EVLA Project Book, Chapter 9

EVLA Monitor and Control System

Wayne Koski, Rich Moeser, George Peck, James Robnett,
Bruce Rowen, Kevin Ryan, Bill Sahr
Last changed 2002-May-31

Revision History
2001-July-16: Initial release
2001-Sept-25: Initial draft of text
2001-Oct-16: Minor cleanup & revisions
2001-Nov-20: Links to Requirements Documents added
Section for Correlator Requirements added
Link to EVLA Backup Monitor And Control System document added
2002-May-31 Major revision. Entire chapter reorganized and updated

Summary

This chapter outlines the current state of requirements and design considerations for the EVLA Monitor and Control System, including hardware, software, and computing systems. It includes discussion of antenna monitor and control, correlator monitor and control, the operational interface for the system, and issues relating to transition planning.

Current milestones for the EVLA Monitor & Control System are as follows:

Milestone	Date
EVLA System PDR	12/04/2001
EVLA Monitor and Control Hardware PDR	03/13/2002
EVLA Monitor and Control Software PDR	05/14/2002
Prototype M&C module interface board	07/15/2002
Bench testing of EVLA M&C modules to begin	01/2003
Outfitting of first EVLA antenna	Q2 2003
EVLA electronics production	Q4 2003
Begin retrofitting 7 antennas/yr with new electronics	Q2 2004
Observing with hybrid array (transition mode)	Q2 2004
Prototype correlator	Q4 2005
Begin outfitting new correlator room	Q2 2006
Begin testing of correlator subset at VLA	Q4 2006
First science with correlator subset	Q2 2007
Last antenna retrofitted to EVLA design	Q1 2008
New correlator fully operational	Q1 2009
Last EVLA receiver installed	Q1 2010

9.1 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, Correlator Monitor and Control, the Operational Interface System, and the EVLA Monitor and Control Network. Additionally, sections have been added for the Observing Layer, the Test Antenna Software, and for Transition Planning. The Observing Layer is the business logic of the system. It is that layer of the software that contains the astronomical knowledge, information and strategies needed to conduct an observation – coordinate systems and transforms, propagation from a position of epoch, calibration strategies, LO chain calculations, determination of delay settings, etc. Transition Planning is an umbrella term which covers the issues, planning and strategies needed to make the transition from the current VLA to the EVLA with a minimal loss of observing time and observing capabilities.

The sections covering the four subsystems all have a similar structure:

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

with additional subsections added as appropriate.

9.1.1 Current Deficiencies

The EVLA Monitor and Control System Software PDR was held on May 14^{th} and 15^{th}, 2002. Prior to the PDR we were aware of a number of deficiencies in the current state of the software planning. A formal PDR Review Panel report has not yet been produced, but the Panel Review session held at the end of the PDR confirmed our view of what is missing. At this point, those of us working on the EVLA Monitor and Control Software have identified the following points as those needing immediate attention:

- The lack of a high level overall software architecture/plan
- The lack of a detailed, timelined development plan for the test antenna software
- The security requirements and a design meeting those requirements have not been developed for the EVLA Monitor and Control Network
- No one is currently working on the Observing Layer of the software
- A detailed, timelined Transition Plan is needed

The lack of an overall software plan is a glaring deficiency. Given the current manpower situation within the Computer Division, a choice had to be made between developing an overall software plan versus progress on issues critical to the test antenna. Since we will not know the true RFI environment of an EVLA antenna until we have at least a large subset of at least semi-functional hardware and software on a test antenna, the needs of the test antenna were given priority.

The same logic applies, in a somewhat different manner, to the lack of a detailed plan for the test antenna software. Certain issues were obvious. A processor for the module interface boards that serve as monitor and control gateways for the antenna subsystems had to be chosen. The use of Ethernet as the monitor and control fieldbus leads to the need for a network stack in the processors. The need for a network stack implies the need for a real-time operating system (rtos). Conformance to RFI guidelines resulted in the selection of processor chip for the antenna monitor and control system that is a new chip, scheduled for volume production in Q2 2002. No rtos could be found that had already been ported to this chip, so a contract for a port must be negotiated, etc. In other words, certain core issues were so obvious and so time critical that it was possible to proceed with these issues in the absence of a detailed plan for the software. It is these issues, plus requirements and the exploration of design issues for the various subsystems that have occupied our time.

Many/most of the time critical decisions relating to the test antenna software have now been made. The paperwork for the needed procurements is proceeding and should be finalized by mid-June 2002. By July 2002, at the latest, it should be possible to begin to address the issues of an overall software development plan and a detailed plan for the test antenna software.

To say that we have not yet developed the security requirements for the EVLA Monitor and Control Network is not to say that a lot of planning and discussion has not been done on the network. A reasonably well developed plan exists for the network within each antenna, and for the connections from the antennas to the VLA Control Building. A good picture also exists of the networking requirements for correlator monitor and control. Security issues have been raised throughout the course of this planning and discussion. However, a systematic attempt to formally address the security issues has not yet been organized. The end of June 2002 has been identified as the time at which that effort will begin.

Personnel to assign to the Observing Layer of the software is a recruitment issue, which is best discussed elsewhere.

As with the security requirements and design for the EVLA network, so has it gone with transition planning. Plans have been made and the development of components critical to the transition is well underway, but no formal, detailed plan with dates and milestones has yet been developed.

9.2 Description of the EVLA Monitor and Control System

9.2.1 General Requirements

9.2.2 General Design Consideration

9.3 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. The AMCS may also be required to operate some of the nearby VLBA antennas and, if Phase II of the VLA Expansion Project occurs, the New Mexico Array (NMA) antennas as well.

9.3.1 AMCS General Description

Physically 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 either VME/CompactPCI single crates or high reliability desktop machines. Processors located within the antennas will be small micro-controller type processor boards with small amounts of 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.

9.3.2 AMCS Requirements

9.3.2.1 AMCS Hardware Requirements

9.3.2.1.1 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.

9.3.2.1.2 Ethernet

The use of Ethernet as the bus is considered a requirement because it allows the entire AMCS to use one bus, there is COTS equipment available, it is maintainable due to widespread commercial use, it allows addressing by slot, and it is well suited for object-oriented programming.

9.3.2.1.3 Data Rates

The maximum data rate from an EVLA module is estimated to be 128 Kbits/sec, and 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.

9.3.2.1.4 Timing

Reconfiguration commands sent by the ACMS must begin not more than 100 μs 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.

9.3.2.1.5 MIB (Module Interface Board) Requirements

The MIB must be small in size. It must implement a communications protocol for exchanges between the MIB and the world external to the antenna. A communications protocol is also required between the MIB and the devices in the antenna that it controls. The MIB must be able to obtain its software from an external source when it boots. It must be able to send back monitor data periodically or on demand.

The MIB must not implement any code that is necessary for the safety of an EVLA module. Each EVLA module must protect itself from damage, even in the absence of a MIB.

9.3.2.2 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".

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 an Ethernet field bus 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 μSec command start latency. This means that a command must be initiated at the hardware within 100 μSec of its scheduled start time.
  - Pointing updates every 50 milliseconds. To maintain a sub-arcsecond level of pointing accuracy, the antenna position must be updated at least every 50 milliseconds. Since the servos will be able to slew on their own, this might not have a direct impact on the AMCS software.
  - Frequency change within band to be completed within one second.
  - 'Nodding' source switch rate of once per ten seconds.

9.3.3 AMCS Design

The AMCS is being designed with the intent of creating not only a system that is state-of-the-art with today's technology but will also 'scale' with new technology as it comes about. To achieve this scalability, the system is being designed with modularity always in mind. By doing so, the system will develop into a collection of 'plug-and-play' components that can be replaced without causing adverse affects on the rest of the system.

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.

9.3.3.1 AMCS Hardware

9.3.3.1.1 General Description

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

9.3.3.1.2 Module Interface Board (MIB)

Every EVLA module or device will contain a Module Interface Board (MIB). The MIB will communicate with the Antenna Control Computer via Ethernet, and will be available to carry out housekeeping tasks for the devices. Communications with a device will primarily be carried out via Serial Peripheral Interface (SPI), and parallel communications. Planned capabilities of the board include Reset Logic and Timing Logic. Reset Logic would include power-on-reset, power perturbation reset, watchdog timing reset, and user-requested reset. Timing logic will time events with external timing signals that the board receives.

The MIB will be a separate small board. It will be placed very near the device that it monitors and controls. A pair of fibers will directly connect each MIB to the Ethernet. (A pair of fibers is needed to accommodate the full duplex nature of fiber-connected Ethernet.)

The core of the MIB will be an embedded processor that has all of the necessary communications interfaces (Ethernet, SPI, parallel) built in. This processor will implement the necessary software protocols such as TCP/IP. There will be enough Flash Memory on board the embedded processor to store the software that runs the processor. There will also be Data Storage Memory, to store commands and monitor requests for the antenna computer.

The MIB and the racks will be designed such that the MIB will detect the slot into which it is plugged, thus eliminating the need to change the module address when the module is moved.

9.3.3.1.3 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.

9.3.3.1.4 Voice Communications

9.3.3.2 AMCS Software

The AMCS software design philosophy is that of ‘scalability through modularity’. Vendor specific technologies are being avoided in favor of more industry-wide approaches in order that no part of the software system will have to rely on any one particular vendor for its future growth. An example of this is favoring the use of SOAP over CORBA for software object brokering over the network.

The one major exception to this rule is the operating systems chosen for the various processors. While versions of real-time Linux (an OS with broad industry support) do exist, it was necessary to select a vendor-specific RTOS for the MIB. Real-time Linux requires too much RAM to be accommodated by the memory available to the MIB processor.

All but the low-level hardware driver software will be designed using Object Oriented techniques.

Because of the highly networked, distributed nature of the AMCS system, it is hoped that it will be possible to use Java throughout the AMCS system in all but the hardware driver areas. Java is a language designed from the onset around networking and distributed processing. Java was also designed to be a language for large systems with its packaging structure and built in documentation generation. It is estimated by developers experienced in both languages that software development in Java takes roughly half the time as software development in C++*.

A port of the Java Connected Limited Device Configuration (CLDC) for the MIB rtos does exist, but it must also be ported to the MIB processor chip. The CLDC includes a java virtual machine plus certain core classes. It requires somewhat less than 100 Kbytes of RAM. Adding java.net would consume approximately an additional 100 Kbytes.

*” Thinking in Java”, 2nd edition, Revision 12; ©2000 by Bruce Eckel

9.3.3.2.1 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. All of the processors in the system will incorporate an Operating System capable of providing the various Ethernet and IP protocols. This means that even the otherwise simple MIBs will have to support an OS capable of higher level network protocols such as TCP/UDP/IP and possibly others such as HTTP and XML.

Data communications between the various devices will become more generic and higher-level in nature as opposed to dedicated field busses like the VLBA's MCB and HCB where simple address/value pairs of binary data are communicated.

Heterogeneous Hardware. AMCS design takes advantage of the fact that every component of the system, including the subcomponents within the antennas, are 'intelligent' devices with a network presence, indeed each with its own IP address. Because of this, the functionality of the system as a whole can be distributed among the various devices that make up the physical system.

Distributed processing allows the design of a modular system separated into its various functional components of arrays, sub-arrays, antennas and antenna sub-components (such as servos, LO, etc.). Each of these components will contain its own processor which will present the 'front-panel' of that component to the rest of the system while hiding the implementation details of the component from the system. The implementation details will be encapsulated.

Encapsulation allows the system to meet the requirement of being able to accommodate heterogeneous antenna types and associated hardware. Components can be sent generic, high-level type commands from the system controller. For example, two different type servos, each with a different position resolution, can both be commanded to say 47.0˚ by sending the floating-point value of 47.0 to each instead of having to send two different binary values corresponding to the resolution of each.

Variety of Users from a Variety of Locations. The modular components of the AMCS will be fully autonomous. They will accept commands that are of a 'what-to-do' nature as opposed to 'how-to-do-it'. Each component will also monitor itself. It is a trivial matter, for example, for a component to monitor the voltage reading of its own power supply. It can sample the values much faster than if they were instead sampled from across the network by a central processor. This has the advantage that loggers, archives and monitor displays will not be 'clogged' with trivial normal-valued monitor data.

A system of autonomous components means that an antenna or antenna subcomponent can be fully operated separately from the rest of the system such as when undergoing development and maintenance.

Because components will be autonomous they can act as servers for their particular functions. This allows the system to be designed as a type of client/server system. Clients can request the services of any of the components that make up the system. This architecture makes for a very flexible system as seen by external systems. User Interface clients can be made 'thinner' (less application specific); in fact, web browsers will probably be used to interface some parts of the system.

Real-Time Considerations. Because of the way the AMCS is being designed, real-time operations can take place where the action is required. Instead of a central processor issuing real-time commands over several kilometers of network, the commands can be initiated within the component itself. This implies that configuration parameters will have to be sent to the various components ahead of time so that they will be ready for implementation time. Just how far 'ahead of time' and whether just one or blocks of several configuration changes will be sent out to the devices is still under discussion with the e2e group since it affects their Dynamic Scheduling process. It will probably work out that 'minutes' worth of configuration information will be queued with the ability to cancel or replace the current set for such occurrences as emergency target-of-opportunity observations.

9.3.3.2.2 AMCS Systems Software & Middleware

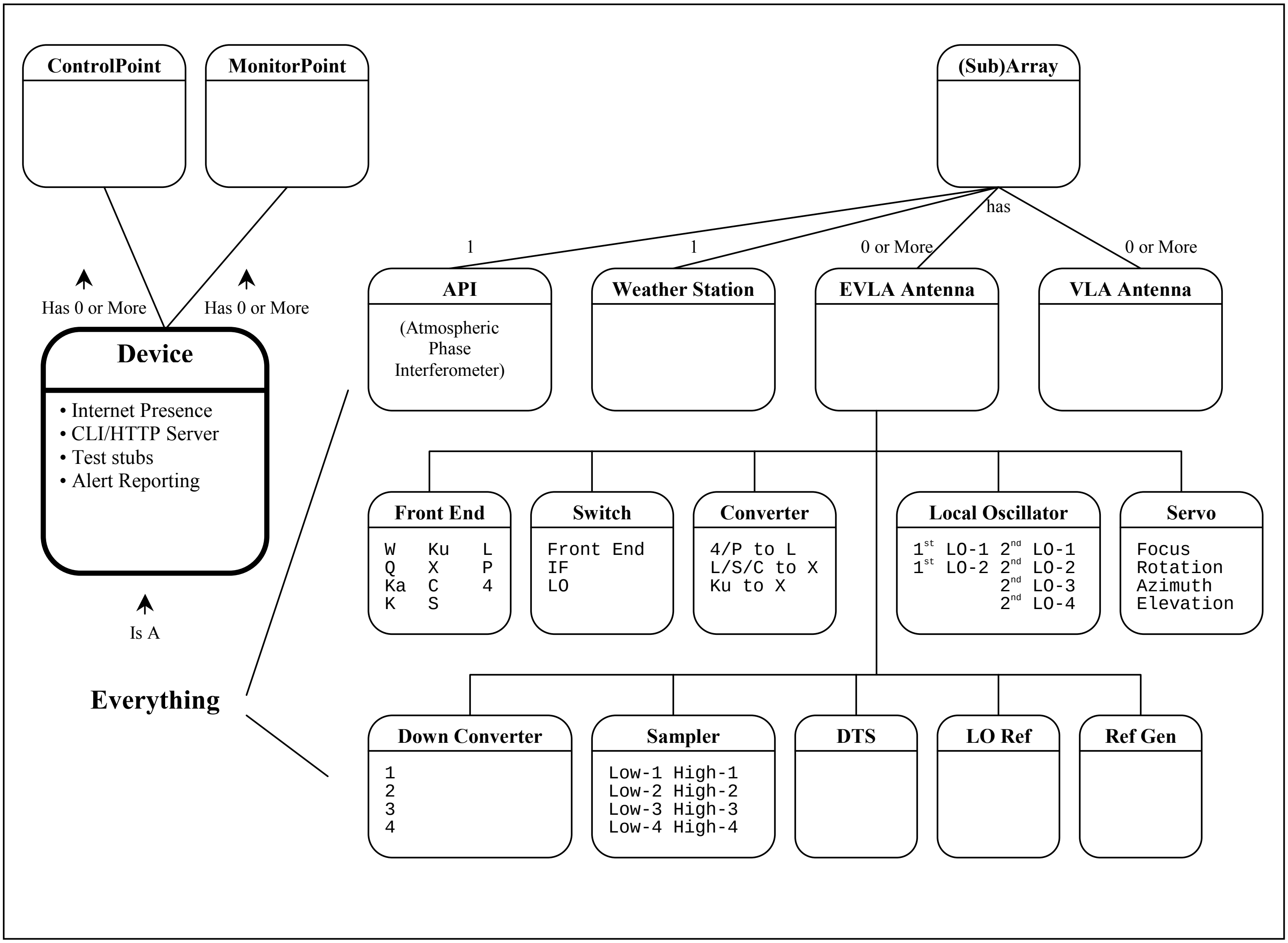
While real-time operations are few, the need for a real-time operating system (RTOS) exists for the processors in the AMCS. The driving force for the MIB was not only for actual real-time operations but also to obtain the necessary networking functions (e.g. the TCP/IP stack) while maintaining a small overall memory footprint. Because of the newness of the MIB's processor and memory size limitations, choices for a MIB RTOS are quite limited compared to that of the other processors that will be used in the system. The choices were narrowed to two and after much research and negotiations the final choice of ATI's Nucleus RTOS was made.

Several technologies are currently being investigated for the so-called middleware. This middleware will essentially be the AMCS's interface to the Operational Interface System (OIS). As such, this area of software is being investigated by both the AMCS and OIS designers. It is a goal to use COTS software with industry wide support and popularity if possible.

The AMCS and OIS will be designed to work together to form a User Interface that will be malleable by the users themselves and that will change with the AMCS system without requiring re-programming. To accomplish this, the middleware will have to allow the OIS to discover the attributes and capabilities of the AMCS's components and be able to operate them directly. Two of the technologies under investigation are Java's built-in RMI (Remote Method Invocation), and SOAP (Simple Object Access Protocol) via XML message passing.

9.3.3.3.3 AMCS Application Software

AMCS Software is being designed using Objected Oriented Design (OOD) principles. Each physical component of the system will map directly to a software object. In this system every component class will inherent from a base class called a *Device* class. The Device class will provide all of the functionality that is common to each of the EVLA components. This includes the network presence, collections of ControlPoint and MonitorPoint objects and stubs for



methods (functions such as confidence and diagnostic tests) common to all components. By inheriting from a common base class, all Devices in the system will have a common interface to the rest of the system.
As mentioned earlier generic User Interfaces will be able to 'discover' the various components of the system thereby eliminating the need for 'hard-coding' lists of devices (and their characteristics) into higher-level system controllers. This in turn allows the system to change physically without necessitating change at User levels of software.

The component classes will communicate with the low-level hardware 'driver' software. This driver software communicates directly with hardware via the processor's I/O ports. The interface between the low-level, driver software and the higher-level component objects will involve the transformation of raw control and monitor data into ControlPoint and MonitorPoint objects. It will also involve the translation of object method invocations into calls to the associated low-level driver routines. These driver routines such as specific hardware control functions and various diagnostic and test functions, will be, for the most part, written and maintained by the Hardware Engineering personnel.

9.4 Correlator Monitor and Control Subsystem (CMCS)

9.4.1 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. A separate layer above the VCI will provide extensive diagnostics and control primitives for engineering and technical support. It is the intent of this separate layer to provide the necessary tools for Correlator development during the construction stage.

9.4.2 CMCS Requirements

A more detailed description of CMCS requirements may be found in EVLA memo #15 "EVLA Monitor and Control System, Test Software and Backend Software Requirements and Design Concepts". 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

9.4.2.1 CMCS Hardware Requirements

The CMCS shall consist of a network of distributed processors with each processor responsible for managing a discrete functional block 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 non-volatile storage for completely self-contained booting of the operating system, run time code, and Correlator board firmware. 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.

9.4.2.2 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 systems 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.

9.4.3 CMCS Design

9.4.3.1 CMCS Hardware

It expected to use PC-104+ mechanical form factor computer boards 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 either be over the PC104+ bus (PCI standard) or through a PCMCIA bus interface. Both are industry standards. This bus will allow the CMIB to download Xilinx personalities and control words to the Correlator hardware as well as extract monitor and ancillary data from the Correlator. It will be desirable to have all Xilinx personalities as well as CMIB OS and run time code stored in CMIB flash style memory. This will allow “safe” power up and bench testing of individual Correlator units without any external networking in place. 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 EVLA memo #15.

9.4.3.2 CMCS Software

Due to the need for flexibility and portability of 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 constrained by memory or CPU speeds, many commercial and public OS choices exist. The preference is to try Linux first since there is already a large installed base within the organization and it has proven to be both reliable and flexible. Selecting the OS for the CMIB is a bit more complex since the system has some modest real time requirements and is more restricted in CPU power and memory (to keep cost down). There exists a wealth of OS choices for these small SBCs among which are the various flavors of real time Linux. Due to the modest interrupt rates and need of good networking connectivity, several of the preempt-able Linux kernels seem like good choices for initial testing. 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.

9.5 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 client software tools that allow the array operators, engineers, technicians, scientists, software developers, and other authorized users to interact with the array in a safe and reliable manner.

9.5.1 OIS General Description

The Operational Interface Subsystem will provide the primary graphical user interface (GUI) tools for the EVLA Monitor and Control System. It is through OIS that users will monitor and command the array. This section will discuss the various components of OIS, the functions OIS must provide and the users of OIS.

9.5.1.1 Components

- **Client Stations**. A client station is a user’s desktop or laptop computer that runs the OIS software. Its primary responsibility is to run the OIS software and will do this either as a stand-alone program that executes on the client station or through a browser. The type and operating system of each client station will likely vary, as will the location. Client stations can be located at the VLA Control Building, the AOC or any Web-accessible location.

- **Operations Server.*** The OIS software will communicate directly with the operations server. The server will have the responsibility of transmitting data acquired from the core monitor and control real-time system to the outlying client stations. It will also receive commands issued from authorized client stations and forward those commands to the core real-time system.

* The idea of an operations server is a preliminary concept that may change if decentralized network architecture such as a peer-to-peer is selected.

9.5.1.2 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.

- **Logging.** The Operational Interface Subsystem will provide a tool that enables authorized users to create and send messages to a message log. This will replace the functionality currently provided by the observing log that is generated by the array operators using Microsoft Excel.

- **User/Operator Communication.** The Operational Interface Subsystem will provide a simple messaging (or chat) tool that allows operators and observers to communicate during an observation. This will allow corrections or changes, suggested by the observer, to be made during the observation in an attempt to increase the quality of the data being collected.

9.5.1.3 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 both monitor and control capabilities of 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 the progress 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.

- **Web users.** Web users are those individuals that are part of the general public. They will have the ability to monitor the system and will not have any control capabilities.

9.5.2 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 "Operational Interface Software Requirements Specification."

9.5.2.1 OIS Hardware Requirements

It is unlikely that OIS will communicate directly with any hardware. It will, however, communicate directly with the software interface for specific pieces of hardware (e.g., the *Device* interface) 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.

9.5.2.2 OIS Software Requirements

The software requirements document referenced above contains a detailed description of requirements imposed on OIS; it is worth mentioning here the 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.

- **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. Several methods exist, including downloading a file (e.g., tar or zip) and running an installation script, accessing the software from a browser using Java applets, or Java Web Start software which is an application deployment mechanism for Java applications. Such methods are currently under consideration, however, Java Web Start is extremely attractive due to the fact that it requires very little interaction on the part of the user and upgrading the software is simple and relatively transparent to the user.

- **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 simply warn the user that a failure has occurred, but it should continue working without incident. For example, if communication to an antenna is lost it should not affect the acquisition of data from all working antennas. And when the antenna is functioning and back online, the system should automatically resume data acquisition as if nothing happened.

9.5.3 OIS Design

The design goal of the Operational Interface Subsystem is to meet the requirements stated in the "Operational Interface Software Requirements Specification" 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.

9.5.3.1 OIS Hardware

OIS will not communicate directly with the hardware. Its 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.

9.5.3.2 OIS Software

9.5.3.2.1 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. An example of a coarse-grained interface that lends itself nicely to a loosely coupled system is the *Device* interface described in section 9.3. The *Device* interface provides a method that retrieves a collection of *Device* objects so that the addition or removal of a device for that particular subsystem in no way affects the software that requests the collection of devices.

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 devices 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 devices. 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, features that were either not thought of or left out due to time or budget constraints during the initial requirements and design phase. 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.

9.5.3.2.2 Middleware Considerations

The EVLA Monitor and Control System will be designed as a distributed object system where each physical component (e.g., antenna) and non-physical component (e.g., a subarray) maps directly to a software object. The system must be able to locate these objects and once located one must have the ability to make the objects do something by invoking the object's methods. Middleware such as CORBA, XML-RPC, Java RMI, Web Services (SOAP/WSDL/UDDI), Jini, JXTA, and many more offer just such functions. The EVLA software effort is in the process of defining the communication requirements in order to determine the suitable middleware solutions that are available. It's likely that a single middleware solution will not suffice and that a system-wide, multi-protocol design will be required instead.

There are several determining factors that will affect the decision as to what middleware is used and where it is used. These include performance, interoperability, industry backing, expected lifetime, in-house expertise and budget. Other considerations for middleware include the network architecture, either client/server (Web services) or peer-to-peer (JXTA), imposed by the middleware.

9.5.3.3 Application Software

OIS is being designed as a flexible, highly adaptive system. The system will attempt to decouple itself as much as possible from the core monitor and control system so changes to the underlying system will not affect the OIS software. It will be designed using object-oriented analysis and design principles that foster code reuse through inheritance and hide implementation changes through encapsulation. Follows are a list of preliminary design concepts that are currently under consideration:

Lightweight Client. In order to achieve loose coupling, OIS must have little or no knowledge of the underlying business logic that is the responsibility of the monitor and control system. 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.

The client software must be as lean as possible so that the size of the entire package that must be downloaded is minimized. In some cases users will be downloading the software over a dial-up connection.

Display Framework. A display framework or infrastructure will be designed so that screens can be easily created and added to OIS in short order. All screens will inherit common behavior and properties from a base *Screen* object the same way in which all devices inherit behavior from a base *Device* object. Some commonalities that might exist for all screens, defined by the *Screen* object, are the rate at which the screen updates, the updating of data on the screen, loading of resources for a screen and the overall look and feel for each screen.

Since all screens will have common behavior defined by the *Screen* object, the display infrastructure will simply treat all screens in the same manner regardless of the content of the screen. The primary function of a display framework will be to manage the screens. The responsibilities of the framework include, but are not limited to, the addition, importing and removal of screens, screen selection and creation and the disposal of screens.

Screen Types. It is thought that two types of screens might exist: automatically generated screens and predefined screens. The automatically generated screens will be dynamically constructed at runtime. The purpose of such screens is to make available at runtime any additions or changes to a device (e.g., a new monitor point added to the device.) The primary benefit of automatically generated screens is that users will not have to wait until a predefined screen is updated or created before seeing the changes. The predefined screens differ from the automatically defined screens in that they are more susceptible to changes in the device. If the device changes, in most cases a change to the predefined screen will be required. The primary benefit of the predefined screens is that the screens are highly customizable compared to the automatically generated screens.

9.6 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.

9.6.1 MCN General Description

The MCN, with one minor exception, will be fiber Ethernet. The exception (noted in 9.6.3.1.2) 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.

9.6.2 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.

9.6.2.1 MCN Hardware Requirements

9.6.2.1.1 MCN Performance requirements

The MCN must be able to sustain an aggregate 200Kb/s per antenna and 5000 packets/s per antenna. (Assumes 1 packet/10ms* 50 MIBs per antenna.)

9.6.2.1.2 MCN RFI requirements

The MCN must meet the RFI requirements defined in section 3.8 of the Project Book.

9.6.2.2 MCN Software Requirements

9.6.2.2.1 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.

9.6.2.2.2 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.

9.6.3 MCN Design

9.6.3.1 MCN Hardware in control building

The MCN will be a mixture of 100-10000Mbit single mode and multi mode fiber. Multiport fiber switches will be used to connect all components of the MCS. The switched fiber fabric should meet performance and software requirements as well as mitigating RFI.

9.6.3.1.1 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.

9.6.3.1.2 LO digital receivers

The MCN connection to the LO DR's will be through 100Mbit twisted pair copper. Though these devices will be in the control building they will addressed as if they were in their associated antenna.

9.6.3.1.3 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.

9.6.3.1.4 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.

9.6.3.1.5 MCS Control building to Antenna link

Each antenna will be connected to the Control building via a 1Gbit full duplex single-mode fiber. Any antenna further than 9Km from the control building will be connected through special long distance network interfaces.

9.6.3.2 MCN Antenna Hardware

9.6.3.2.1 Antenna to MCS Control building link

Each antenna will have a fiber switch with a mate to the control building end of the link. If the control building has a long distance network interface then the antenna must have one as well.

9.6.3.2.2 MCN antenna network

Each antenna will have multiple fiber switches. One switch will connect the antenna with the control building and the other switches to each other. These connections will all be 1Gbit full duplex. The connections internal to the antenna will use multi-mode fiber. The other switches will be connected to the MIBs via 100Mbit multi-mode fiber.

9.6.3.3 MCN addressing

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

9.6.3.3.1 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 either be internal to the NRAO-NM Class-B network of

146.88.0.0 or will be a part of one of the non-routable public Class-B networks. As referenced in 9.6.3.1.2-3 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.

9.6.3.3.2 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.

9.6.3.4 MCN access

Access to the MCN will be restricted based on where the remote connection originates. Types and levels of access from specific sites have yet to be determined.

9.6.3.4.1 MCN access from VLA systems

9.6.3.4.2 MCN access from AOC systems

9.6.3.4.3 MCN access from NRAO systems

9.6.3.4.4 MCN access from non-NRAO systems

9.7 EVLA Monitor and Control Observing Layer

9.7.1 General Description

9.7.2 Requirements

9.7.3 Design

9.8 EVLA Test Antenna Software

In a memo dated 02/28/2002, Barry Clark outlined the following task list for the test antenna software

AMCS Module Interface Board (MIB)

- Systems Software and Toolset
- MIB Communications Functions
- MIB Computation and Control Functions

Test Antenna, Single Dish Phase

- Command Line Interpreter
- Timekeeping Routines
- Geometry Routines
- Antenna Pointing Model
- Archive System for pointing model parameters
- Archive System for monitor data
- Operator Interface Screens

Test Antenna, Interferometer Phase

- Translator, Modcomp card input to new system commands

- Lobe rotator and phase switching drivers
- Fiber optic IF system – test, monitor, control
- Data Flagging System

The current short term working schedule for AMCS MIB development is as follows:

Select AMCS MIB chip (the Infineon TC11IB)	Done
TC11IB development board in-house (late)	Received 05/17/2002
Select Systems Software for AMCS MIB	Done (5/21/2002)
Contract for port and integration of systems software to vendor	06/17/2002
Delivery of simulation environment	06/28/2002
Begin development of MIB software apps	07/15/2002
Prototype MIB board available	07/15/2002
RFI testing of MIB prototype	TBD
Port & integration of MIB systems software complete (~ 12 weeks)	09/09/2002
Port of systems software on MIB development board	09/23/2002
Port of systems software on MIB prototype board	10/07/2002
MIB software apps onto MIB prototype board	10/14/2002
Continued MIB software development	Thru 03/2003
Bench testing & integration of M&C components	Thru 03/2003

9.9 Transition Planning

9.9.1 Overview and Issues

Q2 2003 is the current goal for testing of the first EVLA antenna to begin, while Q4 2009 is the current goal for shutdown of the old correlator. The array will, therefore, be in a hybrid state for a period of approximately 6 years. Clearly, monitor & control of the hybrid array is a major issue, and it will require a considerable investment of man-hours. The current plan for development of a hybrid array M&C system is to grow it from initiatives already underway. Three components are critical – 1) the interim control & monitor processor (CMP) which taps the monitor and control data stream of the current VLA, 2) an upgraded, network accessible controller for the current VLA correlator, and 3) a user interface testbed for the EVLA monitor and control system. The CMP is nearly ready. The upgraded VLA correlator controller is under development now, and the beginnings of a user interface testbed will be developed after the CMP becomes operational. When combined, these elements form a basis from which a system for monitor and control of a hybrid array can be developed.

9.9.2 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 OBSERV/JOBSERVE script files (to maintain backward compatibility with VLA antennas while they exist).

An additional, important operational requirement is that the current Modcomp computers be decommissioned as soon as possible. Eventually, it will become increasingly difficult to support and maintain the Modcomp equipment. It is very desirable to avoid end-of-life issues by replacing the equipment before it has been declared obsolete and no longer supported by the manufacturer.

9.9.3 Design

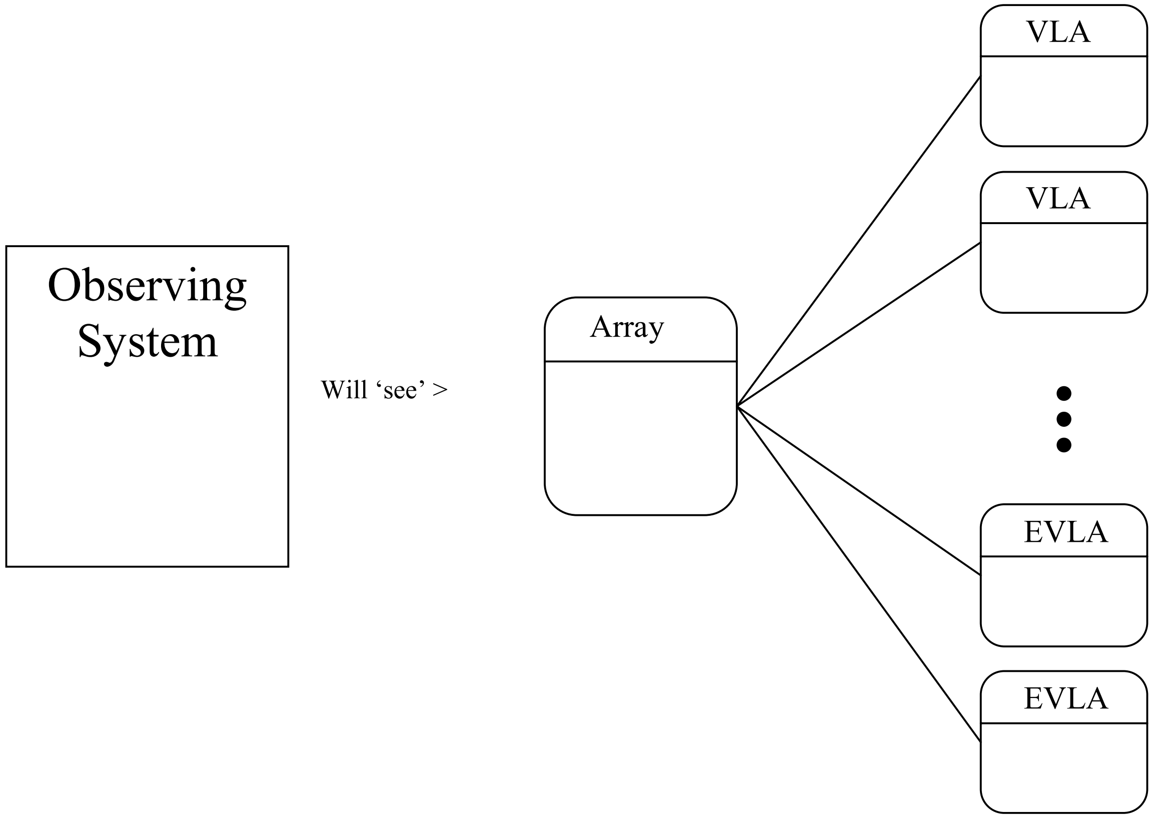
9.9.3.1 Transition Hardware Modules

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 will be designed and built so that the F14 module can be monitored and controlled by the EVLA monitor and control system.

9.9.3.1 Monitor and Control of VLA Antennas

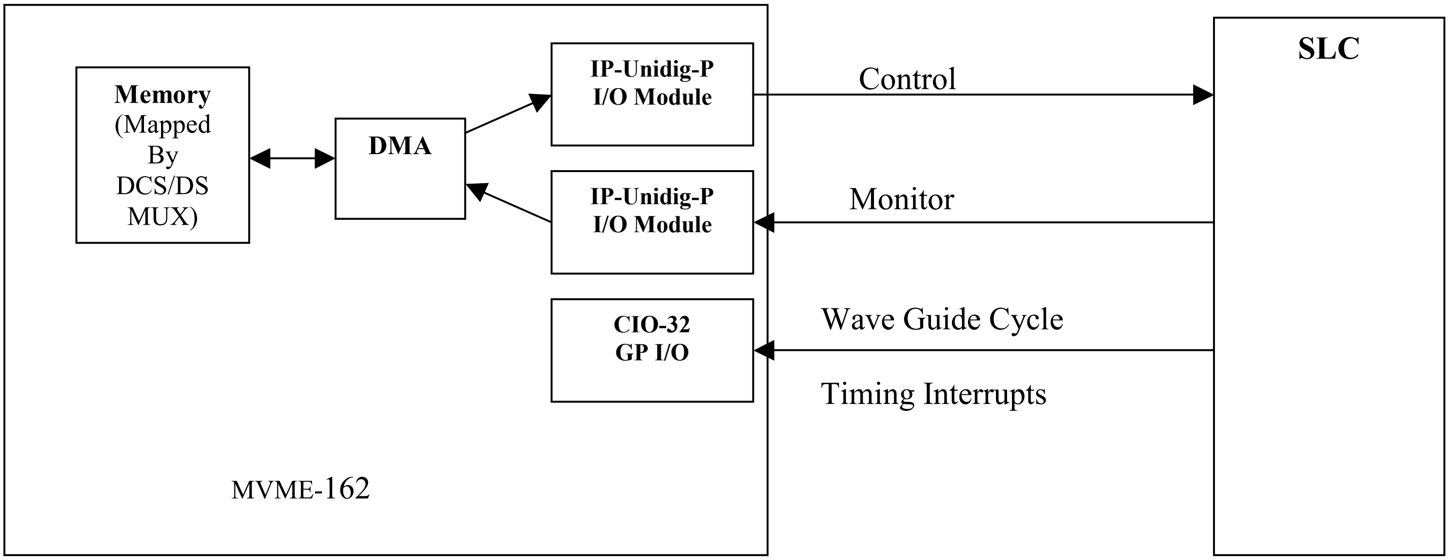
In order for the Observing System to operate both the EVLA and VLA antenna types, it should be able to ‘view’ the array simply as a collection of antenna devices of both the VLA and EVLA types. The Interim Control & Monitor Processor (CMP) is the means by which this view of the array will be developed for the VLA antennas.

All monitor data from the VLA antennas eventually converges upon a device known as the Serial Line Controller (SLC). All commands to the antennas are sent via the Serial Line Controller. Fortunately, when it was designed, the SLC was equipped with two computer ports, both of which have access to either all or any user-defined subset of antennas. The CMP is an interface to the second port of the SLC. It will provide access to the monitor data stream coming from VLA antennas, and the means to send commands to VLA antennas. With the addition of a second processor, it will also provide the means to present an object-oriented interface consisting of Antenna Device objects to the remainder of the EVLA monitor and control system.



The EVLA Antenna Device objects will be served by the EVLA AMCS and the VLA Antenna Device objects will be served by the Interim Control and Monitor Processor. The CMP is a Motorola MVME162 VME single board computer. It contains 4 slots for attaching Industry Pack (IP) daughter modules. Three IP modules will be used to interface the CMP to the VLA's Serial Line Controller.

The SLC takes commands from one of two processors connected to it. These commands contain address information so that they can be routed to the particular antenna and subcomponent for which they were meant. Monitor data from the antennas also contain this address information so they can be mapped to where they originated. All of the VLA antennas are controlled and monitored through this single SLC device.



The CMP will be connected to one of the two SLC computer ports. One IP-UniDig-P I/O Module with handshaking is used to send command data to the SLC; another is used to receive monitor data from it. A third IP module that provides general purpose I/O and interrupt capabilities will be used to coordinate data transfer timing with the VLA's Wave-guide cycle.

Another VME processor will be added to this chassis and will access the raw monitor and control data onboard the MVME162 via shared memory over the VME bus. This second processor will be a PowerPC, probably one of Motorola's MVME2700 family and will be used to convert the raw monitor and control data into VLA Antenna Device objects. The EVLA Observe System will then 'see' this collection of antenna device objects coming from the VLA just as it will the EVLA antennas.

This system has the advantage that with the exception of the SLC driver software in the MVME-162, almost all of the rest of the software will be the same as that used in the EVLA AMCS. Because this system will be available before the MIB hardware, AMCS software development and deployment can take place sooner.

9.9.3.1 Monitor and Control of the VLA Correlator

Both to replace obsolete, unsupportable hardware, and to assist in the transition phase by providing a network accessible controller, a replacement Correlator controller is being built. The new controller will be a VME based computer system designed to accept seamlessly 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 an EVLA monitor and control system. The VME system will consist of a Single Board Computer (SBC) with an Ethernet interface for command/control, a SCSI back end for data storage and/or archiving, and a separate array processor to receive Correlator integrator data and perform final processing. The new system will be installed into the system controller rack of the VLA Correlator.