EVLA Operation's System
Software Requirements Specification
(SRS)

(Initial Draft)

Revision History

Date	Version	Description	Author
09-Nov-2001	1.0	Initial draft	Rich Moeser

Table of Contents

1 Introduction

1.1 Purpose

The primary goal of this document is to provide a complete and accurate list of requirements for the EVLA Operation's System. Upon completion, the document will act as a binding contract between developers and users and will provide a common point of reference for system expectations.

The primary audience of this document includes, but is not limited to, project leaders, the designers and developers of the system and the end user. The document may also be of interest to EVLA project scientists and engineers or as a reference for individuals involved in similar projects with similar requirements.

Note: *Italicized text* indicates a requirement and text found between "<" and ">" indicates questions or comments to myself and/or readers.

< Throughout the document I refer to the system as the "Operation's System". Is that an acceptable name? >

1.2 Scope

The Operation's System is just one of many components that will make up the EVLA Computing System[1]. Its primary responsibility is to provide a suite of software tools that will allow the array operators to safely and reliably perform their daily tasks.

Some of the functions the Operation's System must provide are as follows:

- The ability to monitor the status and overall health of the array.
- The ability to control certain aspects of an antenna or the array.
- The ability to view and manage the operator/system log.
- The ability to view and manage observing schedules.
- The ability to view and manage users' system access and privileges.

This document will define only those requirements that must be fulfilled by the Operation's System.

1.3 Definitions, Acronyms, and Abbreviations

1.3.1 Definitions

Array – A collection of antennas. At times it will be used to indicate a subarray but will most often refer to the VLA, NMA or VLBA.
Observer – An individual granted time on the array to conduct a scientific investigation.
Operator – An individual authorized to issue commands to the array. In most instances, the term

[1] An overview of the EVLA Computing System can be obtained by reading the *System Requirements* and the *EVLA Architecture and Design* documents.

operator will refer to the array operator, an NRAO employee whose duty it is oversee the success and safety of an observation.
Operator station – This is the primary work area of an array operator and is located in the VLA Control Building. The operator's station consists of computers, terminals and communication equipment.
Administrator – An individual with unrestricted access to all aspects of the system.

1.3.2 Acronyms

AOC –Array Operations Center
API – Atmospheric phase interferometer
M&C – monitor and control
NMA – New Mexico Array
VLA – Very Large Array
EVLA – The VLA Expansion Project
VLBA – Very Long Baseline Array
IAT – International Atomic Time
LST – Local Sidereal Time
UTC – Coordinated Universal Time
RA – right ascension
WVR – Water Vapor Radiometer
SyRS – Refers to the *System Requirements* document.
SRS – Refers to the *Software Requirements Specification* document.

1.3.3 Abbreviations

Az – azimuth
Dec - declination
El - elevation

1.4 References

1) ANSI/IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications
2) ANSI/IEEE Std 1233-1996, IEEE Guide for Developing System Requirements Specifications
3) EVLA Memo No. 15, Scientific Requirements for the EVLA Real-Time System
4) EVLA Project Book
5) EVLA System Requirements (SyRS)
6) EVLA Architecture and Design
7) The Very Large Array Observing Log (J. Nieri, February 1994)

1.5 Overview

The remainder of this document contains a more detailed description of the Operation's System as well as the requirements necessary to design and build the system. Section 2 provides a general description of the Operation's System. Section 3 details the requirements of the product and is the core of this document.

The format of the document closely follows that outlined in the IEEE Std 830 document, IEEE Recommended Practice for Software Requirements Specifications.

2 Overall description

2.1 Product perspective

The EVLA Computing System < or M&C System? > will be designed and implemented as a distributed system. Computers in the system may be of different types and operating systems and will communicate with one another over a network. Currently, only a conceptual diagram exists for the system and should be viewed as such[2]. Referring to the diagram, it's likely the Operation's System software will divide its responsibilities between the Operation's Server and the computing stations scattered throughout the diagram.

< I'd like to include Bill's "EVLA Computing System Conceptual Diagram" or "EVLA M&C System Strawman Diagram" here. In the end will it be the "Computing System" or the "M&C System"? >

2.1.1 Operation's Server

< TBD >

2.2 Product functionality

2.2.1 Array Monitoring

Operators will monitor the array using a variety of supplied user interface displays. The displays will use both textual and graphical formats for displaying the data and will be used to monitor the overall health of the array as well as a single device on a particular antenna. Displays will likely be provided for weather, individual antennas, antenna subsystems, subarrays and the like. The number and exact types of displays required is unknown at this time.

2.2.2 Array Control

As with monitoring the array, operators will control the array using the supplied user interface displays. Control functionality will be available through displays dedicated to control and through monitoring displays with built-in control functionality. The number and exact types of required control displays is unknown at this time.

2.2.3 Scheduling

< TBD >

2.2.4 User Management

< TBD >

[2] See Chapter 9, Figure 1 of the EVLA Project Book for a conceptual diagram of the EVLA Computing System.

2.3 User characteristics

2.3.1 Array Operator

The array operator will be the primary user of the system. As such, the majority of the requirements contained in this document directly reflect their needs and duties. An array operator must go through three months of training before operating the array on a solo shift.

2.3.2 Engineers and Technicians

These are highly skilled individuals with specialized areas of expertise. They will likely be interested in only a few devices in the system and are called upon to troubleshoot and test malfunctioning or non-working device(s). The engineers and technicians will need tools to inspect individual devices both from remote locations and at the antenna.

2.3.3 Astronomer/Scientist

These individuals are primarily interested in the science that is obtained from the instrument. It's unclear at this time how much the scientists will directly interact with the instrument during their observations and what level of control they will have over the instrument.

2.3.4 Software developer

These individuals are responsible for developing the software and will interact with the system to ensure that it is functioning properly. The software developer requires remote access to the system so that troubleshooting the system can be accomplished away from the VLA and during non-working hours.

2.3.5 Web User

Most individuals of this type will be eavesdropping on the system as interested bystanders. They will only have the ability to monitor the system and will not be allowed to control any aspect of it. A few authorized individuals, however, will be authorized to access parts of the system that are usually considered restricted.

2.4 Constraints

< TBD >

2.5 Assumptions

< TBD >

2.6 Apportioning of Requirements

This subsection describes several features that have the potential to offer many benefits, but may not be considered critical at this time. If the possibility exists that any one of these features could be added to the system at a later date, it would seem appropriate to generate the requirements for those features now. This would allow the "hooks" for such features to be designed into the system, simplifying the addition of the features to the system at a later date.

2.6.1 Simulator

The ability to simulate real-time situations without affecting the online system would be highly beneficial for the training of array operators and for testing software.

2.6.2 Display Builder

If a decision were made to incorporate a display builder into the system, it would be logical to build it early in the development process so that all M&C displays would be built from such a tool. A display builder would not only allow the standard package of displays to be built but would also allow users to create displays based on their own needs and without support from a software developer.

Users could construct displays from standard components such as buttons, radio buttons and sliders and components built by onsite developers and third party component providers. The user would have the ability to save the display and load it so that it could be used immediately.

2.6.3 Scripting

< TBD >
< Do we want to support any scripting languages (e.g, Python, Tcl, BeanShell, etc.)?

3 Specific requirements

3.1 Hardware

The manufacturer and type of workstations to be used by the array operators to monitor and control the array are unknown at this time < The EVLA SyRS document specifies, however, that Sun/Solaris based workstations will be used. Is this firm? >. In the interest of minimizing the cost of the workstations and without unnecessarily limiting the available hardware options - now and in the future - the software must be reasonably platform independent so that it will run on a variety of machines with minimal impact to the software. This would allow a change in the underlying hardware without requiring a significant rewrite of the software. Instead, it would only require changes to a configuration file or to a small layer of the software system.

At a minimum, the system shall be compatible with the following hardware platforms:

1) *Sun*
2) *Commodity PCs*
3) *Macs*

3.1.1 Prototypical Operator's Workstation

The operator workstation is the computing system designated to run the software used by the array operators to monitor and control the array. The workstation(s) will be located in the control room of the VLA Control Building.

The EVLA System Requirements Specification (SyRS) indicates that the array operators will have Sun workstations running the Solaris operating system. The number of workstations and/or terminals required is undecided at this time and is somewhat dependent on the working style and personality of the operator. However, at a very minimum, the operator should have two workstations, each with access to two or more terminals. One station would be used for array monitor and control tasks and the other would be used for logging, email and other non-M&C activities. Also, in the event that one of the systems becomes unusable the working system can be used as a backup system to monitor and control the array.

The specifications provided below are to be used as a guide in selecting such a system and does not target any particular manufacturer or system model.

- 2GHz processor
- 1 GB RAM
- 80 – 100 GB hard drive
- CD-RW drive
- DVD-ROM drive
- 1.44 MB diskette drive
- Wireless keyboard and mouse
- 19" Digital flat panel display (or a multi-screen LCD system, e.g., http://www.9xmedia.com/Pages/Products.html or http://www.massedi.com/mass_multiples/c3h18_horizontal.htm)

3.1.2 Control Room Configuration

< TBD >

3.2 Operating Systems

Since the software must run on several popular computing platforms, it must also have the ability to run on a number of operating systems supporting that hardware. Again, a change in operating systems should have minimal impact on the software system.

At a minimum, the system shall be compatible with the following operating systems:
1) *Solaris SPARC/x86 (Solaris 8 or greater)*
2) *Linux x86 (RedHat 7 or greater)*
3) *Microsoft Windows (98, ME, 2000, XP or greater)*
4) *Mac OS X (or greater)*

3.3 Security

The software system needs 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.

This mechanism must be built into the system from the start allowing security issues to be properly dealt with as they crop up throughout the entire design and development process. An attempt to add security to the system after the fact would most likely prove to be a costly and difficult task.

All users of the system must be uniquely identified. This could be done by using a username and associated password scheme that would authenticate and authorize the user access to the system and, if applicable, grant the user access to restricted or controlled parts of the system. If a user cannot be identified, she will be given "anonymous" access with read-only capabilities. In order to monitor all past access to the system, all attempts to access the system should be logged.

Users' needs and expectations from the system will be different. The operators, for instance, should be given unrestricted access to all aspects of the system and should have the authority to grant and revoke privileges on a per-user basis. Remote observers and array support personnel, on the other hand, require access to some parts of the system, but not all, indicating that an access level is needed that allows privileges to be granted on a per-user and what-do-you-need-to-do basis. Lastly, an access level for the general user should exist that allows the user read-only access to the system, restricting the user from all types of array control or data entry.

Below is a preliminary list of security related requirements.

All users of the system shall login using some form of unique identification (e.g., username and password).

A default user identifier shall exist that allows anonymous and read-only access to the system (e.g., "guest").

All login attempts shall be done so in a secure manner (e.g., encrypted passwords).

A system administrator shall have unrestricted access to all aspects of the system.

Each user shall have a set of system access properties that defines the user's privileges within the system (e.g., the subsystems a user may control or system tools the user may access).

The administrator shall have the ability to do the following:

1) *create and add a new user to the system*
2) *remove a user from the system*
3) *edit and modify a user's system access properties*
4) *block all commands sent to the array by all users or selectively by user.(All users will remain logged into the system and will not be allowed to issue commands to any part of the array until the command block is released. The administrator will still have the ability to issue commands to the array.)*
5) *block all access to the system for all users or selectively by user. (All blocked users with active sessions will automatically be logged off.)*

< Should remote users have the ability to create and add their own user ID? >

3.4 Remote Observing

The primary goal of the software is to provide the array operators with the tools to safely, effectively and efficiently monitor and control the array. In meeting this goal the system designers must also build into the system a secondary goal that allows access to the system independent of location using the same code base with minimal alterations to the software.

For the purpose of the EVLA, remote observing will be defined as the act of viewing, or interacting with, any aspect of the array from a computer other than an operator's station inside the VLA Control Building or from a technician's computer at the antenna. All other locations will be considered remote connections including sessions established from the AOC in Socorro or from support buildings at the VLA site.

The ability to remotely observe does not imply that the software must be web-based and that it run within the confines of, or is launched from, a browser. It only suggests that the software be accessible via the internet and that it provide the same functionality (or subset thereof), remotely, utilizing the same code base as the user interface software used by the array operator at the VLA Control Building.

As with security, remote observing must be an integral part of the system and must be considered throughout the entire design and development process.

There are several reasons for remote observing.

1) Observers can monitor the progress of their observing program and make or request changes during their observation to increase the quality of data.
2) Hardware and software engineers will have the ability to access the system from remote locations during working and non-working hours to do first-order problem solving.
3) In the future, operators may be stationed at the AOC in Socorro.
4) Other instruments in astronomy offer such a feature and users come to expect it.
5) It's doable!

Remote observing should in no way interfere with the accurate transmission or reception of data to and from the array. And any requests or commands from an operator workstation should take precedence over all remote requests for data or issued commands.

At times, it may be necessary for the array operator to have some form of communications link with a remote observer other than telephone and/or email. A simple solution would be to create a "chat" program that allows the operator and observer to exchange messages in near real-time.

< Peggy doesn't feel that observers should have any control of the instrument. However, if they are allowed any control capabilities, it should be done so through observe file manipulation rather than commands issued directly to the instrument. >

Below is a preliminary set of remote observing requirements.

The software shall be accessible via the internet (downloadable, running within a browser, or launched from a browser, etc.) using a single code base.

A request for data or commands submitted by the operator or from an operator's station shall take precedence over requests and submissions by non-operators.

The system displays shall be capable of, at a minimum, a two second data refresh rate over a low-bandwidth connection.
< This should be the slowest update rate. The operator stations and technician stations must be capable of much greater refresh rates, possibly 10Hz. >

*The system shall support a maximum of **xxx** concurrent user sessions.*
< Peggy says a maximum of 15 users. I would at least double that. >

A remote user with authorized privileges shall have the ability to communicate with the operator using a simple messaging system.

3.5 Installation and Upgrades

There's no doubt that users will want and expect to have the latest version of the software available to them. The user's role and level of involvement in the installation and upgrade process must be determined early in the design phase as it will affect both the design of the product and the technologies used.

Several options exist for distributing the software, each requiring various levels of user involvement. For example, a one hundred percent web-based solution may provide the user with the latest software features without the user having to install anything, whereas, a more active user role would be assumed if the user has to navigate to a web site, download the software bundle and install it. If the user will be required to partake in any aspect of the installation process, the software should be designed such that the process of upgrading is automated as much as possible and it should guide the user through the entire installation/upgrade of the product.

Below is a preliminary list of installation and upgrading requirements.

The system shall allow (remote) users access to the latest version of the software.

< Upgrades to remotely running software shall be automatically detected, downloaded and installed. (maybe?) >

3.6 General User Interface Requirements

The system's user interface should be intuitive, easy to use and provide an overall positive user experience. It should do what the user expects it to do, inform the user of its current state, and when something goes wrong it should explain the problem in a language natural to the user and provide guidance toward correcting the problem. It should also conform to a specified set of user interface guidelines to foster consistency between different tools within the application.

An intuitive and easy to use application will undoubtedly decrease the amount of time it currently takes to train an operator. The operator should only be concerned with what they need to do with the tool to perform the task at hand rather than focusing on how to use the tool. As one of the array operators pointed out, "We're operating an array, not a computing system."

Below is a list of requirements that will apply to all user interface components. Many of the requirements are obvious and are expected from modern software systems.

The system shall allow the user to display multiple windows simultaneously.

Each display shall reflect the level of access and the privileges of the user (security aware).

The user shall have the ability to view the last displayed page (forward and back).
< What is the limit for a page history buffer? Is it configurable? >

The user shall have the ability to define and view a default or "home" page.

If the user requests a new page after editing the current page, the system shall prompt the user to save the changes. The user shall have the option to save or ignore the changes. The system will display the new window. <more like a use case!>

If the user requests to close a window or exit the system after editing the current page, the system shall prompt the user to save the changes. The user shall have the option to save or ignore the changes. The system will dispose of the window. <more like a use case!>

Time shall always be available and displayed in the following formats:
1) *Coordinated Universal Time (UTC)*
2) *VLA local sidereal time (VLA LST)*
3) *International Atomic Time (IAT)*
4) *local time*

The system shall use Coordinated Universal Time (UTC) as the default time format.

The user shall have the ability to copy (or cut) text to the system "clipboard" and paste the text to other text-accepting components.

The system shall adhere to a, as yet unspecified, set of user interface design guidelines.

Where applicable, the user shall have the ability to select the following preferences:
1) *Color*
2) *Font*
3) *< others? >*

The system shall allow for "Undo" and "Redo". <Do we want this? If yes, it needs to be done from the start. >

The system shall be internationalized. <Do we want this? If yes, it needs to be done from the start. >

<Built in accessibility for assistive technologies (screen readers, screen magnifiers, speech recognition, etc.)? May be useful for Wye-Mon.>

All user interface pages shall have the ability to be displayed solely in a user interface window or combined with other components in a window.

3.7 System Login

A means of logging on to the system will be required. Before a user can interact with the system they must first supply their system identification information (e.g., username and password) to the system. Using this information, the system will authenticate the user and either grant or deny the user access to the system.

Below is a preliminary list of requirements for logging-in to the system.

The user shall have the ability to login to the system.

The system shall allow the user to enter and submit his or her unique identifier (e.g., username and password) to the system for authentication.

The system shall not echo password characters to the display. Instead, the system will replace the typed characters with asterisks.

The user shall have the ability to discontinue or cancel a login attempt at any time during the login process.

The login page shall display the name, version and date of release of the application and provide the user with information about the application.

The system shall log, at a minimum, the following login and end of session information to a persistent store (db or file):
1) *the time the session was established and ended*
2) *the user identifier*
3) *the status of the login attempt (login only)*
4) *the duration of the session (end of session only)*
5) *the cause of session termination (user, administrator, error, etc.)*

After 3 unsuccessful login attempts the system shall login the user as an anonymous user with read-only privileges.

3.8 Project

The project, also referred to as the program, observation or experiment, is the core of the business model. It defines what is being observed and the resources required by the project. It's unclear whether this object should be defined in this document, but it's clear that a need for such an object and its data exists.

The user shall have the ability to create a project.

The user shall have the ability to modify a project.

The user shall have the ability to delete a project.

At a minimum, the system shall define the following project properties:
1) *project code*
2) *principal investigator*

3) *title*
4) *description*
5) *astronomical source(s)*
6) *type(s) (phased array, pulsar, continuum, etc.)*
7) *array configuration (antennas)*
8) *observe script*
9) band(s)
10) < others? >

3.9 Scheduling

Scheduling is the act of specifying when a project will take place. Three scheduling modes are anticipated for the EVLA: fixed, dynamic, and goal-oriented. All three modes serve the same general purpose, reserving a block of time and set of antennas for a specified project. The primary difference between the three modes is when the act of scheduling takes place and how the scheduling is accomplished.

3.9.1 General

These requirements apply to all scheduling modes.

The user shall have the ability to add a project to the observing schedule.

The user shall have the ability to remove a project from the observing schedule.

The user shall have the ability to modify a project's schedule.

The user shall have the ability to view the observing schedule for a specified date/time range.

The system shall have access to VLA, NMA and VLBA schedules. < See Section 2 "VLA/PT Link, future real-time VLBA" of the Scientific Requirements document. >

3.9.2 Fixed

Fixed scheduling reserves a block of time for a specified collection of antennas and is often performed weeks or even months in advance of the actual observation. Most of the current VLA scheduling is done this way.

3.9.3 Dynamic

Dynamic scheduling allows a project to be scheduled when the weather and "seeing" conditions are optimal for that project. This is done by intentionally leaving unscheduled time gaps between scheduled projects in which dynamic projects can be inserted. The dynamic scheduling algorithm will require input many different sources; the VLA Weather Station, the Atmospheric Phase Interferometer, ionospheric data from the VLA GPS System and the RFI data archive.

All times for dynamic scheduling shall be specified in VLA LST. < See Section 3.2.3 of the Scientific Requirements document. >

The user shall have the ability to dynamically schedule a project or test.

3.9.4 Goal-Oriented

< TBD >

3.9.5 Observe Files

The operators will need the ability to view and manipulate the project at the scan-level. The operator should be aware of the current scan and have information regarding that scan as well as the ability to manipulate certain aspects of the scan such as overriding the amount of time spent on a particular scan and the order of the scan, all in real-time.

3.9.5.1 General

Observe files shall be archived. < See Section 3.1.5 of the Scientific Requirements document. >

3.9.5.2 Generation

The system shall be compatible with the current OBSERVE/JOBSERVE control script files. < See Section 1 of the Scientific Requirements document. >

The system shall expect scan stop times to be in VLA LST. < See Section 3.2.3 of the Scientific Requirements document. >

3.9.5.3 Real-time Viewing/Editing

< How and what should the operators manipulate in the observe file? >

The user shall have the ability to view the individual scans of a project.

At a minimum, the user shall have access to the following scan properties:
1) end time (in VLA LST)
2) source
3) position
4) frequency
5) equipment configuration
6) subarray affiliation
7) mode(s)
8) quality of calibrator

The user shall have the ability to reorder the scans.

The user shall have the ability to extend on a source or scan:
1) up until a specified time
2) or for a specified duration

The user shall have the ability to set the end time for a specific source or scan (overriding the end time in the observation script file).

The system shall have the ability to skip a source for the following:
1) at a specified time
2) for a specified duration

3.10 Warning and Error Message Reporting

Warning and error conditions are meant to alert users of problems in the system that require immediate attention or less serious conditions, that, if left unacknowledged, could have the potential to develop into more serious problems. Conditions that warrant warnings or errors include, but are not limited to, loss of communication with a subsystem, device errors, out of range monitor points, invalid device commands, network failures and software errors. The messages should accurately describe the problem and its consequences and provide access to the source (or cause) of the problem as well as a list of possible solutions to the problem.

The system should make every attempt to limit warning and error conditions to only those relevant to the current system activity. All others must be suppressed or ignored. It should not be the responsibility of the array operator to visually weed through a list of faults to determine the actual problem.

< What should I call these? Warning and error messages? Fault conditions? Alarms? Other? >

Below is a list of preliminary requirements for the display of fault conditions.

At a minimum, the system shall provide access to the following fault condition properties:
1) *the date/time of the message*
2) *origin of the message (computer, subsystem, etc.)*
3) *the message type (WARNING or ERROR, etc.)*
4) *the message code*
5) *a terse description of the problem*
6) *a severity level*
7) *a detailed description of the problem*
8) *a detailed description of possible or likely consequences*
9) *a detailed description of possible or likely causes*
10) *a detailed description of corrective actions*
11) *a link to relevant or helpful documents*

The terse description of the message shall be limited to 50 characters. The detailed message will allow ***xxx*** *characters, as will the descriptions for the consequences, the cause and actions.*

The system shall allow messages to be color-coded based on the type and severity level of the message.

The user shall have the ability to specify the color of the message based on the type and severity level of the message. < Once this is determined, it's unlikely to change. Keep? >

The system shall provide audible alerts for fault conditions based on the type and severity level of the message.

The user shall have the ability to specify the audible alert of a message based on the type and severity level of the message. < Once this is determined, it's unlikely to change. Keep? >

The user shall have the ability to create and enter a known solution to a problem and submit the solution to a persistent store for future reference.

The user shall have the ability to suppress fault conditions for a specified period of time (minutes, hours, days, etc.) or event (source change). <risky? >

The user shall have the ability to view the fault conditions for the following groups:
1) *a single antenna*
2) all antennas
3) a subarray

The user shall have the ability to plot the frequency of warning and error messages over a specified time range.

The user shall have the ability to redirect warning or error messages to:
1) *the operator (or system) log*
2) *a printer*

Below is a list of requirements for viewing archived or historical warning or error messages.

The user shall have the ability to retrieve messages by time range, message code, type, severity level or error message. < any property? >

The user shall have the ability to view and sort the messages by time, code, type, source (subsystem) and severity level. < any property? >

3.11 Log facility (operator or system log)

The current VLA operator log contains information concerning an observer's project and is intended to alert the observer of conditions or events that occurred during the project that may have affected the collection or quality of the data. The operator logs are project based, meaning that a new log is created and exists for each project rather than a continuous entry of, possibly unrelated, time-stamped events. At the completion of a project the observer is e-mailed a copy of the log.

The decisions of when to make a log entry and the content of the entry are not cut and dry. A general rule of thumb for entering a log is – if something happens that may affect the data – log it. The VLA operators enter general information into the log that describes the project (e.g., the observing script name, project code, the observers e-mail address, information on the source), weather conditions (e.g., wind direction and speed, temperature, dew point, barometric pressure and cloud type) and comments they feel may be useful to the observer.

The logging facility should be designed as a distributed tool that can be easily embedded into other applications. There has been discussion that logging tools with similar functionality are needed at other sites for other instruments. The logging facility should thus be designed and developed with this in mind so that the tool is as general purpose as possible with preferential treatment given to the EVLA.

Below is a list of preliminary requirements for the operator's (or system log).
< Peggy feels observers should not be able to submit entries to the log. >

(In all cases below, the phrase "The user" can be replaced with "An authorized user".)
General:

3.11.1 General requirements

The system shall save log entries to a persistent store.

The user shall have the ability to query the persistent store (by time range, message, type, program, etc.).

The logging system must be distributed and platform independent.

3.11.2 Log Entries

A log entry is any user or system generated message that will be submitted to the operator/system log. Users should have the ability to create and submit log entries as well as modify and delete them.

The user shall have the ability to create a log entry.

The system shall automatically provide the event timestamp at the moment the entry is created.

The user shall have the ability to select the type of entry (operator note, operator to operator message, antenna visit, observers note, weather, etc.).

The user shall have the ability to edit a log entry.

The user shall have the ability to delete a log entry.

The user shall have the ability to save or submit a log entry.

The system shall automatically timestamp the entry on submission.

3.11.3 Reminders

A reminder is simply a mechanism that allows the user to generate a message and have that message displayed at a specified time. This feature does not necessarily have to be a part the logging system and could easily be promoted to an overall system feature.

The user shall have the ability to create a reminder.

The user shall have the ability to specify all properties of the reminder (the time it will "go off", the message to display, etc.)

The user shall have the ability to enter the reminder trigger time in any of the time formats:
1) *UTC*
2) *IAT*
3) *VLA LST*
4) *local time*

The user shall have the ability to edit a reminder.

The user shall have the ability to delete a reminder.

The user shall have the ability to view all scheduled reminders.

The user shall have the ability to specify the frequency of the reminder (once, every hour, daily, monthly, etc.)

The system shall automatically delete all triggered and acknowledged nonrecurring reminders.

The user shall have the ability to link a reminder to an event (program change, source change, parameter value).

The user shall have the ability to specify the method of notification: popup message (default), beeping, flashing, email.

The user shall have the ability to postpone a reminder.

3.11.4 Scheduled tasks

A scheduled task is similar in functionality to the reminders, however, rather than displaying a message, a scheduled task would instead launch a script (or executable) at a specified time. This feature could be used to enter weather data on an hourly basis into the operator/system log.

The user shall have the ability to create a scheduled task.

The user shall have the ability to edit a scheduled task.

The user shall have the ability to delete a scheduled task.

The user shall have the ability to postpone a scheduled task.

The user shall have the ability to view upcoming scheduled tasks.

3.11.5 Macros

The ability to record and save a macro would allow the user to create predefined entries that could be linked to user interface buttons. The current VLA operator log uses Excel macros to provide this functionality.

The user shall have the ability to record a macro.

The user shall have the ability to save a macro.

The user shall have the ability to name a macro.

The user shall have the ability to delete a macro.

The user shall have the ability to run a macro.

The user shall have the ability to link a macro to a user interface component (button or menu component) or keyboard shortcut.

3.11.6 Maintenance database

This feature would allow the user limited access to the maintenance database (MainSaver). Simple queries could be performed so that users could locate maintenance database entries for known problems that are potentially causing or having some impact on an observation. When the user finds the database entry he or she should be able to link that entry to a log entry.

The system shall have the ability to search the maintenance database.

The system shall have the ability to link maintenance database entries to log entries.

3.11.7 Reporting

The user should have the ability to create a number of reports from the data contained in the log. A standard package of reports should be available to the user and possibly in the future a report building feature would be added to the system.

The user shall have the ability to create a report for a specified project.

The system shall have the ability to compute downtime related to a particular problem.

3.14 Parameters (monitor points)

A parameter can be defined as an autonomous piece of data that is either raw or derived. Raw parameters are passed through the system unaltered whereas derived parameters are generated from within the system from other raw or derived parameters in the system.

Each parameter should have an ID, description, value, units, etc. A user should have access to all the information concerning a parameter.

The user shall be able to view the following parameter properties:
1) *date/time stamp*
2) *current value*
3) *engineering units*
4) *description (terse and detailed)*
5) *type (raw, derived, etc.)*
6) *data type (binary, decimal, hexadecimal, octal, logical, floating point, or integer)*

The user shall be able to plot a parameter over a specified time range.

The user shall be able to list a parameter's values over a specified time range.

3.15 Monitor and Control Displays

The monitor and control displays will allow users to visualize the current state of the system at varying levels of detail as well as provide the required control capabilities. The number and details of the M&C displays to be generated is unknown at this time, however, the need for specific types of displays and the basic characteristics of those displays can be addressed. The displays can be categorized in terms of the content they are to present.

The system monitor and control displays shall update the parameter value whenever the parameter's value changes or at the specified update rate for that parameter.

3.15.1 Weather Displays

3.15.1.1 General

Displays of this type will provide current and historical weather information for all sites and will alert the user of extreme and potentially hazardous weather conditions.

Each weather page shall display the following VLA weather station data:
1) *wind direction*
2) *wind speed*
3) *current temperature*
4) *dew point*
5) *barometric pressure*

The user shall have the ability to plot the history of any of the above weather data.

The user shall have the ability to submit current weather data to the log.

3.15.1.2 Atmospheric Phase Interferometer

< Need to extract requirements for this from http://www.aoc.nrao.edu/vla/html/PhaseMonitor/phasemon.html >

3.15.1.3 Ionospheric

The information for this display will be obtained from the data generated by the, currently undefined, EVLA GPS system. The display must provide access to both real-time and archival ionospheric data.

The user shall have the ability to display real-time ionospheric data.

The user shall have the ability to display archival ionospheric data for a user-specified time range.

3.15.2 Antenna Displays

Displays of this type will provide information on the overall health of a single antenna and the components that comprise the antenna. It will also allow authorized users to control the antenna and subsystems connected to the antenna.

Each page shall display the following:
1) *antenna number*
2) *DCS number*
3) *station or pad ID*
4) *subarray*
5) *azimuth and elevation*
6) *current source*
7) *RA and Dec of current source*
8) *pointing errors*
9) *cable wrap*
10) *weather data (anemometers 1 & 2)*
11) *stop time of observation*
12) *affiliated project*
13) *observation title (observation file name)*

14) slew rate (azimuth and elevation)

Authorized users shall have the following antenna control capabilities from this display:
1) park
2) stow
3) point
4) "avoid snow"
5) "tip for snow"

3.15.3 Antenna Subsystem Displays

Displays of this type will supply detailed information on antenna subsystems such as focus/rotation, cryogenics, front-end, LO system, and power supplies. These displays will be of particular interest to engineers and technicians responsible for those systems and will likely be the displays they will use to access information remotely as well as at the antenna. The displays will also include the ability to control the relevant devices for users authorized to do so.

< TBD >

3.15.4 Subarray Displays

Displays of this type will provide information on the overall health of the subarray and will allow quick access to the antenna data page for antennas within the array. It will also allow authorized users to issue commands to the group of antennas contained in the subarray.

Each page shall display the following:
1) affiliated project
2) current source
3) current position
4) antenna(s)
5) mode(s)
6) < others? >

Each page shall provide access to the antenna data page for each antenna in the subarray.

Authorized users shall have the following subarray control capabilities from this display:
1) park
2) stow
3) point
4) "avoid snow"
5) "tip for snow"

3.15.5 Control display(s)

Displays of this type will allow the operator to control the most common elements of the antenna or array from a single display.

< TBD >

3.15.6 Emergency Shutdown Display(s)

A display of this type will be necessary and will be similar in functionality to the "wye-mon" display which is part of the current VLA observing system.

The user shall have the ability to shutdown the following components in the event of an emergency:
1) *entire array < will this include VLBA and NMA antennas? >*
2) *array arm*
3) *subarray*
4) *single antenna*

The system shall have the ability to monitor and reset the following system components:
1) *array controller*
2) *antenna controllers*
3) *non-critical power*
4) *correlator*

The system shall have the ability to monitor and reset the following support equipment:
1) *computers*
2) *generators*
3) *UPS*

Authorized users shall have the following array control capabilities from this display:
1) *park the array*
2) *stow the array*
3) *point the array*
4) *"avoid snow"*
5) *"tip for snow"*

3.15.7 RFI Display(s)

< TBD >

3.15.8 Quality of Data Display(s)

< display data from the correlator and image pipeline? >
< TBD >

3.15.9 Correlator Display(s)

< will this be built by the correlator folks or will we supply this?>
< TBD >

3.15.10 Parameter Watch Display

A display of this type will allow users to quickly look up a parameter and add it to a list of displayed parameters.

3.15.11 Project Details Display

This display page would not necessarily be considered a monitor and control page, however, one would expect that information related to a project is readily available from other M&C displays.

3.16 Interaction With Other Systems

3.16.1 Image Pipeline

< TBD >

3.16.2 Distribution System

< TBD >

3.16.3 Data Archive

< TBD >

3.16.4 AIPS++

< TBD >

3.17 Online Help

A help facility should be incorporated into the design of the software system from the start and it should allow the array operator to find the information she seeks quickly and accurately. At a minimum, it should include a table of contents, an index, full-text search and context sensitive help.

Below is a preliminary set of requirements for online help.

The system shall provide an online help facility with the following features:

1) *table of contents*
2) *index*
3) *full-text search*
4) *context sensitive help*

3.18 Documentation

The system shall provide the following online documentation for all tools associated with the system:

1) *user manual*
2) *software design documents*

3.19 Printing

Users of the system will need the ability to capture and print screens as well as the ability to create and print reports.

Below is a preliminary list or requirements for printing.

The user shall have the ability to capture and print any display.

3.20 Plotting

A general-purpose plotting component will be needed. Several of the standard M&C displays will likely contain plots of parameters relevant to that display and a user must have the ability to plot any system parameter on the fly.

Below is a preliminary set of requirements for a plot.

The user shall be able to generate the following plot types:
1) *scatter plot*
2) *histogram*
3) *<others ?>*

The user shall be able to view the plot dynamically (real-time) or statically (offline).

The user shall have the ability to give the plot a title.

The user shall have the ability to define the plot axis labels in the following manner:
1) *user defined character string*
2) *import predefined labels from a parameter definition database*

The user shall have the ability to specify the number of major and minor tick marks for a plot.

The user shall have the ability to connect plotted points by lines.

The user shall have the ability to view multiple parameters on the same axis.

The user shall have the ability to specify whether the major and minor tick marks are:
1) *linear*
2) *logarithmic*

The user shall have the ability to auto-scale plots.

The user shall have the ability to specify the scale of the plot for all axes.

The user shall be able to plot any parameter in the system.

The user shall be able to plot the same parameter for all antennas or selected antennas.

The user shall have the ability to infinitely zoom in and out of the plot.

The user shall be able to select various point styles:
1) *none*
2) *dots*
3) *points*
4) *shapes (triangles, squares, circles, etc.)*

The user shall be able to show error bars.

The user shall have the ability to view the legend for the plot.

The user shall be able to print a hardcopy of the plot.

The user shall be able to save the plot (to a file?).

The system shall identify the following characteristics of a parameter over a specified time range:
1) *minimum value*
2) *maximum value*
3) *average*
4) *RMS*

The user shall have the ability to set maximum and minimum limit markers.

4 Non-Functional Requirements

< Some of the topics and requirements specified in section 3 will be moved to this section. >