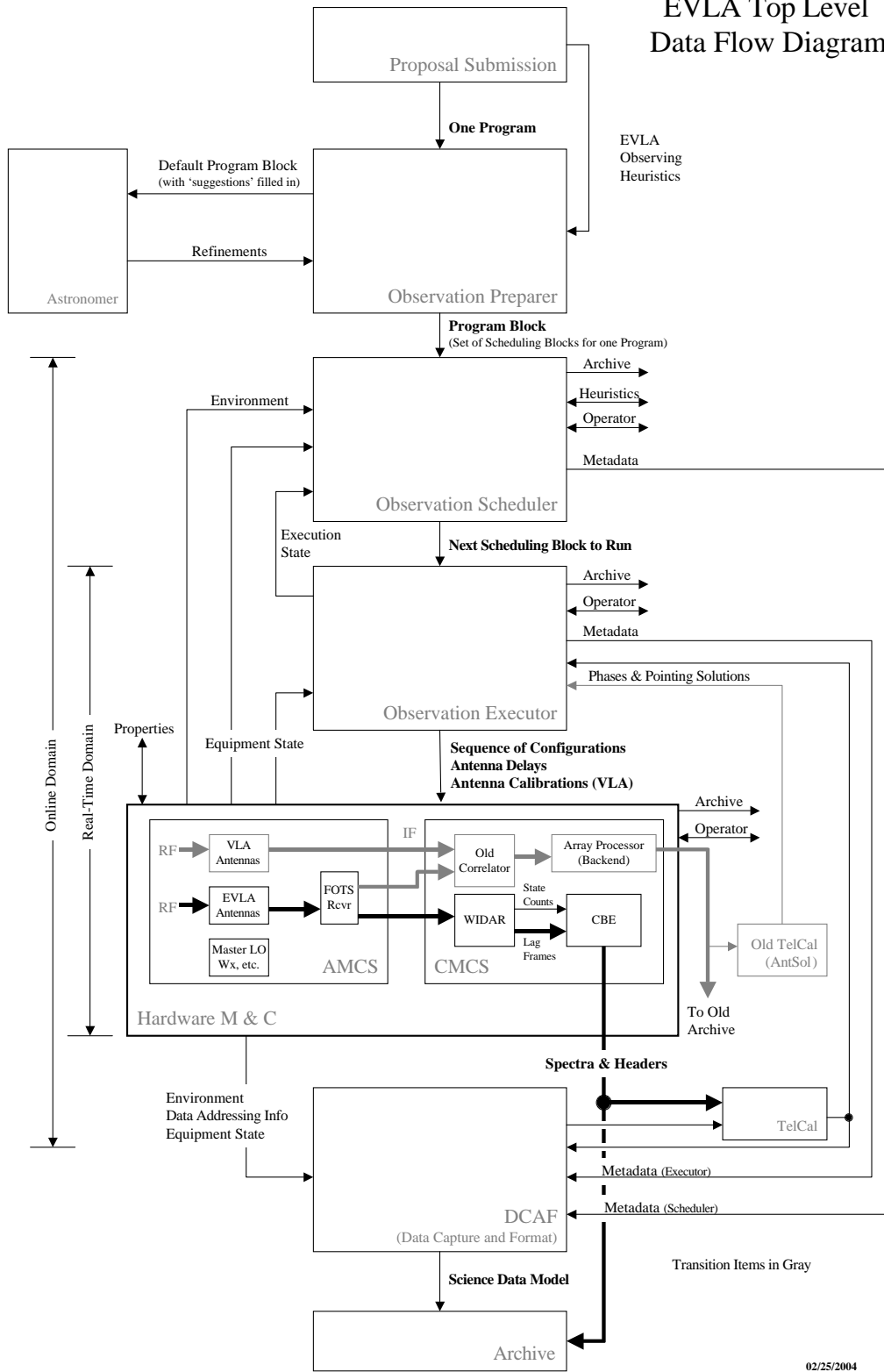
10.2.2 Operations Security

The VLA must be protected 1) from both random and malicious incursions from the Internet outside the NRAO, and 2) from accidental or mistimed interference from within the NRAO. The primary protection from the outside world is via the network routers. The machines of the EVLA reside on a private network that is not routed beyond the NRAO. Within the NRAO (or within the Virtual Private Network that is effectively within NRAO), the usual password protections apply. We shall trust password protection to keep out malicious incursions. Only a modest addition need be provided to guard against accidental interference by technicians with modules actually being used for the current observation. It is planned to implement this safeguard at the module interface board (MIB, see section 10.3.1) level. The observing system will be granted sole command access to those MIBs that it sees as necessary for operation. Only the array operator will be able to release a module for outside command

10.2.3 Software Standards

Software in the MIBs will be written in C. This will be cross-compiled on Windows systems system under the control of the Maven compilation control system. The central Executor system will be written in Java, also compiled under the Maven compilation control system, and documented through the javadoc documentation extraction system. Appropriate documentation should always be included in the code, and indentation should be used to make code maximally readable. Nightly builds of the entire system are a standard practice.

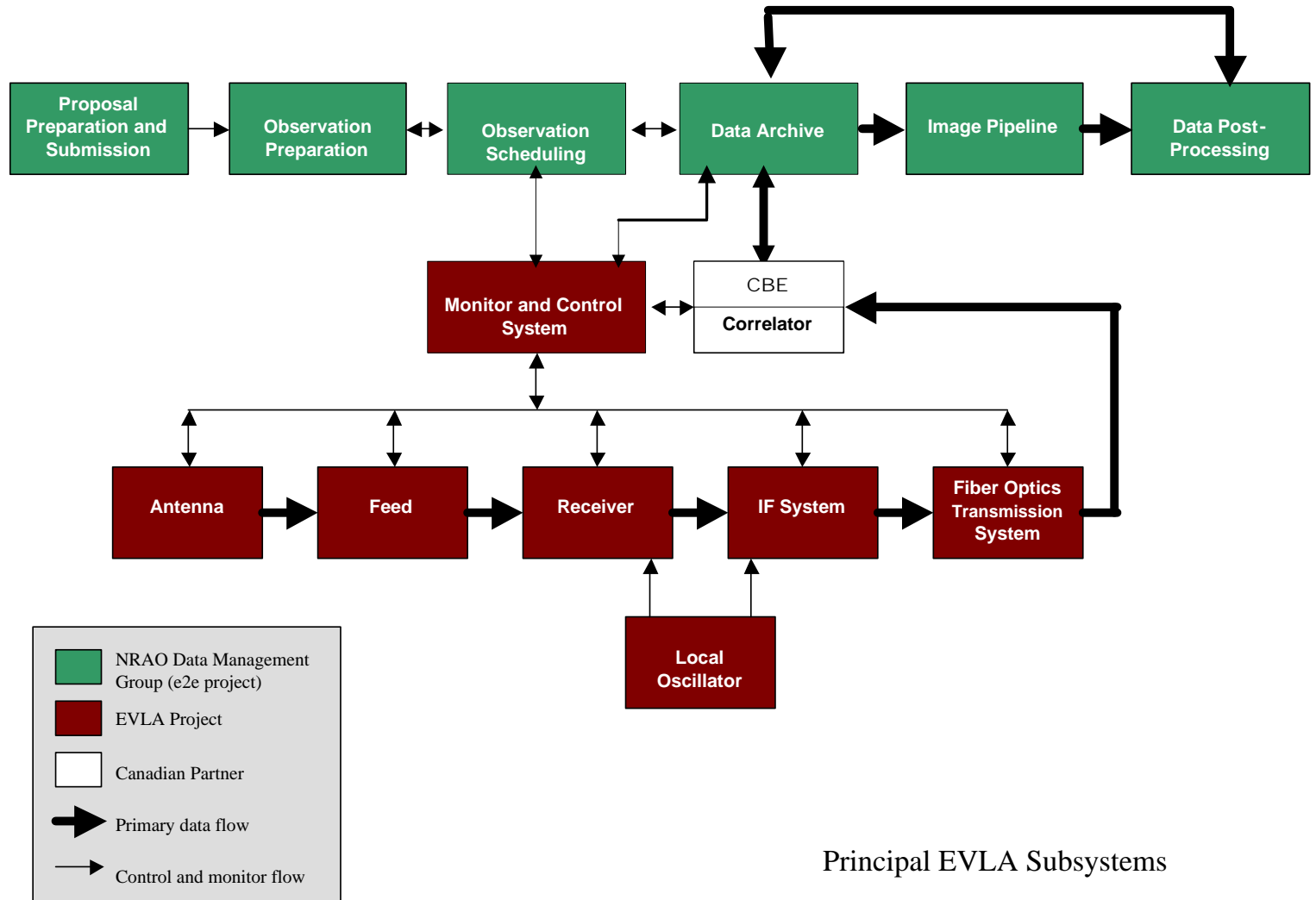
EVLA Top Level Data Flow Diagram



02/25/2004
MRSW
Rev. 6/11/04

10.3 Antenna Monitor & Control Subsystem (AMCS)

The Antenna Monitor and Control Subsystem is that portion of the EVLA Monitor and Control System responsible for operating the array of antennas, both the new EVLA antennas as they come online and the existing VLA antennas during the transition phase.



Principal EVLA Subsystems

10.3.1 AMCS General Description

The Antenna Monitor and Control Subsystem will consist of processors located throughout the system from the Control Building at the VLA site to within the EVLA antennas themselves. Processors residing in the Control Building will have no limits on size and complexity and will take the form of high reliability desktop and rackmount machines. Processors located within the antennas will be small micro-controller type processor boards with minimal RAM and low clock speeds to help reduce RFI. These micro-controllers will interface the components of the antenna to the rest of the monitor and control system; they are referred to as Module Interface Boards (MIBs).

All AMCS processors, including the MIBs within the antennas will be networked using Ethernet over fiber-optic cable.

10.3.2 AMCS Requirements

10.3.2.1 AMCS Hardware Requirements

10.3.2.1.1 Minimum RFI

The most basic AMCS hardware requirement is low emission of RFI. Minimum emission of RFI is necessary in order to prevent the scientific data from being corrupted by noise from the AMCS. Emissions from all components must meet the requirements specified in EVLA Memo #46, "RFI Emission Goals for EVLA Electronics", Rick Perley, 10/15/02, which can be found at <http://www.aoc.nrao.edu/evla/memolist.shtml>.

10.3.2.1.2 Ethernet

The use of Ethernet as the bus is actually an implementation decision. As such it is not a requirement, but this decision has such a widespread impact on such fundamental levels that it has been included in this section. The use of Ethernet allows the entire AMCS to use one bus, to use COTS equipment, guarantees maintainability over a long timespan due to widespread commercial use, allows addressing by slot, and is well suited for object-oriented programming.

10.3.2.1.3 Data Rates

The maximum data rate from an EVLA module is estimated to be 128 Kbits/sec. The majority of the EVLA modules will have a data rate very much less than 128 Kbits/sec. Overall the maximum data rate from an EVLA antenna is expected to be 200 Kbits/sec. It is possible that most of the monitor data from an antenna will be from a single module, where the total power detectors are located.

10.3.2.1.4 Timing

Reconfiguration commands sent by the ACMS must begin not more than 100 μ s after the intended implementation time. This requirement will necessitate the queuing of commands at the MIB before the scheduled implementation time.

The monitor and control system must be able to keep absolute time to a resolution of better than 10 ms.

10.3.2.1.5 MIB (Module Interface Board) Requirements

The design of the MIB was largely driven by three main requirements. Additional important design requirements are also necessary for the MIB to perform its duties in a robust manner.

The three main requirements were 1) the choice of TCP/UDP/IP over Ethernet as the communication protocols, 2) low RFI characteristics, and 3) small board size. The RFI emissions requirement limited the choice for the microprocessor and on-board electronics. The small board size is especially important, given space limitations, which are also affected by the low RFI emission requirement.

Additionally, the MIB must utilize both serial and parallel communications to the EVLA devices. It must have the intelligence to implement monitor and control tasks in the modules and devices, and must sometimes be part of control loops. Periodically or on demand, the MIB must be able to send back monitor data to other points on the network. It must be possible to remotely load new software into the MIB. In order to synchronize commands and send monitor data on a periodic basis, the MIB must be able to receive a timing signal and keep track of time. A watchdog timer must be implemented so that the MIB will recover if the processor hangs up. The MIB must be fused. A separate maintenance port that communicates via RS-232 must be included. There must be power indicator LEDs to indicate the presence of each voltage. A test LED must be included to facilitate programming and debugging.

10.3.2.2 AMCS Software Requirements

The current version of the Antenna Monitor and Control Subsystem Requirements document can be found at <http://www.aoc.nrao.edu/evla/techdocs/computer/workdocs/index.shtml>, under the title “Antenna Monitor & Control Subsystem Preliminary Requirements Specification, V2.0” (document # 17).

This specification was written to address the technical requirements of the AMCS needed for a design that will satisfy the user-related requirements described in section 10.2.2.

While the requirements specification document referenced above contains a detailed description of all of the requirements imposed on the AMCS, it is worth mentioning here the few major requirements that have the most influence in ‘shaping’ the AMCS software design.

- **Heterogeneous Array.** The EVLA will, from the onset, consist of different types of antennas. During the transition phase this will be the older VLA types as well as the new EVLA types. Eventually VLBA and New Mexico Array (NMA) antennas may be added as well. Because of this, the design of the AMCS must accommodate differences in antenna hardware.
- **Ethernet Based Communications.** The EVLA will be a highly networked system; even the antenna subcomponents will utilize Ethernet with each subcomponent having its own IP address. Ethernet and the associated network communications protocols (IP/TCP/UDP) will require that the AMCS design accommodate this higher level of data communications between the various components of the system.
- **Widespread Operational Interface.** The EVLA will be operated (at various levels) from a potential variety of sources: normal programmed observing from the e2e system, Interactive Observing, control from the AOC and other NRAO entities, subcomponent operation from the technician’s workbench and even monitoring from over the Internet at large. The AMCS design must serve a variety of users from a variety of physical locations.
- **Transition Phase Operation.** The transition from the current VLA antennas into the EVLA antennas will take a number of years to complete. The AMCS must be designed so that during this time 1) both antenna types will be operated together under one control system, 2) system down time is minimized and 3) transition specific software (throw-away) code is minimized.
- **Real-Time Requirements.** There are few hard real time requirements imposed on the AMCS but those that do exist most certainly must be accounted for in the AMCS software design. They are:
 - 100 μ Sec command start latency. This means that a command must be initiated at the hardware within 100 μ Sec of its scheduled start time.
 - For EVLA antennas, to maintain a sub-arcsecond level of pointing accuracy, the antenna position must be updated every 50 milliseconds.
 - Frequency change within band to be completed within one second.
 - ‘Nodding’ source switch rate of once per ten seconds.
 - For the most extreme case of OTF mosaicing, the correlator will require new delay polynomials at the rate of 10 per second.
 - The issue of the rate of pointing updates required by the most extreme case of OTF mosaicing has been raised (antenna movement at 10X the sidereal rate). The pointing update rate is set by the servo bandwidth. It does not depend on the rate of motion. For this case, the pointing rate update will be the same as delay polynomial update rate for the correlator – 10Hz.

10.3.3 AMCS Design

The AMCS is being designed to be state-of-the-art with respect to today's technology, and to be in a position to utilize new technologies as they arise. To help achieve these goals, the choice of the foundation level technologies is biased as much as possible to non-vendor specific, pervasive, industry-wide standards. For hardware this means computers that will use industry interfaces such as IP/Ethernet for their external communications and, for embedded hardware controllers, standard signal protocols such as SPI and standard mechanical form factor and connectors*. For software it means using commonly available and widely used operating systems and communications protocols. These choices will allow components of the system to be replaced as technology changes with minimal effect on interoperability with other system components. The ultimate goal of this approach is to create a system that 20 years from now will not be locked into 20-year old technology.

*The MIB does use IP/Ethernet and a SPI interface to hardware but it uses a proprietary form factor and hardware interface connectors.

10.3.3.1 AMCS Hardware

10.3.3.1.1 General Description

The hardware portion of the AMCS will consist of a MIB (Module Interface Board) and various other boards to interface the MIB to devices.

10.3.3.1.2 Module Interface Board (MIB)

Every EVLA module or device will contain a Module Interface Board (MIB). The MIB is the interface between the antenna control computer and any module or device electronics in the EVLA. Command and Monitor information will be sent between the MIBs and network computers over 100BaseFX full duplex Ethernet. Communications between the MIB and EVLA devices are primarily carried out via Serial Peripheral Interface (SPI) and General Purpose I/O (GPIO) lines (parallel communications).

The core of the MIB is the Infineon TC11IB microprocessor. This chip incorporates several peripheral functions that often require separate chips. These include the Ethernet controller, 1.5 Megabytes of on-chip memory, a SPI port, and two serial ports. On-chip timers satisfy the timing requirements of the MIB. The TC11IB requires a 12 MHz crystal oscillator that is multiplied, on chip, to create a 96 MHz system clock. It has a watchdog timer to ensure that the program does not hang.

A 64 or 128 Megabit Flash memory chip is used to store the program image(s) for the TC11IB. The MIB can run from the Flash memory, however it is planned that the program will be transferred from Flash to memory on the TC11IB during the boot sequence. It has been shown that RFI is minimized by running from on-chip memory. It will be possible to load new program code into the Flash memory from a network computer via Ethernet. The on-chip memory will also be used to store commands and monitor requests from the antenna computer.

Access is provided to one of the General Purpose Timer Units on the TC11IB chip via connections to 2 timer unit outputs and 2 timer unit inputs. The General Purpose Timer Unit could provide a pulse or a clock to the EVLA module that the MIB is controlling.

The voltage regulator chip on the board includes power management features that reset the TC11IB if the voltages fall below their nominal values. During the power-up sequence, the reset line of this chip keeps the TC11IB in the reset

state until all voltages have risen to the correct values. All power supply voltages on the MIB, both input and output, are fused for protection against shorts.

The MIB will receive a 19.2 Hz system heartbeat for timing purposes. A computer on the network will be able to tell the MIB to start keeping absolute time at the arrival of the next system heartbeat. A timer on the TC11IB will then be used to keep time. This will make it possible to queue commands in advance, to be executed by the MIB at a specified time.

The MIB will detect the slot into which it is plugged. This feature eliminates the need to change the module address when the module is moved.

The MIB will not be used to ensure the safety of any EVLA modules or devices. Each module or device must be designed such that it will be protected even in the absence of a MIB.

10.3.3.1.3 Battery Backed Utility

EVLA modules at the antenna and control building, that are powered from the system 48 volt supply, will remain powered for a specified amount of time in the event of a commercial power outage. The specified amount of time will be long enough for the generators to start operating and restore power. In the event that the generators do not start operating, there will be plenty of time for computers in the control building to determine the state of each antenna before the UPS units in the EVLA antennas lose power.

10.3.3.1.4 Voice Communications

Voice communications between an antenna and the outside world will be enabled via VoIP (Voice over IP). The system will carry standard telephone voice communications for an antenna over the Monitor and Control Network link in TCP/IP form. It is predicted that the voice traffic will not hinder antenna control traffic. A spare fiber pair is available if such turns out not to be the case.

The VLA and AOC phone systems are maintained by New Mexico Tech (NMT). Implementing full voice communications between an EVLA antenna and legacy phone systems will require cooperation with New Mexico Tech. NMT is currently implementing a VoIP system. We (NRAO) will be able to tap directly into this system, relieving us of both the need to purchase transition hardware and the need to manage the system. Currently only 100Base-T (twisted pair) IP phones are available. Phones will be connected via a media converter to the antenna switch until fiber based phones are available.

10.3.3.2 AMCS Software

Where feasible, AMCS software is being designed to use open-source Linux operating systems, Web-based tools, and standard Internet protocols.

Two exceptions exist: the MIB and the low-level, real-time processor in the Control and Monitor Processor (the CMP). (The CMP provides the EVLA M&C interface to VLA antennas). Both of these processors are resource limited and operate in a hard realtime environment, making the use of small footprint, realtime OS kernels the preferred choice.

For all but these two exceptions, software will be designed using Object Oriented techniques and written in the Java programming language

10.3.3.2.1 AMCS Software, General Description

This section describes the AMCS software in general terms with loose reference to the major requirements that the design seeks to satisfy.

Ethernet Communications. With no exceptions, all of the processors in the system incorporate operating systems capable of providing TCP and UDP over IP over Ethernet. Data communications between the processors will be non-proprietary using industry standards such as XML, URIs, and HTTP.

Heterogeneous Hardware. Operating different types of hardware can present difficulties if the controlling process needs to be cognizant of the differences and operate accordingly. These difficulties are lessened in the AMCS by using a distributed processing approach in its design. Implementation differences are encapsulated at appropriate levels throughout the system so that client applications can send commands and receive monitor data of a more generic nature. Using this approach, ‘what-to-do’ rather than ‘how-to-do-it’ type information is communicated from client to server.

Variety of Users from a Variety of Locations. AMCS components are designed to be functionally autonomous so that each component can be operated separate from the rest of the system. Components operate together to form the overall functionality of the system for normal observing but each can be operated independently for maintenance and other special uses.

All components are accessible over the Internet so they can be operated from any Internet connected client application within policy restrictions.

Real-Time Considerations. The nature of the communications infrastructure of the AMCS precludes real-time control of hardware from higher-level processes. Rather than attempting real-time function calls between processes over the network, the AMCS is designed so that control instructions are sent ahead of the time of their execution. Once present at the target process, the instructions can be executed at the appropriate time in real-time.

In an attempt to simplify design for application engineers, Remote Procedural Call (RPC) type middleware such as CORBA and Java’s RMI attempt to make remote function calls appear as if they are being done locally. In so doing they tend to hide the fallibilities of the network over which they communicate giving a false sense of security that functions and responses are being handled in real-time. The communications mechanisms chosen by EVLA engineers expose the network and its characteristics, such as latency, to the design engineers so they may be dealt with explicitly.

10.3.3.2.2 AMCS Communications Infrastructure

The AMCS uses a client/server architectural style. Servers represent system entities that are to be controlled and monitored (a piece of hardware, a software function, a property file, etc.); their ‘service’ is to provide a representation of the entity for manipulation by client applications. Client applications communicate with the servers to ‘operate’ the system.

Different technologies were investigated for the middleware to be used for communications between the client and server components. The chosen technology would have to operate under the constraints imposed by the requirements described above; additionally, it would be desirable if communications could be consistent among all components of the system including the embedded processes in the MIBs and CMP.

Remote procedural call (RPC) type middleware such as CORBA and SOAP were, in general, ruled out as candidate technologies. Instead, a message passing communications architecture that utilized information formatted as XML

over UDP and HTTP was favored. Lower level entities, such as the MIBs, use UDP while higher level entities like the Executor communicate using HTTP.

A report giving the analysis and logic supporting the decision to use a message-based approach can be found on the EVLA Computing Working Documents web page -

<http://www.aoc.nrao.edu/evla/techdocs/computer/workdocs/index.shtml>.

It is document #39, "EVLA Monitor and Control Communications Infrastructure", Version 1.0.2, Bill Sahr, 02/16/2005.

10.3.3.2.3 AMCS Application Software

Some of the key AMCS software components have been developed and deployed. This section gives a brief overview of selected components, describing their purpose and current state.

10.3.3.2.3.1 Executor Software

The Executor is the process responsible for conducting an observation. It is the high level real-time control element. It receives instructions from control scripts and translates those instructions into commands that are sent to the antennas under its control. The scripts are written in Jython, which is very similar to Python but includes extensions that allow it to interact with and create Java objects. The scripts can be created manually or auto-generated by the program *obs2script*, which converts a VLA Observe file into an EVLA control script. Below is an example of a simple EVLA control script that configures antennas for C band.

```
(Cband.py)

myband = LoIfSetup('6GHz') # default continuum setup for C band
mysource = Source("16h42m58.81", "39d48'36.9")

subarray.setLoIfSetup(myband)
subarray.setSource(mysource)
subarray.execute(array.time() + 2 / 86400.0)

subarray.execute(array.time() + 60 / 86400.0)
array.wait(array.time() + 60 / 86400.0)
```

The Executor has two execution modes. The first mode, used entirely for testing, requires a list of arguments defining the path to the script and the antennas to be controlled. In this mode the Executor will run the specified script and upon completion it will exit. The second mode is the normal operating mode. In this mode the Executor is run as a Java servlet within a web-server/servlet container, communicating with user interfaces using HTTP. In this mode, the Executor does not terminate after running a script. It is always available, running scripts as requested.

When a script is submitted to the Executor, it is automatically added to the runtime queue. If all of the specified antennas are available, the script runs until it completes or is aborted. If any of the needed antennas are in use, the script remains in the queue until all antennas are available at which time it is placed into run mode.

The Executor is responsible for the division and management of subarrays. A subarray is typically defined when a new script is submitted, but can be adjusted at runtime by adding or removing antennas using the operators user-interface software.

10.3.3.2.3.2 Alert Handling Software

Alerts are generated by software processes and most often represent monitor points outside of normal operating range; for example an alert would be sent when the antenna ACU approaches a critical left or right cable wrap limit in azimuth. The majority of the alerts originate from the MIBs, however, any software capable of sensing critical or unusual states can report an alert condition.

All monitor points have an alert property that represents the monitor point's alert state. If a monitor point is in alert, its alert property will have a value of 1, otherwise, the alert property will be 0. Alerts are multicast. Any client interested in receiving alerts simply subscribes the alert multicast group. When a software process identifies an alert condition it will create a UDP datagram containing information on the source of the alert (e.g., device id and monitor point id) as well as the alert condition and send it to the alert multicast group. The information contained in the packet is formatted as XML that conforms to a published XML schema.

The EVLA system has a software process called the Alert Server that collects and manages alerts. The Alert Server is written in Java as a Web application and is deployed to a servlet container (Apache Tomcat). HTTP queries can be sent to the Alert Server to get a list of all active alerts or an historical list of alerts. In order to receive the alerts, the Alert Server subscribes to the alert multicast group and then goes into receive mode, processing the alerts as they arrive. JAXB (Java API for XML Binding) is used to convert the XML into Java objects which are then inspected to determine whether a new monitor point is in alert or if an existing monitor point has been cleared. The alerts can be viewed using a standard Web browser or the Java-based operator user-interface.

10.3.3.2.3.3 Monitor Data Archive Software

The monitor data archive software, *monarch*, is responsible for receiving monitor and control point data and loading it into an Oracle database. A Web-based interface is available for data retrieval. The primary sources of monitor point data are 1) MIBs - in antennas, on the bench, in the test rack, and 2) VLA antenna data sent from the CMP. As with alerts, the archive data is multicast as XML encoded UDP datagrams. Any interested task can receive the archive monitor data stream by subscribing to the appropriate multicast group. As of Spring 2006, roughly 9 million rows of data are being archived daily.

10.3.3.2.3.4 MIB Software

The MIB software is implemented as a generic framework common to all MIBs plus module specific software. Descriptions of the MIBs module hardware are abstracted to 1) a C language table that defines hardware access parameters, and 2) an XML point configuration file that defines the user view of the hardware. The XML point configuration file can be downloaded to the MIBs flash memory at runtime, and will be activated when the MIB reboots. A detailed description of the MIB Framework can be found on the Computing Working Documents web page, <http://www.aoc.nrao.edu/evla/techdocs/computer/workdocs/index.shtml>. It is document #36, "MIB Framework Software, Version 1.1.0", Pete Whiteis, 10/08/2004.

The MIB monitors hardware in a periodic polling loop, converts the raw data to engineering units, performs bounds checking (alert detection) and writes the resultant data into a memory resident data base known as 'Logical Points'. All of the data contained in the 'Logical Points' database can be accessed though a command line interface. The ASCII command interface to the MIB, known as the Service Port, can be accessed over TCP/Telnet, or by UDP. This interface allows one to query the list of available hardware devices and access detailed information on one or more monitor and/or control points. The two primary commands implemented by the Service Port are 'get' and 'set', which are used to read or write data to one or more Monitor or Control points. A detailed specification of the Service

Port can be found in the document entitled ‘MIB Service Port ICD’, Version 1.2.0 on the Computing Working Documents web page <http://www.aoc.nrao.edu/evla/techdocs/computer/workdocs/index.shtml>, as document #34.

Additionally, data is ‘pushed’ from the MIB over a one-way data channel known as the Data Port. The Data Port is used to multicast monitor data and alerts to any interested clients. Typically monitor data is multicast at a periodic interval and alert messages are multicast when they occur. The protocol specification for the Data Port is contained in the document “MIB Data Port ICD, Version 1.2.0”, which can be found on the Computing Working Documents web page <http://www.aoc.nrao.edu/evla/techdocs/computer/workdocs/index.shtml>, as document #35.

The MIB application is time aware and uses a Modified Julian Date format for time stamping monitor data as well as time-deferred command execution. The MIB application initializes its time management facility by way of an NTP server and will scale this time to the nearest 52 milliseconds Waveguide Cycle, if a 19.2Hz interrupt is present.

The MIB application has the capability of remotely loading code and data over the Ethernet. Code images must be loaded in Motorola S-Record format and data must be formatted as XML Points Configuration files. Additionally, code and data loading progress can be monitored by use of a MIB monitor point.

The remote location of most MIBs (in the antennas) mandates a high degree of reliability and some degree of a self-healing capability in the event of a software failure. The use of a hardware watchdog timer, implemented on the TC111B microprocessor, helps meet these requirements. A software task must service the watchdog timer within a 5 second period in order to keep the MIB software in an operational state. If MIB the software falls into a hard loop or a software trap, the watchdog timer circuit will reboot the MIB. A watchdog reboot will generate an alert which is multicast to higher level client software. Additionally, the MIB framework software maintains a count of watchdog timer reboots that is available via the MIB service port.

10.3.3.2.3.5 CMP Software

The CMP is a transition system which exposes VLA antennas to the EVLA Monitor and Control System using the same interface as has been implemented for EVLA antennas. It is a real-time system used in conjunction with the Serial Line Controller (SLC) to manage VLA antennas until conversion of all antennas to EVLA electronics has been completed. The CMP is located in the control building at the VLA site, in close proximity to the SLC.

The CMP implements the MIB interface for VLA antennas. The only difference is that the MIB exposes data on a module-by-module basis, while the CMP exposes data grouped as an antenna. In all other respects – command syntax, the multicast of alerts, the multicast of monitor data, etc, CMP-mediated VLA antennas, from the perspective of software interactions, behave as EVLA antennas.

The CMP implements the same hardware monitoring mechanism as the MIB. The actual data is collected from the SLC. It also implements the MIB Service Port and the MIB Data Port. Even though the CMP is a single device, it uses multi-threading and IP aliasing to present a virtual collection of VLA antennas, each appearing to be associated with a separate software interface and a separate IP address, allowing commands to VLA antennas to be sent on an antenna-by antenna basis.

The CMP is time aware and uses a Modified Julian date format for time stamping monitor data and for time-deferred command execution.

The CMP consists of 2 VME processor boards – a high level interface board, running a real-time Linux, that implements the MIB interface, and a low level board, running VxWorks, that handles the hard real-time task such as interrupt processing, time keeping, monitor data collection, command processing, and command execution in sync with the waveguide cycle and the SLC. The 2 boards communicate via shared memory.

10.4 Correlator Monitor and Control Subsystem (CMCS)

10.4.1 CMCS General Description

The CMCS will provide Correlator monitor and control through a network of distributed processors and processes. General access to this network will be through a “Virtual Correlator Interface” (VCI) that will provide a unified and flexible means of integrating the Correlator into the overall EVLA system.

Details of the VCI and scheduling may be found in document # A25201N0000, “EVLA Monitor and Control Virtual Correlator Interface”, NRC-EVLA memo #16, “Scheduling and Activating the Configuration Data”, and NRC-EVLA memo #18 “MCCC Software – Generating Baseline Board Configuration ...”, all by Sonja Vrcic of DRAO. Document A25201N0000 may be found on the Software Documents web page of the DRAO web site:

<http://www.drao-ofr.hia-ihp.nrc-cnrc.gc.ca/science/widar/private/Software.html>

NRC-EVLA memos #16 & #18 are located on the Memos web page of the DRAO web site:

<http://www.drao-ofr.hia-ihp.nrc-cnrc.gc.ca/science/widar/private/Memos.html>

A suite of graphical user interfaces are integrated into the correlator modules to provide engineering level access and control during initial system development and deployment. It is the intent that these interfaces remain functional after correlator commissioning.

10.4.2 CMCS Requirements

A more detailed description of CMCS requirements may be found in EVLA Computing memo # 13, “EVLA Correlator M&C Software Requirements & Design Concepts”, Brent Carlson, 1/23/2002. This document can be found on the VLA Expansion Project Computing Working Documents web page:

<http://www.aoc.nrao.edu/evla/techdocs/computer/workdocs/index.shtml>

Additional information on the Correlator software architecture, CMCS, and other topics can be found on the Software Documents web page of the DRAO web site:

<http://www.drao-ofr.hia-ihp.nrc-cnrc.gc.ca/science/widar/private/Software.html>

10.4.2.1 CMCS Hardware Requirements

The CMCS shall consist of a network of distributed processors with each processor responsible for managing a single circuit board unit of the Correlator hardware. These Correlator module interface boards (CMIBs) will provide the hard real time control necessary for Correlator operation. There shall be one master Correlator control computer (MCCC) to coordinate and manage the distributed network, host the operational VCI gateway, and centralize Correlator system management. This computer shall be considered a high reliability platform and shall be made fault tolerant through the use of hot standby or other methods to maximize system up time. A separate and similarly equipped computer will manage power monitor and control for the Correlator (CPCC) and will operate independently of the MCCC thereby isolating power control from any faults in the MCCC. The CMIB hardware modules should be

of an electrical and mechanical form factor that lends itself to mounting on Correlator hardware devices in a replaceable and standardized fashion.

The CMIB design shall allow for future upgrade with minimal impact on the remaining installed systems. Modules shall provide sufficient memory and other resources to operate within a boot-from-network computing environment.

The modules shall provide a standardized method of communication with the Correlator monitor and control network and Correlator hardware. Correlator hardware shall be capable of being powered up and initialized into a quiescent state without any external network connections.

Unlike the processor chosen for the AMCS MIBs, the selection of a processor for use as the CMIB is not constrained by RFI considerations. The CMIBs will be located on the Correlator boards, inside a heavily RFI-shielded room.

10.4.2.2 CMCS Software Requirements

The Operating systems used for the MCCC, CPCC, and CMIBs shall provide reliable and maintainable operation for the expected life of the Correlator. CMIB operating systems and run time software shall be capable of responding to a 10 ms interrupt period, provide low level access for Correlator hardware communications, and provide reliable networking with the MCCC. The MCCC operating system and run time software shall provide a reliable and easily managed environment with easy integration into the EVLA MC network. It shall perform predictably under various network loads and fault conditions without operator intervention.

10.4.3 CMCS Design

The CMCS will make extensive use of hardware abstraction such that each functional unit of the correlator will be represented as a black box to higher layers of control software. The details of switch settings, data paths, hardware organization, etc. will be hidden except where this knowledge is needed by higher processes and when accessed through various utility methods. Each CMIB will present a unified interface to its methods and control points such that upper level software is decoupled from any changes in CMIB design.

10.4.3.1 CMCS Hardware

PC-104+ mechanical form factor computer boards will be used for the CMIB hardware. This industry standard lends itself well to creating a piggyback style module for mounting on the Correlator hardware boards.

Communication between the CMIB and Correlator hardware will be over the PC104+ bus (PCI standard). This bus will allow the CMIB to download FPGA personalities and control words to the Correlator hardware as well as extract monitor and ancillary data from the Correlator. The MCCC and CPCC will most likely be high reliability PCs or VME/CPCI type SBCs with sufficient I/O connectivity to communicate with the Correlator MC network. The network itself will be based on 10/100 Base-T Ethernet using transformer coupled copper connections to reduce potential ground loop problems. The CMIBs, MCCC, and CPCC will need to support communication over this medium and protocol. It is anticipated that much of the MC network between the MCCC and CMIBs (around 300 units) will be routed through switches and hubs to reduce the port requirements on the MCCC. Further details of the topology and networking may be seen in the previously referenced EVLA Computing memo #13.

10.4.3.2 CMCS Software

Due to the need for flexibility and portability on the network side MC code, selection of an OS for the MCCC and CPCC should place connectivity high on the requirements list. Since these computers will not be overly constrained by memory or CPU speeds, many commercial and public OS choices exist. Current versions of the preempt-able

Linux kernel seem capable of providing for the realtime and networking needs of the system. It is expected to divide all run time code into logical processes/threads and assign priorities to best utilize system resources and network bandwidth. Watchdog processes will be used to monitor MC system health and take corrective action when possible.

10.5 Operational Interface Subsystem (OIS)

The Operational Interface Subsystem is one of several major components that constitute the EVLA Monitor and Control System. The primary responsibility of OIS is to provide a suite of software tools and screens that allow the array operators, as well as engineers, technicians, scientists, software developers, and other authorized users to interact with the array in a safe and reliable manner.

10.5.1 OIS General Description

The Operational Interface Subsystem provides the primary graphical user interface (GUI) tools for the EVLA Monitor and Control System. It is through the OIS that users, especially operators, will monitor and control the array on a daily basis.

This section will discuss the various components of the OIS, the functions it must provide, its users, the implementation, current state, and deployment of the OIS.

10.5.1.1 Components

- **EVLA Resources.** EVLA resources are components that reside within the EVLA Monitor and Control System. They represent both physical (antennas, weather station) and non-physical (subarray) entities. Each resource will be network addressable and have a common software interface that accepts requests for information and commands.
- **Client-side tools.** OIS is a client-side tool that will allow access to the EVLA system. Parts of OIS will be written as standard Java applications. Other portions will be written as Web-based applications accessible from standard Web browsers. OIS will be accessible from the VLA Control Building, the AOC or any Web-accessible location. Outside access to the EVLA control system must conform to network security guidelines described in the section entitled Monitor and Control Network (MCN) Access.

10.5.1.2 Functions

- **Array Monitoring.** The Operational Interface Subsystem will supply the array operators and other users with high-level and low-level monitoring abilities. High-level screens will provide information on the overall health of the array whereas the low-level screens will give detailed information on specific components within the system. The screens will be composed of textual and graphical components and will use color and audible alerts to inform the user of unexpected events and conditions.
- **Array Control.** Many of the OIS screens will allow authorized users to control all or parts of the array. Control functionality will be built into the screens using graphical user interface (GUI) components (sliders, buttons, combo boxes, etc.) that accept keyboard or mouse input from the user.
- **Reporting/Logging.** The ability to create and manage a variety of reports, including the operator notes pertaining to particular observations (observing log). The Operational Interface will provide a tool that enables authorized users to create and send messages to a message log, presumably a database. This will replace and

expand on the functions currently provided by the observing log that is generated by the array operators using Microsoft Excel.

- **User Access Control.** The ability to view and manage users' system access and privileges. This is required for security purposes.
- **System Management.** The ability to manage system files and parameters. The Operational Interface will provide a means for operators to update system parameters, such as pointing, delays, baselines, and to maintain a history of parameter changes.

10.5.1.3 Users

- **Array Operators.** The array operators are responsible for the overall success and safety of all observations and will be the primary users of the Operational Interface Subsystem software. They require the ability to both monitor and control the array and perform their duties from either the VLA Control Building or the AOC.
- **Engineers.** Engineers are responsible for the design, development and testing of the mechanical and electrical components within the system. They require the ability to inspect/control individual system components both remotely and at the antenna during working and non-working hours.
- **Technicians.** Technicians are responsible for the day-to-day monitoring and maintenance of the mechanical and electrical components within the system and are usually the first to be notified in the event of a non-working or malfunctioning component. As with the engineers, technicians require the ability to inspect/control individual system components both remotely and at the antenna during working and non-working hours.
- **Scientists.** Scientists, both NRAO and non-NRAO, are granted time on the array to conduct scientific investigations or tests. Their primary interest lies in the scientific data obtained by the instrument. They require remote access to both monitor data and visibility data to assess progress and to help make decisions during an observation.
- **Programmers.** Programmers are responsible for creating the software that drives the system. They must have access (with control capabilities) to the system, both locally and remotely, for testing and troubleshooting during working and non-working hours.
- **Others.** Access to the system, most likely via a Web browser with read-only access, will be provided to individuals that have an interest in the array and its activities.

10.5.2 OIS Requirements

This section highlights the major requirements of OIS. A detailed description of OIS requirements can be found at <http://www.aoc.nrao.edu/evla/techdocs/computer/workdocs/index.shtml> in the document titled "EVLA Array Operations Software Requirements" (document #27).

10.5.2.1 OIS Hardware Requirements

OIS will not communicate directly with any hardware. It will, however, communicate directly with the software interfaces for specific pieces of hardware (e.g., an Antenna Object) that will in turn execute the request on behalf of OIS.

- **Supported Platforms.** The OIS software must be relatively platform independent, as it will run on a wide variety of machines hosting various operating systems. Specifically, the OIS software must be capable of running on commodity PCs hosting Windows and Linux operating systems and Sun Microsystems workstations hosting the Solaris Operating Environment. An optionally supported platform will be the Macintosh/Mac OS.

10.5.2.2 OIS Software Requirements

The software requirements document referenced above contains a detailed description of requirements imposed on OIS; it is worth mentioning here a few major requirements that have the most influence on the design of the OIS software.

- **Remote Observing.** Remote observing will provide users with the ability to run the OIS software from locations other than the VLA control building such as the AOC, other NRAO sites or from any Web-accessible location.

Several reasons exist as to why remote observing is necessary:

- Observers can monitor the progress of their observing program and make or request changes during their observation to increase the quality of data.
 - Engineers, technicians and programmers will need the ability to access the system from remote locations during working and non-working hours to do first-order problem solving.
 - Operators may be stationed at the AOC in Socorro in the future.
- **Secure.** The Operational Interface Subsystem will need a robust security mechanism in place so that unauthorized users are not allowed access to parts of the system that may compromise the success of an observation, cause damage to an antenna or jeopardize the safety of personnel in or around an antenna.

A coarse-grained security mechanism is under consideration that separates users into one of two groups: trusted or non-trusted. Trusted users will have privileged access to the system, namely control capabilities, whereas the non-trusted users will have only monitoring capabilities. Membership in the trusted group will likely be a function of identity and location. Users who would otherwise be trusted may be treated as non-trusted if they are located outside the NRAO or are not part of the NRAO VPN.

- **Easy to Obtain, Install and Update.** Since the OIS software will be geographically dispersed, a simple procedure must exist that allows users to obtain and install the software via the Internet. Due to its simplicity and minimal user interaction, Java Web Start is being used for deployment.
- **Easy to Use.** A feature often overlooked in the design of software for the scientific community is ease-of-use. A goal of the EVLA project is to have graphical user interface tools that are easy to use and intuitive. Besides being intuitive the GUIs must also adhere to a specified set of user interface design guidelines to create consistent interfaces and behavior across the various tools. Software that is easy to use is also often easy to learn which could reduce the three months it currently takes to train an array operator.
- **Robust.** The system must be capable of surviving failures within the system. It should not be affected by network glitches, broken socket connections, or the resetting or rebooting of devices within the system. In the event of such failures, OIS should warn the user that a failure has occurred, and it should continue working without incident. For example, loss of communication with one antenna should not affect the acquisition of data from other antennas. And when the antenna is functioning and back online, the system should automatically resume data acquisition as if nothing happened.

10.5.3 OIS Design

The design goal of the Operational Interface Subsystem is to meet the requirements stated in the “EVLA Array Operations Software Requirements” document. At the same time the system must be designed so that parts of the system can be replaced with newer technologies. “Designing for the future” will allow new technologies, both hardware and software, to be “plugged-in” to the system for a gradual upgrade process rather than waiting for the next VLA expansion project.

10.5.3.1 OIS Hardware

OIS will not communicate directly with the hardware. The only hardware design constraint is that it be relatively platform independent so it can run on many types of computers with little or no changes. This has little impact on the design and more impact on the selection of the implementation language.

10.5.3.2 OIS Software

10.5.3.2.1 General Description

The design of the Operational Interface Subsystem and the EVLA Monitor and Control System as a whole should exhibit the following general characteristics:

Loosely Coupled. Loosely coupled implies that components within the system are not tightly joined at the hip, but instead communicate via a coarse-grained interface that rarely changes. The primary benefit of loose coupling is that changes to one subsystem or subsystem component will not require changes to the subsystem that uses the changed component.

Highly Adaptive. The EVLA as a physical system will change not only through the transition phase and EVLA phase II, but also on a daily basis. During the transition phase VLA antennas will be upgraded to EVLA antennas and eventually NMA antennas will be added to the system. The system should easily adapt to these long-term changes without incident and without specialized code. It should also adapt to short-term changes such as the addition of new modules or monitor points.

Discovery-Based. A discovery-based system allows objects (e.g., subarrays, antennas, antenna subsystems, etc.) to be located at runtime rather than referring to a hard-coded list of known modules. In such a system the client can dynamically locate and manipulate any component within the system as the system dynamically changes beneath it. The more the client can find out about the system at runtime, the more flexible and extensible the system.

Extensible. An extensible system allows new features to be added to the system. The system should be designed so that these new features can be “plugged-in” at a later date with little or no impact on the overall system. Some examples of extensible features are a screen builder that allows users to create their own screens and a system simulator that could be used to test software or train operators.

Scalable. The physical elements of the EVLA will change over time. The number of antennas will increase and hence the number of antenna subsystems. As with most systems, the addition of new elements, in this case antennas, could possibly lead to degradation in performance. The system must be designed such that the addition of new antennas has minimal impact on the overall performance of the system. Likewise, as the number of users increases the overall performance of the system should not degrade.

Lightweight Client. In order to achieve loose coupling, OIS must have little or no knowledge of the implementation of EVLA components. OIS should only be concerned with the presentation of information and the sending and receiving of messages from other subsystems. The less OIS knows about the business logic, the less likely changes to the core system will affect OIS and the more detached the client software can be from the core components.

10.5.4 Implementation and Current State

The OIS software has been in regular use by array operators and engineers since the latter part of 2004. It first appeared as a simple tool that enabled communication to any EVLA device. Since that time it has evolved into a feature-rich tool that provides enough information and functionality to control and monitor the EVLA antennas. There are essentially two forms of the software; a rich-client Java application that is highly interactive and visual, and a Web application that is less dynamic, has minimal graphics, but is highly accessible. Most of the development effort has been focused on the Java application since it will be used by the operators.

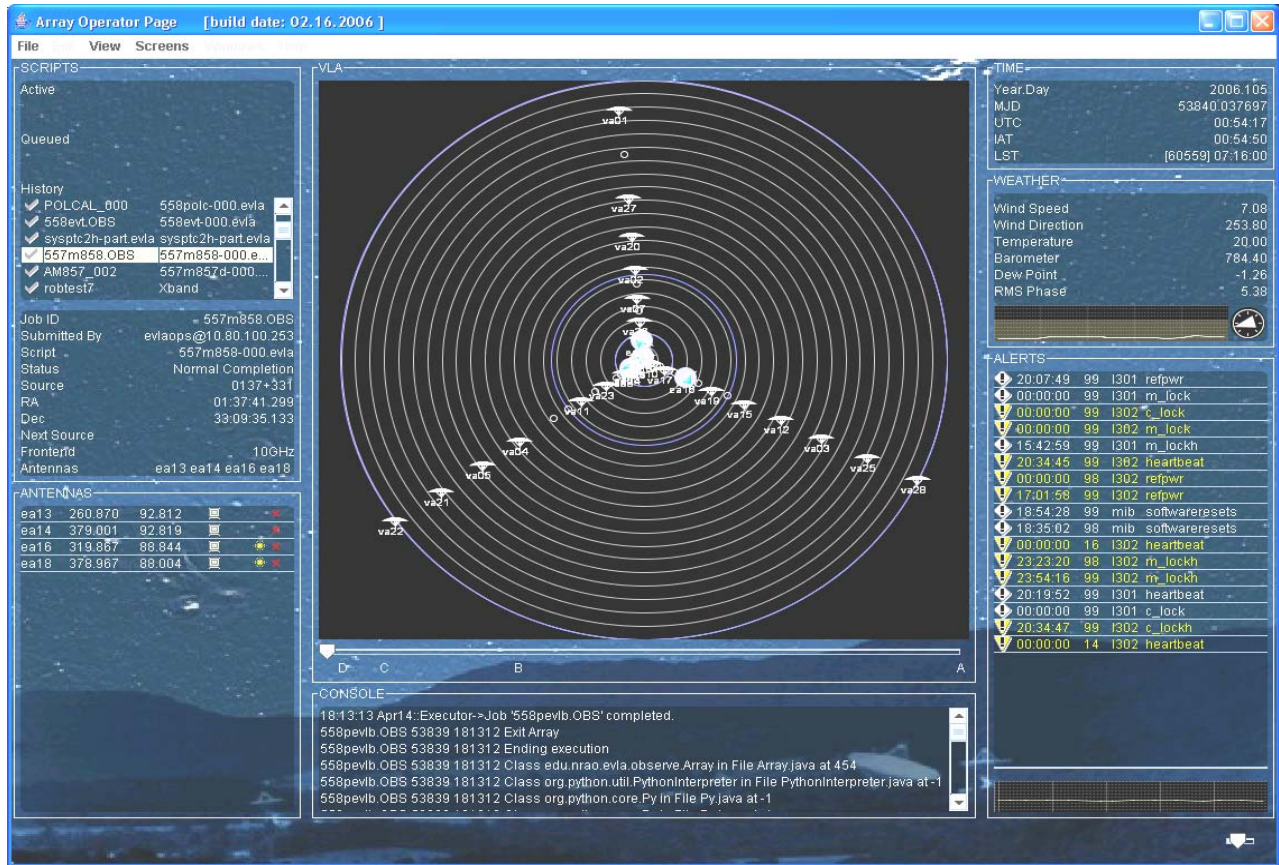


Figure 1 Screenshot of the Array Operators Page

10.5.4.1 Java Application

This version of the software is written entirely in Java and requires the host computer to have a recent version of the Java Virtual Machine, currently Java 1.5.x. The software is written using the many GUI components included in the standard Java download as well as many open source components, such as JFreeChart, a library of plotting and graphing components. The application is deployed using Java Web Start.

The software is used daily by operators, and to date the following capabilities exist:

- **Script Submission.** Users can select a script from the filesystem and provide a list of antennas for the script to control. At any time, the user can abort the script. During execution of the script the user may remove an antenna or insert an antenna into the subarray.
- **Script Monitoring.** The progress of a script can be monitored from a subarray screen and the script console. The subarray screen displays information related to the progress of the script; source being observed, the right ascension and declination, and the antennas being used. The console screen displays text output written by the Executor running the script and messages generated by the script.
- **Weather Data and Plots.** The weather screen displays important weather data; wind speed, wind direction, temperature, barometric pressure, dew point and RMS. There is also a time-history plot of the wind speed and a meter showing the current wind direction.
- **Drill-Down for Detail.** Many of the components provide a drill-down feature. Double-clicking on monitor point values or antenna icons will trigger a screen showing detailed information for the target component. Double-clicking on an antenna icon, for example, will launch the ACU screen shown below.

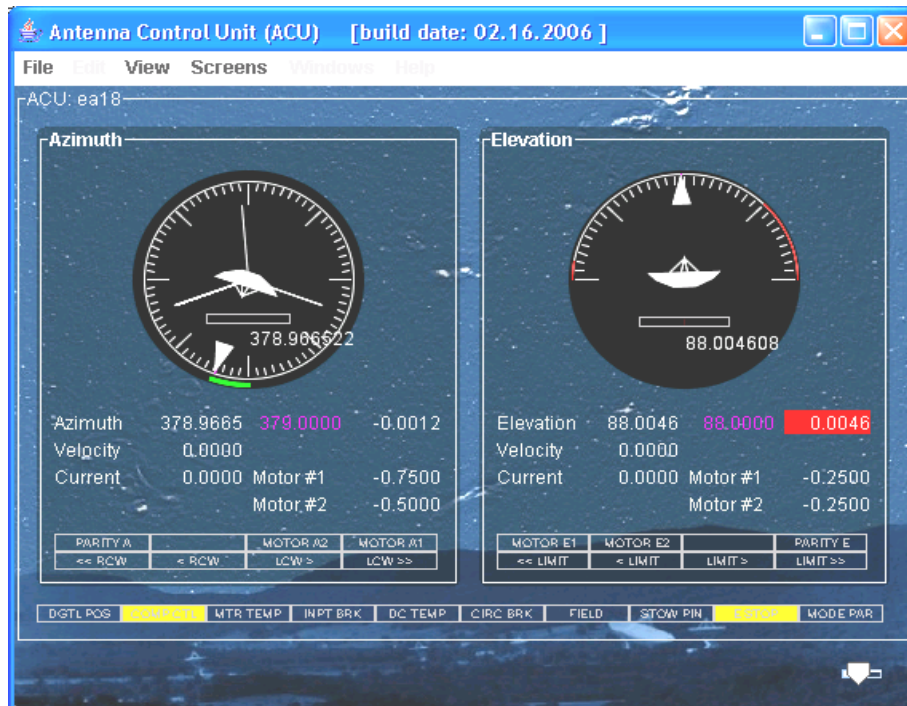


Figure 2 Screenshot of ACU screen

- **Alert Monitoring.** Active alerts are viewed using the alert screen. The alert screen displays alerts in time order with the newest alert appearing at the top of the list. Users can select the alert to get detailed information on the alert. The user also has the option of removing a selected alert.
- **Bird's Eye View of the Array.** This is a top-level view of the array. This component shows the current locations (pad) of all of the antennas in the array as well as their current azimuth positions. Placing the mouse pointer over an antenna triggers a tool-tip with additional information on the antenna.

- **Device Browser.** The Device Browser is a tool that can connect to any EVLA device and provides detailed information on that device. Users enter the name of the device, for example ea14-acu.evla.nrao.edu, and the tool will connect to the device and display a hierarchical view of the device listing all monitor and control points. The tool also provides the ability to plot multiple monitor points simultaneously and change control point values.

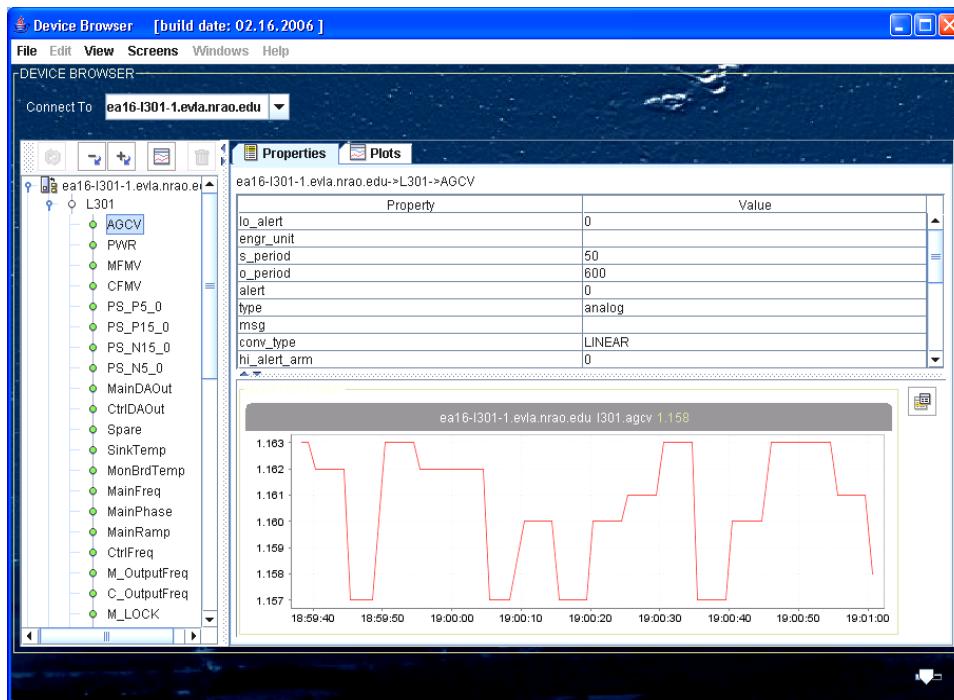


Figure 3 Screenshot of Device Browser

10.5.4.2 Web-based Client

The Web-based client can be accessed by launching a Web browser and navigating to <http://mcmonitor/evla-screens>. The current version of the Web-based application has limited functionality; this will change, however, as development on the Java application decreases.

- **Script Submission Screen.** This screen allows the user to submit a script to the Executor. It also shows scripts that are actively running as well as those that have completed. Several icons appear next to the scripts that allow the user to get more information on the script or to abort the script.

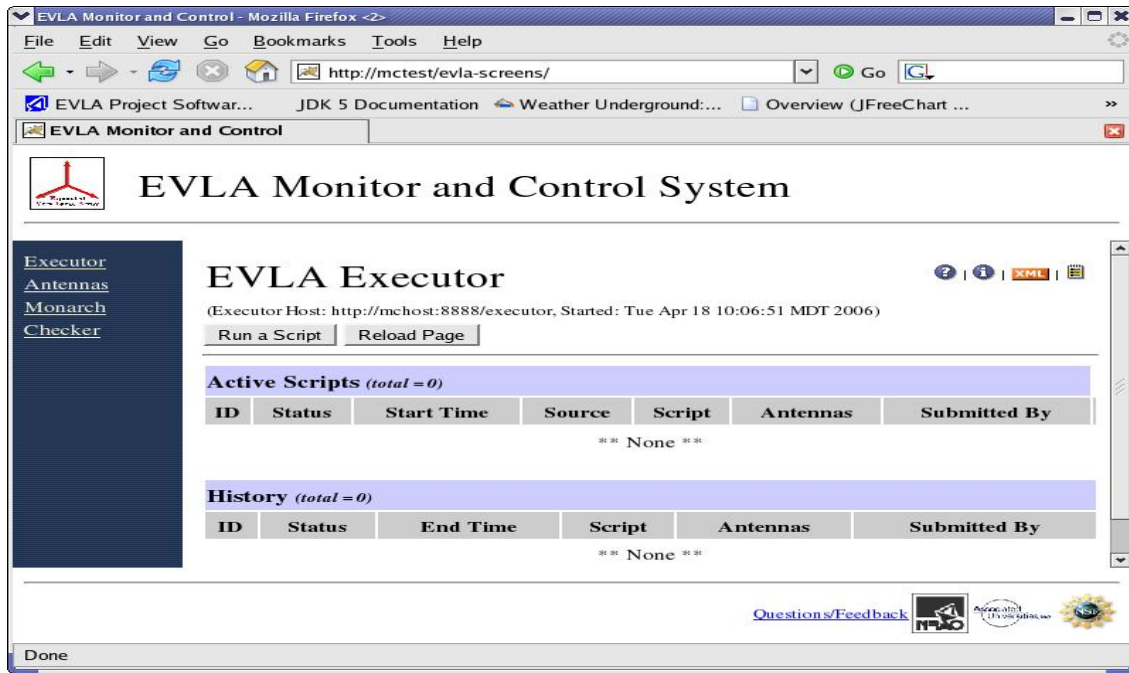


Figure 4 Screenshot of sample web-based page

- **Checker Screen.** The Checker screen displays a table containing the active alerts. The alerts are listed in time-order, listing basic information about each alert; time the alert occurred, the antenna from which it originated, and the module and monitor point IDs. A “history” link exists that, when selected, displays the last 100 alerts
- **Archive (Monarch) Screen.** The archive screen provides information about the data received from the monitor archive multicast group and the monitor data being stored into the Oracle database. The page has a table that contains a list of all sources of archive data, and for each source there are columns with the number of packets and total bytes received from that source. This page has been a valuable tool in measuring the amount of data being sent to the archive.

10.6 EVLA Monitor and Control Network (MCN)

The EVLA Monitor and Control Network links all antenna, correlator, and backend devices to the central Monitor and Control systems.

10.6.1 MCN General Description

The MCN, with one minor exception, will be fiber Ethernet. The exception (noted in 10.6.3.1.2) will be twisted pair copper. TCP and UDP packets will carry commands and status information between the control systems and devices. Each antenna will be treated as its own Class C network.

10.6.2 MCN Requirements

The MCN must be able to support expected M & C traffic both in functionality and in load. The MCN must also not hinder instrument performance either through RFI or availability.

10.6.2.1 MCN Hardware Requirements

10.6.2.1.1 MCN Performance requirements

The MCN must be able to sustain an aggregate 200Kb/s per antenna and 4000 packets/s per antenna. (Assumes 1 packet/10ms* 40 MIBs per antenna.)

10.6.2.1.2 MCN RFI requirements

The MCN must meet the RFI requirements defined in section 3.8 of the Project Book, and the requirements given in EVLA memo #46, “RFI Emission Goals for EVLA Electronics” (<http://www.aoc.nrao.edu/evla/memolist.shtml>).

10.6.2.2 MCN Software Requirements

10.6.2.2.1 MCN Protocol support

The MCN must support both TCP and UDP packets. The MCN must support any protocol such as FTP, HTTP, RPC mandated by the MC software system. The central distribution switch must support both Layer-2 and Layer-3 routing. This switch must have VLAN capabilities on all ports.

10.6.2.2.2 MCN Access requirement

Access to portions of the MCN may be required from remote locations. The exact details of this access will be defined at a later time. Those details should not directly affect the physical design of the network.

10.6.3 MCN Design

10.6.3.1 MCN Hardware in control building

The MCN will be a mixture of 100-1000Mbit single mode and multi mode fiber. Multiport fiber switches will be used to connect all components of the Monitor and Control System (MCS). The switched fiber fabric should meet performance and software requirements as well as mitigating RFI. QoS (quality of service) functionality may be desirable to ensure proper prioritization of traffic. Specifically the QoS capabilities must extend to VoIP traffic.

10.6.3.1.1 MCS Central Hardware

All MCS computers in the control building will be connected with 1Gbit full duplex multi-mode fiber through switches. The link between this cloud and other sections of the MCN will be 1Gbit multi-mode as well though 10Gbit may eventually be required.

10.6.3.1.2 Deformatters

The MCN connection to the deformatter boards will be through 100Mbit twisted pair copper. These devices will be physically located in the (shielded) correlator room. They will be addressed as if they were in their associated antenna.

10.6.3.1.3 LO Tx/Rx and power

These devices will be in the control building but will also be addressed as if they were internal to their antenna via the VLAN capabilities of the central distribution switch.

10.6.3.1.4 Other MCS devices in control building

All other MCS devices in the control building such as the weather station, correlator, and backend cluster will be accessed via multi-mode full duplex fiber. Individual connections will be run at 100Mbit or 1Gbit as required.

10.6.3.1.5 MCS Control building to Antenna link

Each antenna will be connected to the Control building via a 1Gbit full duplex single-mode fiber. All antennas will be connected using attenuated long distance network interfaces that will work over the entire range of distances.

10.6.3.2 MCN Antenna Hardware

10.6.3.2.1 Antenna to MCS Control building link

Each antenna will have a fiber switch with a mate to the control building end of the link. All antennas will be connected using attenuated long distance network interfaces that will work over the entire range of distances.

10.6.3.2.2 MCN antenna network

Each antenna will have a single fiber switch. One port will be connected to the MCS network as described in the previous section, the remaining ports will be directly connected to the MIBs via 100Mbit multi-mode fiber. Additional ports on the switch will be available for transient devices such as laptops or test equipment. These devices will also connect via 100Mbit multi-mode fiber. Until fiber based phones are available the 100Base-T VoIP phones will be connected to the switch via a media converter. The antenna switch should be capable of isolating broadcast between MIBS while allowing direct MIB to MIB communication where needed.

10.6.3.3 MCN addressing

The scale of the MCN requires that device addressing be separated into logical blocks of reasonable size.

10.6.3.3.1 Antenna addressing

Each antenna will be a single Class-C network of the form aaa.bbb.xxx.yyy where xxx defines the antenna and yyy defines the device in the antenna. The aaa.bbb portion will have a fixed value of 10.80. The xxx portion will be the antenna number +100. The yyy portion will be the slot number assigned as per the document “EVLA Hardware Networking Specification (Wayne M. Koski, document #A23010N0003*”). As referenced in 10.6.3.1.3 some devices may be addressed as part of an antenna even though they are not physically in the antenna. Two or three of these Class-C networks will be set up in the AOC to facilitate testing. These networks will be addressed as 10.64.x.y.

* The referenced document can be found at:

\\Filehost\evla\techdocs\sysintegrt\system_documentation_standards\A23010N003.pdf (Windows), or
file:///home/evla/techdocs/sysintegrt/system_documentation_standards/A23010N003.pdf (Unix)

10.6.3.3.2 Control building addressing

The MC systems in the control building that include both control computers, switches and AMCS devices will be addressed together as the zero'th antenna.

10.6.3.4 MCN access

Access to the MCN will be restricted and based on the point of origin of the remote connection. Types and levels of access from specific sites have yet to be determined. The selection of the 10.x.y.z network automatically precludes direct access from non-NRAO facilities. We are capable of allowing (or blocking) direct traffic to the EVLA for those links for which we have complete end-to-end management.

10.6.3.4.1 MCN access from VLA systems

Specific access requirements still to be addressed. The VLA network and EVLA network are separated by the site router. Allowing or disallowing traffic flow between the networks can be easily controlled at either router.

10.6.3.4.2 MCN access from AOC systems

Specific access requirements still to be addressed. The AOC and EVLA sites are directly connected by the VLA and AOC routers. Allowing or disallowing traffic flow between the networks can be easily controlled at either router.

10.6.3.4.3 MCN access from NRAO systems

Specific access requirements still to be addressed. All NRAO facilities have direct connections to the AOC router and therefore direct access to the VLA site router and EVLA. Traffic flow can be controlled between the EVLA and any of the sites to meet access requirements. The Mauna Kea and Los Alamos VLBA stations are the lone exceptions. From a network perspective they appear as non-NRAO systems

10.6.3.4.4 MCN access from non-NRAO systems

Because the EVLA is in the 10.x.y.z network, direct traffic flow to the MCN is not possible from non-NRAO systems. By convention packets with this network address are not forwarded by internet routers. Indirect access to the EVLA network from non-NRAO facilities will fall into one of two categories.

Non-NRAO entities at non-NRAO facilities will first connect to a non-EVLA system likely located at the AOC. From there traffic will be limited in the same manner as it is for AOC systems. Since the link from the remote site to the AOC and from the AOC to the EVLA are disjoint, some form of interface or proxy will have to be designed for the AOC end of the system.

NRAO entities at non-NRAO facilities will be supplied with VPN (Virtual Private Network) client software. This will enable them to appear to be physically in the AOC even though they are not. Traffic flow will appear to be direct from the non-NRAO system to the MCN even though it will go through an intermediate system at the AOC. This form of link will not require a separate interface or proxy as the previous style will.

In both cases access can be restricted at the AOC independently of standard AOC traffic if so desired.

10.7 Transition Planning

10.7.1 Overview and Issues

Several documents exist that describe the basic plan for and status of the transition from the VLA through a hybrid array to the final form of the EVLA Monitor and Control System. All of these documents can be found on the EVLA Computing Working Documents web page, <http://www.aoc.nrao.edu/evla/techdocs/computer/workdocs/index.shtml>. The relevant documents are:

- #37, “Draft VLA/EVLA Transition Observing System Development and Re-engineering Plan”, Tom Morgan, 09/03/2004
- #40, “EVLA Monitor and Control Transition Software Development Plan”, Version 1.0.0, Bill Sahr, 03/25/2005
- #43, “EVLA Monitor and Control Software, Status as of Q2 2005”, Ver 1.0.0, Bill Sahr, 06/20/2005
- “EVLA Monitor and Control Transition Software Development Plan”, Ver 2.0.1, Bill Sahr, to be completed and posted to the web page sometime in late April 2006

As one would expect, the plan calls for the gradual migration of functionality from the VLA Monitor and Control System to the EVLA Monitor and Control System. In fairly high level terms, the steps or milestones of the Transition Plan are:

1. Support for EVLA antenna hardware development
2. The participation of EVLA antennas in scientific observations
3. The monitor and control of VLA antennas by the EVLA Monitor and Control System
4. The monitor and control of the VLA correlator by the EVLA Monitor and Control System, coupled with the distribution of VLA correlator output within the EVLA Monitor and Control System
5. The formation and writing of VLA format archive records by the EVLA Monitor and Control System
6. A period of parallel operation and testing

Item 1 is complete and item 2 is happening now. Item 3, the monitor and control of VLA antennas, is expected to be complete before the end of Q2 2006. Monitor and control of the VLA Correlator will have two stages – first the new VLA correlator controller (see section 10.7.3.3) will operate under the Modcomp-based VLA Monitor and Control System. Once fully correct and complete operation has been unambiguously confirmed, control of the VLA correlator will move to the EVLA Monitor and Control System. While the new VLA correlator controller is still under the control of the VLA Monitor and Control System, the interface to the correlator that is needed to distribute VLA correlator output within the EVLA Monitor and Control System will be developed. The last development milestone is creation of what is being called an interim data capture and format (IDCAF) component. It will be the responsibility of IDCAF to form and write VLA format archive records on behalf of the EVLA Monitor and Control System. IDCAF development was started in April 2006.

While it is conceivable that all of the work needed for the retirement of the Modcomp-based VLA Control System could be completed by the end of Q4 2006, a more likely date is Q2 2007, especially given the needed period of parallel operation and testing. There is also the matter of the WIDAR prototype correlator. On-the-sky (OTS) testing of the WIDAR prototype correlator is currently scheduled to begin in Q3 2007. If analysis of the requirements shows it to be desirable, retirement of the Modcomps and the VLA Control System may be deferred until OTS testing is complete – mid Q4 2007.

10.7.2 Requirements

The scientific community stipulated three general requirements for the transition phase:

- The EVLA Monitor and Control must support simultaneous operation of the old VLA antennas and the EVLA antennas during the transition phase,
- Array down time shall be minimized as much as possible during the transition phase,
- Operations using the old VLA shall be possible using the current OBSERVE/JOBSERVE script files (to maintain backward compatibility with VLA antennas while they exist).

10.7.3 Design

10.7.3.1 Transition Hardware Modules

During the transition, EVLA antennas will contain the F14 module that is present in VLA antennas for control of some of the Front Ends. A transition module that enables monitor and control of the F14 module by the EVLA monitor and control system will be designed and constructed.

An interface will be provided between the EVLA M&C system and two VLA antenna subsystems - the Antenna Control Unit (ACU) and the Focus Rotation Mount (FRM). The ACU controls the movement of a VLA antenna, while the FRM establishes the proper positioning of the subreflector for a given band. This transition interface will exist until a replacement ACU and FRM is designed for the EVLA project.

The Digital Transmission System (DTS) deformatter will contain a filter that transforms an EVLA digital IF to an analog IF that is compatible with the current VLA correlator. This module will also match EVLA sidebands to VLA conventions when necessary.

10.7.3.2 Monitor and Control of VLA Antennas

VLA antennas awaiting upgrade to EVLA status will be controlled using the EVLA M&C system through the Interim Control & Monitor Processor (CMP). The CMP interfaces to the VLA antennas through the waveguide via the Serial Line Controller (SLC). The CMP is connected to the second port of the SLC and currently operates in parallel with a Modcomp connected to the first port. Eventually, the Modcomp will be retired and the CMP will become the only monitor and control path for the VLA antennas.

The CMP is physically two VME Single Board Computers residing in a VME chassis that is located above the SLC in the Control Building at the VLA site. An MVME142 (Motorola 68040) running VxWorks contains the SLC driver software and provides the real-time services timed by waveguide cycle interrupts from the SLC. The second processor is an MVME5100 (Motorola PPC750) running TimeSys Linux that provides the EVLA M&C functionality. The two processors communicate via shared memory over the VME bus.

10.7.3.3 Monitor and Control of the VLA Correlator

To replace obsolete, unsupported hardware, and to assist in the transition phase by providing a network accessible controller, a replacement Correlator controller is being built. The new controller is a VME based computer system designed to accept the current Modcomp control and data dump formats. The control path is a network connection, which makes it possible to connect the controller to other external systems, including the EVLA Monitor and Control system. The VME system will consist of a Single Board Computer (SBC) with an Ethernet interface for command/control, and a separate array processor to receive Correlator integrator data and perform final processing. The new system will be installed in the system controller rack of the VLA Correlator.

Currently the system is in the test and deployment stage. The new system has been demonstrated to work in both line and continuum with all modes and options. Astronomical testing has begun, as have plans for operational deployment. Commissioning is expected in Q2 2006. The system has already been introduced to the operations staff during overnight tests.

10.7.3.4 One Stage of the Hybrid Array

The diagram given below shows one stage of the VLA in transition to the EVLA. The dotted boxes show EVLA resources. VLA and EVLA antennas are both present, with the CMP providing a monitor and control path to VLA antennas. An EVLA monitor data archive is up and running, and the new VLA correlator controller is in place. With the exception of the new VLA correlator controller not yet having reached operational status, this diagram is an accurate reflection of the system as of April 2006.

