

EVLA Project Book Chapter 9

System Overview and Integration - Software

Bryan Butler

2004-Dec-08

1 Introduction

In order to achieve the full potential of the hardware being installed for the EVLA, software must be in place to operate the instrument properly and to enable users and staff to productively use the telescope. While the VLA software, both that used to operate the instrument and peripheral software used by astronomers, engineers, operators, and others, has performed admirably, it is clearly in need of modernization. Much of the material in this chapter is taken from Butler et al. (2004a) and Morgan et al. (2004).

2 General Design Principles and Methodology

The primary goals of the EVLA software are, in order of importance:

- maximize the scientific return of the EVLA;
- provide ease of use, for both novices and experts;
- exploit commonality amongst all NRAO telescopes where possible.

The methodology employed to achieve these goals is a mix of traditional and more modern, agile software development techniques. Astronomers are expected to come up with detailed requirements and use cases, which are then used to drive a high level design and architecture. This is refined into functional requirements and more detailed design, where the higher level functions (subsystems) are defined in much more detail, and communication between the subsystems is more clearly organized. The higher level design, as well as the detailed design of the subsystems, is achieved by making “teams” to attack the problem. Each team is comprised of the team members (generally a small number - three or four), a team leader (one of the team members), a team manager, and any number of team consultants. Each team has a specific charge, and a set of deadlines and milestones to achieve. Most teams are intended to be dissolved at some point, when their work is done. The team members are drawn from the available pool of EVLA software designers, a relatively small but very competent group. The “users” of the software are generally considered to be the astronomers, and all teams have at least one astronomer assigned as a consultant. In this way, the users are embedded directly in the design process, and are available for direct and nearly immediate consultation.

3 Requirements and Use Cases

The design of the software flows from the requirements and use cases, which are meant to be written by astronomers, engineers, or operations staff. For the specific case of the EVLA, the requirements documents are in the following areas:

- real-time system (Butler et al. 2004b);
- e2e (Butler et al. 2003a);
- post-processing (Myers et al. 2003);

- operations (Perley et al. 2003);
- engineering (Butler et al. 2003b).

All of these documents are intended to be updated as needed, by the appropriate groups of people (the authors, mostly).

Use cases are being written by members of the scientific staff, and incorporated in the design as needed. In general, use cases are produced as requested by the design teams, when needed. As described above, the staff scientists are intimately involved with the teams, with direct and frequent contact encouraged, in order to best facilitate the flow of communication between the two groups (astronomers and programmers).

4 Common Software for NRAO Telescopes

It is desirable to have the software at the various NRAO telescopes be as similar as possible. At the highest level, the interfaces into the instruments should have a similar look-and-feel, and similar functionality as far as possible. In this way, an astronomer using the EVLA will be comfortable using ALMA or the GBT, since the interfaces will be similar. The users of NRAO telescopes have been asking for this for many years. At a lower level, having similar software makes it easier for NRAO as an observatory to maintain and extend the capabilities of all of its telescopes, since software expertise which is spread across many sites can be brought to bear on such efforts. Of course it makes no sense to blindly copy software that might not be appropriate from one instrument to another. The software must at a minimum satisfy the requirements for the instrument. In situations where it might make sense to copy software, a cost-benefit analysis must be done, to determine whether the software re-use is beneficial. In addition, a decision must be made on what level to copy the software: high-level design, low-level design, actual code? Some software must not only operate a new telescope, but must also continue to operate an existing telescope (this is the case for GBT, VLBA, and EVLA, but not the case for ALMA), and this may place additional constraints on software sharing across the projects. However, it is clear that it is advantageous overall to have a common high level design across NRAO telescopes, and to share as much software as sensible between the various projects. A committee has been formed at NRAO to address exactly this issue, with representation from the four main projects, as well as from the general expertise at the observatory. This committee, the NRAO e2e Oversight Committee, is tasked with developing the common software design for NRAO telescopes, as far as it can be extended.

The common software design is facilitated by the definition of various “models” which describe the overall observing process and some of the components within it:

- Observatory Model;
- Archive Model;
- Project Model;
- Observing Model;
- Science Data Model.

It is the responsibility of the NRAO e2e Oversight Committee to come up with the detailed descriptions of the global models above, while each of the projects must determine in which way they deviate (most of the models are constructed as a base or core model, with the intention that each telescope defines extensions to that core model). At this point, only draft versions of the descriptions of the above models have been produced by the e2e Oversight Committee, which are available via the NRAO wiki.

The current design for the Observatory Model is shown in Figure 1. This figure represents the divergence of the model at the telescope level (the “Telescope Domain”), and those parts of the model that are common (the “Observer Domain” and “Science Domain”). In fact, some of the components of the Telescope Domain may also be common, for instance the scheduling software will have many common aspects. However, this does serve to broadly describe and delineate those areas of the software most likely to be common.

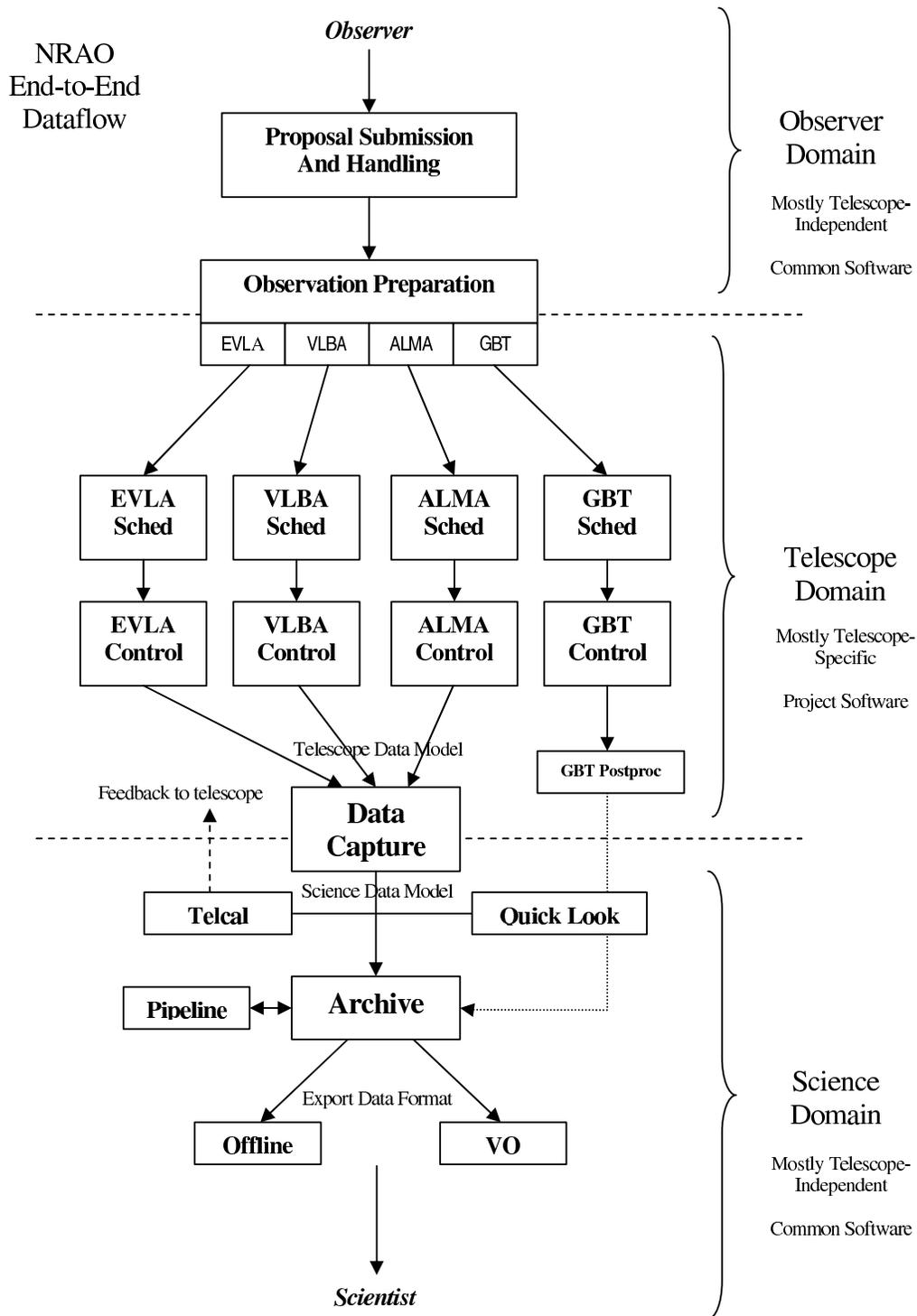


Figure 1: The Observatory Model for NRAO telescopes - the highest level description of dataflow for these telescopes.

Given that one of the primary goals of NRAO telescope software is to make it easier to use, it is clear that not only should software for a given telescope have a common look-and-feel, but this should be true, as much as possible, across all NRAO telescopes. It will be much easier for users, and especially new users, to only have to learn one type of user interface for all of the various interactions with all of the NRAO telescopes. The software should be designed and implemented with this in mind. Here, however, there is a point that illustrates a problem when trying to integrate the ALMA software with the rest of NRAO: as an observatory, NRAO would like to have all of its software have a common look-and-feel, yet the European Southern Observatory (ESO - the European partner for ALMA) would also like ALMA software (notably the user interfaces) to be as similar as possible to software for other ESO telescopes. It is hard to satisfy completely these very different constraints, since it is clear that NRAO and ESO have different user requirements in some areas (note that ESO mainly supports optical telescopes), yet it must be attempted.

5 High Level Design for EVLA Software

The common architecture shown in Figure 1 has been adopted by the EVLA overall design team, and modified slightly where needed. This design is shown in Figure 2. This design was presented at a review in June 2004, with the NRAO e2e Oversight Committee acting as the design review panel. The design was approved by that panel, though the final report is still forthcoming.

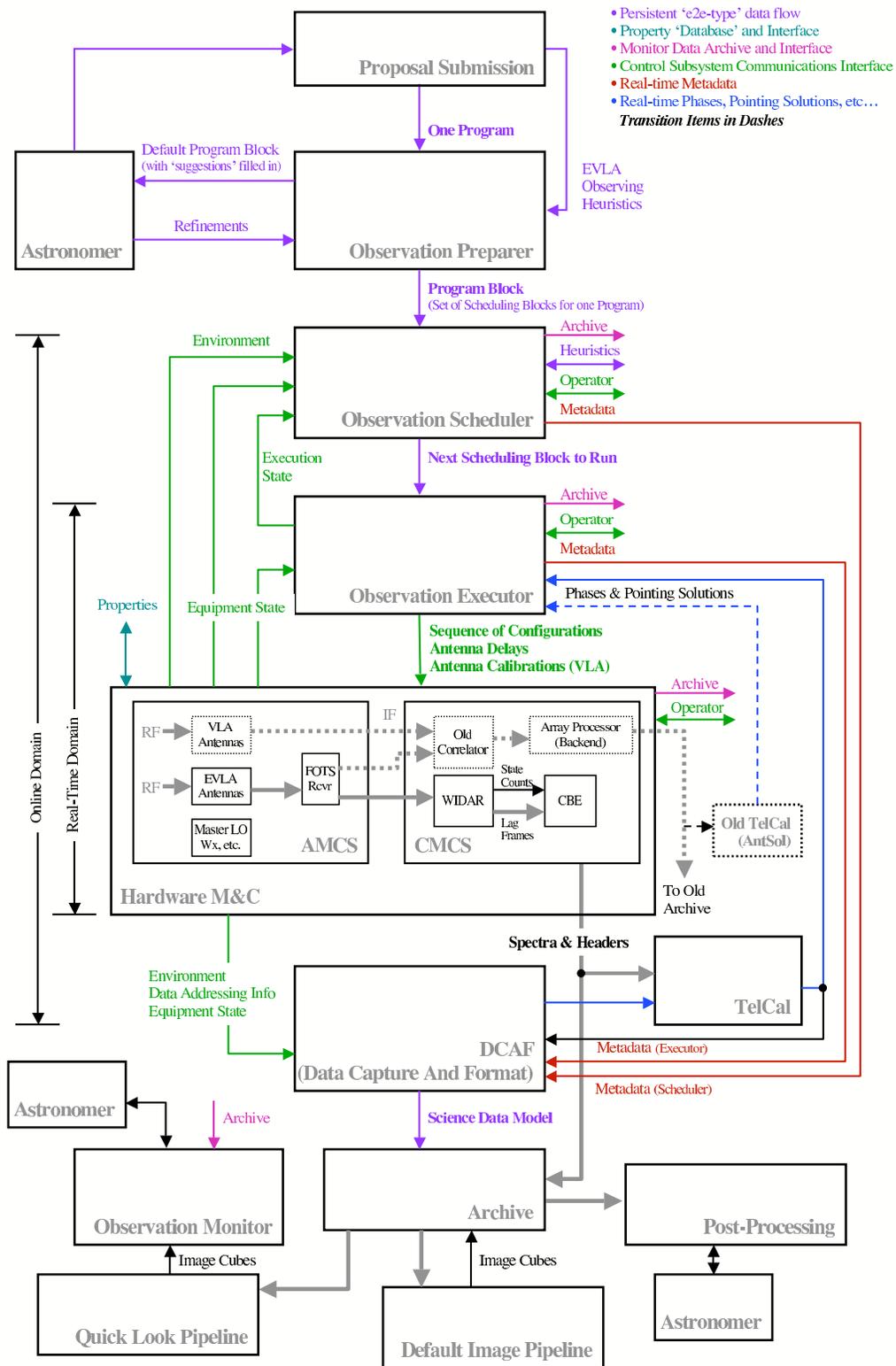
The EVLA Software System as a whole can be characterized as a hierarchically structured, static set of components. All of the sub-systems must be encountered in a fixed order during the course of proposing, preparing, scheduling, executing, archiving and reducing the data from an observation. All functional services in the System will generally be needed during observational activities. There are very few functions that will be performed on time irregular intervals of more than a few tens of minutes. As a result, the vast majority of software functionality must be online and available during observing operations. Failures must be immediately detected and remedied even for components not currently in use. Except for start-up, shutdown, and failure recovery, it is not expected that there will be a large amount of creation and destruction of processes or objects.

The System consists of a number of major components or sub-systems. These will be introduced and discussed one-by-one subsequently and in later chapters. The sub-systems fall into three distinct categories defined by operational context, especially in the area of time constraints imposed on communications.

First are the offline sub-systems: Proposal Construction and Submission, Proposal Management, Program Preparation (including Observation Preparation), and Data Reduction and Archive. Each is an independent software application or in the case of data reduction, a package of applications. External communications are limited to the User Interface and an Archive or Database for bulk data I/O. They are run independently on possibly widely geographically located processing hardware. There is a logical ordering to their use (e.g. a Proposal must be submitted before Program Preparation can be done, Data Reduction cannot be done until the observation has been executed) but there are no other restrictions to their use with respect to other parts of the system.

Next are the online Subsystems: Observation Scheduler, Data Capture and Format, and the Quick-Look Pipeline. These subsystems require external communications and hence must maintain or periodically establish direct communications with other parts of the system. This communication however does not have to be done in “real-time”. It will be permissible for some delays to be present in the delivery and receipt of external data and some data may be sent or received in advance of its need. These sub-systems will perform no real-time operations, and the output products of these subsystems are either not needed to perform real-time functions, or can be determined and delivered in advance of their need by a real-time function.

The third group is the Real-time Subsystems: Observation Executor, Monitor and Control (including both Antenna and Correlator Monitor and Control) and several independent modules such as Calc and TelCal. These components either come in direct contact with the real-time hardware systems, require real-time input to perform their function, or produce output products that are needed to perform real-time functions. At least some of the external communications of each component will be continuous or periodic with a high frequency of occurrence. A premium will be placed on rapid and secure delivery and receipt of data. Communications failures will frequently result in a loss of Science Data. The online and real-time



2004-12-08
MRSWB

Figure 2: The high level design for EVLA software.

subsystems will be started together and must remain in a correct functional state for continuous, long term observing progress to be made. Brief outages in online components can be tolerated only to the extent that storage buffers do not overflow or empty waiting for resumption of service. Outages of real-time components in the critical operational path will result in immediate cessation of operations and loss of Science Data until the fault has been corrected.

Each of the major subsystems within this design will be discussed now.

6 Proposal Preparation

The first step in any observation is the preparation and submission of a proposal for telescope time by the astronomer. This proposal is then evaluated for scientific merit and technical feasibility, and awarded time if deemed appropriate by a Time Allocation Committee (TAC). As part of this process, the observatory must have some system of handling the proposals - accepting and organizing them, sending them out to be reviewed by external reviewers (this is done at virtually all modern observatories), and gathering information for the use of the TAC.

The information which must be gathered in the proposal includes: cover sheet information (including proposers names and contact information, required telescope resources, crude source information [position, flux density, etc...], and an estimate of the time required to meet the scientific objective); a detailed scientific and technical justification for the requested time; and a separate statement of financial support. Historically, observatories have employed different methods of gathering the information required in a telescope time proposal, from typing the proposal out directly on a typewriter and sending it by mail, to filling out TeX/LaTeX forms which can be submitted electronically. Recently, the drive is toward more modern web-based GUIs and electronic submission, because of their general ease of use, and broad acceptance within the astronomical community. The NRAO proposal system will certainly be of the more modern type. One of the immediate advantages of electronic submission (if done properly) is that the information gathered can be stored in a database which can be subsequently queried for information. For instance, if an astronomer is considering submitting a proposal, the database can be queried to determine whether a similar proposal has been submitted in the past. Matches on source names, proposer names, keywords, etc..., can be searched for, making the proposal preparation process much more efficient for the astronomer. In addition, observatory staff can use the database to easily construct statistics on types of proposals, demographics of proposers, oversubscription rates, and other useful entities.

Once a proposal is accepted for time on the telescope, a “project” is created, which encapsulates the information necessary to complete the scientific objective in the proposal. That project information is carried along at all further stages of the observation, across subsystem boundaries.

7 Observation Preparation

In order to actually observe with the telescope, the more generic scientific objective carried in the proposal (and thence in the project), must be translated into something with much more detail. This is the task of the observation preparation software, which takes the higher level instructions contained in the project and turns them into commands to actually drive the telescope.

After approval by the TAC, the astronomer works within the observation preparation software to create the elements necessary to command the telescope. In an effort to make the EVLA (and all NRAO telescopes, for that matter) more accessible to the non-expert, it is a requirement that the minimal information gathered in the proposal preparation stage, augmented with a very small amount of information gathered in the observation preparation stage (such as the selection of a recommended existing default observing template), be all that is needed to complete the entire observation preparation step. Expert users will, of course, want to fiddle with as many of the observing parameters as possible, and such must be allowed.

The output of the observation preparation software is one or more *Program Blocks*, each of which describe the particular parts of the project that can be observed within a single configuration of the EVLA. Each Program Block is a collection of one or more Scheduling Blocks, which are the shortest allowable contiguous

block of observing time allowed. An example is that if a project contains 40 hours of observing a particular (non-circumpolar) source, 20 hours in each of two configurations, then there would be one Program Block for each of the configurations, and each of the Program Blocks would be broken into several Scheduling Blocks, since all 20 hours cannot be observed contiguously (it might be two blocks of 10 hours, or four blocks of 5 hours, depending on the details of the constraints on the observing).

8 Observation Scheduling

Once the Program and Scheduling Blocks have been prepared and submitted, they must be sensibly scheduled for actual observing time on the telescope. Historically, telescopes have had set schedules, well in advance (weeks to months), where particular observations were scheduled at particular times, and there was little flexibility to change these times. It is clear that this is an inefficient way to schedule any telescope, as it results in observations occurring under inappropriate conditions. The EVLA, like many other modern telescopes (and all other NRAO telescopes) has the goal of being completely dynamically scheduled. That is, the particular observation selected at any given point in time is determined by selection from a pool of available observations (Scheduling Blocks), with the selection determined by some combination of scientific priority and current observing conditions. For the most part, the astronomer does not interact directly with this scheduling, except that required conditions for observing must be entered at either the proposal or observation preparation stage, and that the proposer is notified as associated Scheduling Blocks are getting close to being observed. One exception is *manual* or *interactive* observing, during which the astronomer has much more direct contact with the telescope control software. This will only be allowed for approved observers.

9 Executor

The EVLA will be controlled by commands, which are bundled together into scripts. During normal observing, these scripts will be embedded in Scheduling Blocks, which are created during Observation Preparation. The SBs are fed to the Executor by the Scheduler, which determines what is to be observed next. In interactive mode, SBs are fed directly to the Executor (it is a matter of semantics whether it is considered that they still are fed to the Scheduler, but given infinite priority so that they are always executed immediately) by an operator, astronomer, or engineer. In manual mode, commands are typed directly into the Executor. In any of these modes, the Executor then interprets the script (or command), gathers necessary system parameters and information (some of which it may have to calculate or determine itself), and passes along commands to the M&C systems.

9.1 CALC

The EVLA software must have a mechanism to calculate (and deliver to the antennas or the WIDAR correlator after it is functioning) delays to each antenna, given the geometry of the array and sources as a function of time. The software chosen to do this is the CALC¹ package. CALC will be called by the Executor when needed to calculate the delays to the antennas or WIDAR. A wrapper will have to be written for CALC, which is written in FORTRAN, in order to fit in the software structure (note that this has already been done).

10 Monitor & Control

The Monitor & Control software takes commands from the Executor and commands the EVLA instrumentation as required to properly control the observations. This is described in much more detail in

¹CALC was developed by the Goddard Space Flight Center with assistance from other geodetic VLBI groups. It is maintained and distributed by Goddard as a service to the VLBI community.

Chapter 10. The M&C software has two major components: the Antenna M&C Subsystem (AMCS); and the Correlator M&C Subsystem (CMCS).

10.1 Antenna

The AMCS 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. The AMCS must control and monitor all hardware within the antenna itself, including the physical antenna and the various hardware subsystems (LO, IF, receivers, communications, etc...). Each hardware device is controlled by a MIB (Module Interface Board) which must be programmed with low level code. The MIBs communicate with the antenna control computer via Ethernet using the TCP/IP protocol. This antenna control computer communicates with the overall M&C computing system in a similar way.

10.2 Correlator

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.

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

11 TelCal

There are particular telescope calibration quantities which must be determined by the system - some in near-real-time. The TelCal subsystem is responsible for calculating these calibration quantities, given the type of calibration desired (antenna location, global pointing model, reference pointing, focus, bandpass, delays, polarization, flux density scale, antenna aperture efficiency, autophasing quantities, or complex gain). It takes in visibility data and metadata, performs the necessary calculations/calibrations, and passes along the data to the appropriate place - which can be the Executor or one of the Archive databases (see section 13 below) via DCAF. As a historical note, ANTSOL in the current VLA online system is a limited version of TelCal. Also note that TelCal is called RTCAT in the e2e requirements document (Butler et al. 2003a). TelCal will require a specialized Calibration Pipeline, described in section 14.1 below.

12 Observation Monitoring

After actually being scheduled on the telescope, astronomers often want to monitor the progress of the observations. The intent is to make this possible via a web-based GUI tool which displays information about the observations as they progress. Such information includes telescope setup and state, current meteorological parameters, and actual progress of the observations. The telescope setup and state information includes at least: date/time; source name; sky position; observing frequency; bandwidth; and correlator setup. The meteorological information includes: ambient temperature and pressure; dew point temperature; wind speed and direction; and auxiliary atmospheric sounding instrumentation results (atmospheric phase stability monitor, water vapor radiometers, GPS or other information on ionosphere, solar monitor, RFI monitor, etc...). The observation progress information includes: actual visibility data; instrumental spectral response (bandpass - though this might be considered part of the telescope state); current calibration information; project status; program status; and any results from the Quick-look Pipeline (see section 14.2). Where it makes sense, time histories of all the quantities need to be displayed.

13 Data Archiving

The data, once collected by the telescope control software, must be stored in a permanent archive, to allow subsequent retrieval by astronomers (both proposers and others - after the proprietary period has expired). The “archive” can be thought of as a collection of a number of databases: Project database; Calibration database; Monitor database; and Science data archive. In fact, there are a number of other related databases (including the Proposal database; Observations database; and Publications database), but they are not as tightly coupled to the actual raw data coming from the telescope, and in many cases transcend a particular NRAO telescope, so are not included in the above list. In addition, there are a number of other databases which hold information mostly of interest only to the control system or to operations staff (including the Telescope Configuration database; Site Properties database; and Maintenance History database), but these are not of interest in the context being discussed here. When referring to *the* “archive,” the entire suite of databases above is meant, and access to all of them in all combinations must be well supported by the EVLA software.

The information contained in the archive includes: raw visibility data; related header information (meta-data); calibration data; environmental conditions (meteorological and related) data; instrumental monitor data; and image cubes which result from the pipeline data processing (see section 14.3). The archive must be searchable, with complex queries, and it must be possible to retrieve data from it (when the proprietary period has expired). In addition, Virtual Observatory (VO) queries must be supported. The best method for distributing the large amounts of data produced by the EVLA (see next section) has yet to be determined.

13.1 Output Capture and Storage

As the Scheduling Block passes from Scheduler to Executor and as commands are generated from the executing Observe Script, time tagged metadata will flow to DCAF (“Data Capture And Format”). The arriving metadata will be associated and organized according to data type and time tag by DCAF into a time ordered stream. This stream is segmented on-the-fly by DCAF into Integrations, Sub-scans, Scans and Execution Blocks.

Astronomical data will flow directly from the CBE to the Science Archive without going through DCAF. Additional streams needed from time to time by functions like TelCal will be teed and directed from the appropriate source.

Each Execution Block corresponds to one complete execution of the entire Observe Script of a Scheduling Block. As each Execution Block is completed, the corresponding Scheduling Block is marked in the Project database as having been executed one additional time. The Scheduling Subsystem will compare the updated execution count for the Scheduling Block against the iteration limit for that block. If the limit has been reached, the Scheduling Block will be removed from further consideration by the Scheduler and marked as finished in the Project database. Once all Scheduling Blocks in a Program Block have been finished, the Program is marked as finished. Once all Programs in a Project have been finished, the Project is marked as finished.

When a sufficiently complete block of metadata is available in DCAF, it will be ready for export to the Science Archive. Sufficiently complete means that one or more data reduction processes may be meaningfully applied to it. The point at which this will occur may vary with the nature of the observation and the purpose of various data reduction tasks and may happen at the Execution Block, Scan or Sub-scan level. The data will be output according to a prescribed format. Data needed by the QLP will be exported directly to it by DCAF in a format appropriate to the needs of the QLP services being provided.

13.2 Science Data Model

This model defines the content of science data products stored in the Science Archive or exported to the user community. All post-processing is defined in terms of the SDM. At the end of the Observing process data is produced which is conformant to the SDM. Data is stored in the Science Archive in this form, and is exported to the user in an Export Data Format, which is a mapping of the SDM to some external representation such as FITS or XML. An Export Data Format will be specified that will be used to convert

input received by DCAF into a form suitable for storage in the Science Archive and use by Data Reduction systems. Active efforts are underway to specify the format in conjunction with the needs of ALMA and GBT. The current thinking is that there will be a core NRAO SDM, with extensions for each of the particular telescopes. The core SDM will be adapted from the current ALMA Science Data Model (Viallefond & Lucas 2004).

13.3 Data rates

The volume of data produced by the EVLA is dominated by the actual raw visibility data, and is quite prodigious. The WIDAR correlator is capable of producing as much as 350 GB/sec. This is clearly beyond the storage capabilities that will be accessible to the observatory in 2012 (and far beyond). Peak and average data rates have therefore been derived based on more reasonable estimates of what it will be possible to store (and process - see section 15.1) at that time, combined with what is thought is a reasonable range of scientific programs and their expected data production (Perley 2004). As storage and processing capabilities increase, so does the allowable average and peak data rate into the archive. The staged peak data rates are shown in Table 1, including into the EVLA Phase II era.

Table 1: EVLA Peak Data Rates

Date	Data Rate MB/s
2009	25
2012	250
2017	1600

14 Data Reduction Pipelines

A large fraction of data collected by the EVLA will be taken in one of a handful of standard observing modes, for example, low frequency continuum, high frequency continuum, HI (neutral hydrogen) spectral line, or polarization. Given the considerable experience in reducing data taken in similar kinds of modes with the VLA, it is reasonable to assume that reduction of this type of data can be completely automated. The post-processing package (see section 15), when combined with some information collected in the observation preparation stage, during actual observing, and with some *heuristics* (rules for what to do given certain situations) should be sufficient to complete such automatic reductions. This does imply, however, that certain critical parts of the post-processing package which are not currently in existence are implemented and robust, for example automatic flagging of bad data, and imaging of wide-bandwidth ratio and wide-field data. There are three types of pipelines which will be utilized in the software for all NRAO telescopes, including the EVLA: the *calibration pipeline*; the *quick-look pipeline*; and the *default image archive pipeline*.

14.1 Calibration pipeline

There are a number of types of observations of calibrators which will be used by the monitor and control system to set instrumental parameters necessary for observing. Among these are antenna location determination, focus determination, and reference pointing determination. In fact, for every calibrator observed, a quick reduction of the data will provide the current atmospheric conditions as well as other information. It is thus intended that all calibrators will pass through the calibration pipeline, and the results be made available to the monitor and control system. This pipeline will be a part of the TelCal subsystem (see section 11).

14.2 Quick-look pipeline

During observations of a source, it is instructive to obtain an initial data reduction to see whether the instrument is behaving as expected, and in order to guide further observations (in the case of manual or interactive observing). Such an initial data reduction might use the default stored calibration parameters for things like bandpass and polarization calibration, and would not do detailed deconvolutions of the raw images. This is the job of the quick-look pipeline. Even though it is quite simple, there are a few parameters which should be able to be tuned for the quick-look pipeline, for example, when looking at maser sources, it may be only interesting to look at a small subset of the total number of available channels in order to see if the maser emission is present. The astronomer should also be able to set the frequency of how often the quick-look pipeline is run, or be able to initiate it manually (assuming that the reduction can actually be done in a relatively short amount of time).

14.3 Default image archive pipeline

One of the primary data products of the EVLA is a standard image archive. This will be a valuable resource for future astronomers (“data-miners”), and will provide a consistent record of the images produced from the data taken by the EVLA. This archive should only be initiated once for each project, although if the processing is not excessive it can be repeated if more accurate calibrations become available. Since it must be consistent across projects, there should be very few (if any) selectable parameters for this pipeline. Of course, for complicated projects which do not fit into one of the standard observing modes or models, there will be no guarantee that this pipeline will produce meaningful results. One of the difficulties is then estimating exactly what the quality of the images in the default image archive is. Various measures of image quality can be constructed, but in practice it is extremely difficult to apply them to images taken across a very wide range of the possible observing mode parameter space.

15 Data Post-Processing

For all data which cannot be reliably reduced via a pipeline, or for astronomers who wish to modify or extend what is done within the pipeline, there must be a post-processing software package capable of performing all steps necessary to turn the measured visibilities into final image cubes. For the VLA, several packages have been used for data editing and calibration over the years, but for over a decade, AIPS (Greisen 2003) has been the primary package for this. For nearly the entire lifetime of the VLA, AIPS has been the primary software package for imaging. For some time now, however, AIPS++ (Cornwell & Wieringa 1997) has been developed as the modern successor to AIPS. Currently, AIPS++ is considered the primary candidate package for the calibration and imaging of data from all NRAO telescopes, and this is true for the EVLA. However, we note that any package that satisfies all of the requirements for EVLA post-processing (Myers et al. 2003) can be used for these purposes. Currently, the AIPS++ software is being developed with the Scientific Software Group (SSG) at NRAO. There is another committee looking at the longer-term issues involved with data reduction for the observatory, which is providing input to the SSG.

15.1 Computing requirements

Whatever the software package, it is clear that the amount of data produced by the EVLA, and the algorithms involved in turning that data into final image cubes, will demand considerable computing power (see Table 1 for an estimate). There have been several recent studies of this (Cornwell 2001a; Perley & Clark 2003; Cornwell 2004), concluding that the required computing for imaging of the bulk of EVLA data is feasible given expected increases in computing power. For the most demanding projects (for example, full-beam imaging with the Phase II EVLA), however, advances in raw compute power are not enough, and further algorithmic advances are needed. These advances are also needed to reach the impressive theoretical sensitivity levels predicted for the EVLA (given simple arguments using system temperature and bandwidth).

15.2 Algorithm development

The developments needed for EVLA (and future interferometers in general) can be generally grouped into three categories: doing things faster (parallelization [Golap et al. 2001; Young & Roberts 2000; Roberts 1996], grid computing (Berriman et al. 2003); doing things that we are already doing, but better (for example, multi-scale CLEAN (Bhatnagar & Cornwell 2003); and doing completely new things. Specifically, in the second and third of these categories for the EVLA there are a number of important developments needed which are common with ALMA:

- automatic flagging;
- tropospheric corrections, including “phase screen” derivation (Carilli & Holdaway 1999; Cai & Cornwell 2004);
- exotic imaging algorithms (pixons [Cornwell 2001b; Puetter & Yahil 1999], wavelets (Maisinger et al. 2004), other Bayesian methods, etc...);
- combination of single-dish and interferometric data (Stanimirovic 2002);
- wide-field full-polarization imaging (Cornwell 2003) especially with different primary beams per antenna and time variable primary beams (Cornwell 2002) (some of this is common with ALMA, some unique to the EVLA).

And a number which are unique (at least mostly) to the EVLA:

- wide-field, wide-bandwidth imaging (Cornwell et al. 2003; Cornwell 2002);
- RFI excision and subtraction (Perley & Cornwell 2003);
- ionospheric corrections, including “phase screen” derivation (Smirnov 2000).

The intent is for scientists and programmers to work on these algorithmic developments in concert, since input from both is necessary to solve these very difficult problems.

16 Testing and Acceptance

All of the software described above must go through rigorous testing, both internally (within the software teams implementing the designs), and externally. In places where the astronomer must interact directly with the system, the software must be tested by actual astronomers. In the early stages, this will be limited to scientific staff members at NRAO (those same staff members who sit on the requirements committee). In later stages, it is foreseen that this will be extended to astronomers at other institutions. A detailed testing plan is currently being developed, given constraints on current levels which can be supported (which is highly dependent on the total amount of scientific staff support available). In these same areas, the Project Scientist for Software is responsible for the ultimate acceptance of the software, in consultation with the overall Project Scientist. The committee of staff scientists will certainly contribute critically here as well.

17 Timeline

The due date for the various subsystems is determined by the timeline of hardware availability, along with an assessment from operations on whether certain types of operations can be supported, and with an estimate from the EVLA Computing Division management on whether it is feasible to meet that date. The following table lists the major subsystems and the target dates for completion of the various intermediate releases.

Table 2: EVLA subsystem timeline.

subsystem	first release	beta release	final release	driver+date
Proposal Preparation	Aug-06	Aug-07	Mar-08	2
Observation Preparation	Dec-06	Mar-08	Mar-09	3
Scheduling	Dec-07	Mar-09	Dec-11	4
Monitor Data Archive	Mar-04	Jun-05	Dec-05	1
Science Data Archive	Dec-05	Mar-08	Mar-09	3
Pipeline	Mar-08	Mar-09	Dec-11	4
Observation Status	Mar-06	Jun-07	Mar-08	2

1 - multiple antennas available (Mar-04); 2 - shared risk observing (Mar-08); 3 - full science observing (Mar-09); 4 - end of construction (Dec-11).

18 Computing Hardware

The computing hardware for EVLA will include computers at the antennas, at the control building, and at the AOC. There are plans for all of these - for the antennas and control building see Morgan et al. (2004) and Morgan (2004). The computers at the AOC will include those necessary for operating the archive, and any other applications (e.g., web servers, database servers, a proposal handling computer), as well as, if necessary, powerful computers (or clusters) for post-processing. We are currently investigating the necessity of such a post-processing cluster.

The network connecting the antennas to the control building is described in Chapter 10. The network between the control building and the AOC is currently a single T1 internet connection, and clearly needs upgrading for EVLA use. Negotiations with Western New Mexico Telephone have been ongoing for this purpose.

References

- Berriman, G.B., D. Curkendall, J.C. Good, J.C. Jacob, D.S. Katz, M. Kong, S. Monkewitz, R. Moore, T.A. Prince & R.E. Williams, 2003, Architecture for access to a compute intensive image mosaic service in the NVO, in *Virtual Observatories*, ed. A.S. Szalay, Proc. SPIE, 4846, 91
- Bhatnagar, S., & T.J. Cornwell, 2003, Scale-sensitive (multiscale) deconvolution of astronomical interferometric images, in *Astronomical Adaptive Optics Systems and Applications*, eds. R.K. Tyson & M. Lloyd-Hart, Proc. SPIE, 5169, 331
- Butler, B., C. Brogan, C. Chandler, B. Clark, S. Myers, P. Napier, F. Owen, R. Perley, & M. Rupen, 2003a, EVLA e2e Science Software Requirements, EVLA Computing Memo 26
- Butler, B., B. Clark, S. Durand, B. Hayward, J. Jackson, & B. Sahr, 2003b, EVLA Engineering Software Requirements, EVLA Computing Memo 29
- Butler, B.J., G. van Moorsel, & D. Tody, 2004a, Software for the EVLA, Proc. SPIE, 5493, 1
- Butler, B., J. Benson, B. Clark, F. Owen, R. Perley, & K. Sowinski, 2004b, EVLA Real-Time Science Software Requirements, EVLA Computing Memo 38
- Cai, C., & T. Cornwell, 2004, Parameterized Phase Screen Modelling for the VLA, in preparation
- Carilli, C.L., & M.A. Holdaway, 1999, Tropospheric phase calibration in millimeter interferometry, *Radio Sci.*, 34, 817
- Cornwell, T., & M. Wieringa, 1997, The Design and Implementation of Synthesis Calibration and Imaging in AIPS++, in *ADASS VI*, eds. G. Hunt & H.E. Payne, ASP Conference Series 125, 10

- Cornwell, T., 2001a, Computing for EVLA Calibration and Imaging, EVLA Memo 24
- Cornwell, T., 2001b, Pixon imaging in AIPS++, AIPS++ Note 242
- Cornwell, T., 2002, High dynamic range radio synthesis imaging with time variable primary beams, Presented at 2002 URSI meeting, Boulder
- Cornwell, T.J., 2003, Full Primary Beam Stokes I, Q, U, V Imaging, EVLA Memo 62
- Cornwell, T.J., K. Golap, & S. Bhatnagar, 2003, W Projection: A New Algorithm for Non-Coplanar Baselines, EVLA Memo 67
- Cornwell, T., 2004, EVLA and SKA computing costs for wide field imaging, EVLA Memo 76
- Golap, K., A. Kembell, T. Cornwell, & W. Young, 2001, Parallelization of Widefield Imaging in AIPS++, in *ADASS X*, eds. F.R. Harnden, Jr., F.A. Primini, & H.E. Payne, ASP Conference Series 238, 408
- Greisen, E.W., 2003, AIPS, the VLA, and the VLBA, in *Information Handling in Astronomy - Historical Vistas*, ed. A. Heck, Kluwer, 109
- Maisinger, K., M.P. Hobson, & A.N. Lasenby, 2004, Maximum-entropy image reconstruction using wavelets, *MNRAS*, 347, 339
- Morgan, T., 2004, Draft VLA/EVLA Transition Observing System Development and Re-engineering Plan, EVLA Computing Memo 37
- Morgan, T., K. Ryan, K. Sowinski, & B. Waters, 2004, EVLA High Level Software Design, EVLA Computing Memo 33
- Myers, S., B. Butler, C. Chandler, B. Clark, F. Owen, M. Rupen, C. Brogan, R. Perley, & P. Napier, 2003, DRAFT - EVLA Data Post-Processing Software Requirements, EVLA Computing Memo 28
- Perley, P., J. Campbell, C. Chandler, R. Moeser, K. Sowinski, & P. Van Buskirk, 2003b, EVLA Array Operations Software Requirements, EVLA Computing Memo 27
- Perley, R., & B. Clark, 2003, Scaling Relations for Interferometric Post-Processing, EVLA Memo 63
- Perley, R., & T. Cornwell, 2003, Removing RFI Through Astronomical Image Processing, EVLA Memo 61
- Perley, R., 2004, Wide-Field Imaging with the EVLA: WIDAR Correlator Modes and Output Data Rates, EVLA Memo 64
- Puetter, R.C., & A. Yahil, 1997, The Pixon Method of Image Reconstruction, in *ADASS VIII*, eds. D.M. Mehringer, R.L. Plante, & D. A. Roberts, ASP Conference Series, 172, 307
- Roberts, D., 1996, Parallelization of AIPS++, AIPS++ Note 203
- Smirnov, O., 2000, Ionospheric Corrections in AIPS++, AIPS++ Note 235
- Stanimirovic, S., 2002, Short-Spacings Correction from the Single-Dish Perspective, in *Single-Dish Radio Astronomy: Techniques and Applications*, eds. S. Stanimirovic, D. Altschuler, P. Goldsmith, & C. Salter, ASP Conference Series, 278, 375
- Viallefond, F., & R. Lucas, 2004, ALMA Export Data Format v2.0, ALMA Document ALMA-70.00.00.00-004-A-SPE
- Young, W., & D. Roberts, 2000, Parallelization Test Results, AIPS++ Note 232