Software for the EVLA

Bryan J. Butler, Gustaaf van Moorsel, Doug Tody

National Radio Astronomy Observatory, Socorro, NM, USA

ABSTRACT

The Expanded Very Large Array (EVLA) project is the next generation instrument for high resolution long-millimeter to short-meter wavelength radio astronomy. It is currently funded by NSF, with completion scheduled for 2012. The EVLA will upgrade the VLA with new feeds, receivers, data transmission hardware, correlator, and a new software system to enable the instrument to achieve its full potential. This software includes both that required for controlling and monitoring the instrument and that involved with the scientific dataflow. We concentrate here on a portion of the dataflow software, including: proposal preparation, submission, and handling; observation preparation, scheduling, and remote monitoring; data archiving; and data post-processing, including both automated (pipeline) and manual processing. The primary goals of the software are: to maximize the scientific return of the EVLA; provide ease of use, for both novices and experts; exploit commonality amongst all NRAO telescopes where possible. This last point is both a bane and a blessing: we are not at liberty to do whatever we want in the software, but on the other hand we may borrow from other projects (notably ALMA and GBT) where appropriate. The software design methodology includes detailed initial use-cases and requirements from the scientists, intimate interaction between the scientists and the programmers during design and implementation, and a thorough testing and acceptance plan.

Keywords: NRAO, EVLA, software

1. INTRODUCTION

The Very Large Array (VLA) was commissioned in 1980, and provided more than an order of magnitude improvement in imaging over all preceding radio telescopes. There is little doubt that it is, scientifically, the most productive radio telescope ever constructed. Unfortunately, little improvement in the VLA's fundamental capabilities has taken place since commissioning, other than in post-correlation image processing. In particular, the fractional bandwidth, frequency access, and correlator are all areas in which the VLA has not taken advantage of vast improvements in technical capabilities which have occurred since 1980.

Beginning in the early 1980s, plans were made to apply newer technologies to the VLA in order to improve its technical, and hence scientific, capabilities. These plans were presented to the 1990 Astronomy and Astrophysics Survey Committee, which gave them a high recommendation. Despite this favorable view in the decadal report, funding was not obtained at that time. By the late 1990s, the gap between current technologies and those used in the VLA was so large that a complete redesign and replacement of the electronics was clearly the appropriate path for upgrading the instrument. At that time, it was decided that the project to effect this replacement should be renamed from the "VLA Upgrade" to the "Expanded VLA" (or EVLA), to reflect the enormous improvements in telescope capabilities that a complete redesign, using modern technologies, could provide.

The EVLA project makes sense because the most costly parts of any multi-element telescope array - the antennas and the infrastructure to support them - are in place. The EVLA is hence a "leveraged" project. For a modest incremental investment (about 1/3 of the replacement cost of the existing facility), the astronomical capabilities are multiplied more than ten-fold.

The specific top-level goal of the EVLA project is to enhance the performance of the VLA by an order of magnitude or more in the areas of sensitivity, frequency coverage, spectral resolution, and spatial resolution. The project is being implemented in two overlapping phases: Phase I, for which funding began in 2000 and completion should be achieved by 2012; and Phase II, which is not currently funded, but is planned to begin in 2006 and finish by 2013. Progress in Phase I is very good, with tests of the new hardware and software now underway. In fact, first light and first fringes have already been achieved. A critical component of Phase I is a new correlator (WIDAR), being designed and built by the Canadian Herzberg Institute of Astrophysics at the

DRAO in Penticton, BC, Canada. This new correlator will be delivered beginning in 2007. The first shared risk science with the early correlator will be done in late 2007. A proposal for funding Phase II has been delivered to the National Science Foundation.

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 VLA control system software is written in FORTRAN and assembly code, which while able to do the job, is hard to maintain and update. A similar situation exists for the peripheral software. The VLA control system software also depends on a single hardware and software vendor, while the EVLA software must be much more flexible. In addition, the new electronics will require slightly different control software, which should fit in with the new overall control software. The control (and monitor) software should also fit into an overall system involving all stages of the EVLA observing process, i.e., the entire *dataflow* of the EVLA should fit together into a seamless software system which is easy to use, maintain, and extend or upgrade. This software thus encompasses all stages of EVLA observing, from proposal preparation to final data reduction. The software required by engineers and operations staff to maintain the instrumentation and operating procedures of the EVLA must also fit into the overall software system. Finally, it is imperative that the software across all NRAO telescopes (including the EVLA, the Very Long Baseline Array [VLBA], Green Bank Telescope [GBT], and the Atacama Large Millimeter Array [ALMA]) be made as similar as possible - both for ease of use by astronomers, engineers, and operations staff, and for the software developers themselves, during development, maintenance, and modification. All software developers, as well as the users of the instrument (in the broad sense, not just the astronomers) have the potential to benefit from using similar software, as long as it satisfies all of the requirements. That is not to say that software should be blindly copied from instrument to instrument. The perils of copying software from one project to another are well documented in the literature. It is also clear that the software must diverge at the level where the instrumentation differs. It would make little sense for the Green Bank Telescope (GBT) to copy parts of the EVLA (or ALMA) software concerned with correlating the signals between the various telescopes. However, a common top-level design is clearly advantageous, and the lower-level designs should be as similar as possible, to as close to the instrument as makes sense.

In this document, we will focus on those parts of the software system which impact the astronomer directly. These are only part of the total dataflow, of course, but are critical in terms of the scientific output of the instrument. Specifically, we will focus on: proposal preparation, submission, and handling; observation preparation, scheduling, and remote monitoring; data archiving; and data post-processing, including both automated (pipeline) and manual processing. We will not discuss the detailed monitor and control software for the instrument. We will also not discuss the software needed by engineering and operations staff to maintain the instrument.

2. GENERAL DESIGN PRINCIPLES AND METHODOLOGY

The primary goals of the EVLA software are to: 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 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. For the specific case of the EVLA, a committee of staff scientists in Socorro came up with the main requirements document.[1] That document is intended to be updated as needed, by that committee. It outlines the detailed requirements in all areas being covered here except the post-processing software, which is described in a companion document, also written by that committee.[2] These documents do not cover the engineering and operations software requirements, which are covered in other places.[3, 4] Use cases are being written by members of that committee, 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 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 several system models, largely common to all NRAO telescopes, which describe the overall observing process and some of the components within it. The highest level of these models is the Observatory Model, 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.

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.

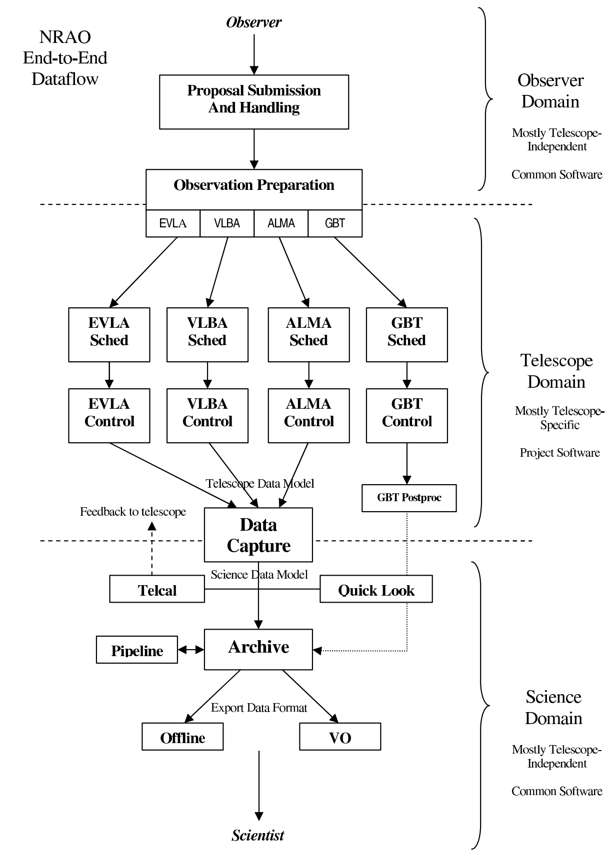

Figure 1. The Observatory Model for NRAO telescopes - the highest level description of dataflow for these telescopes.

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

6. INDIVIDUAL SUBSYSTEMS

6.1. 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

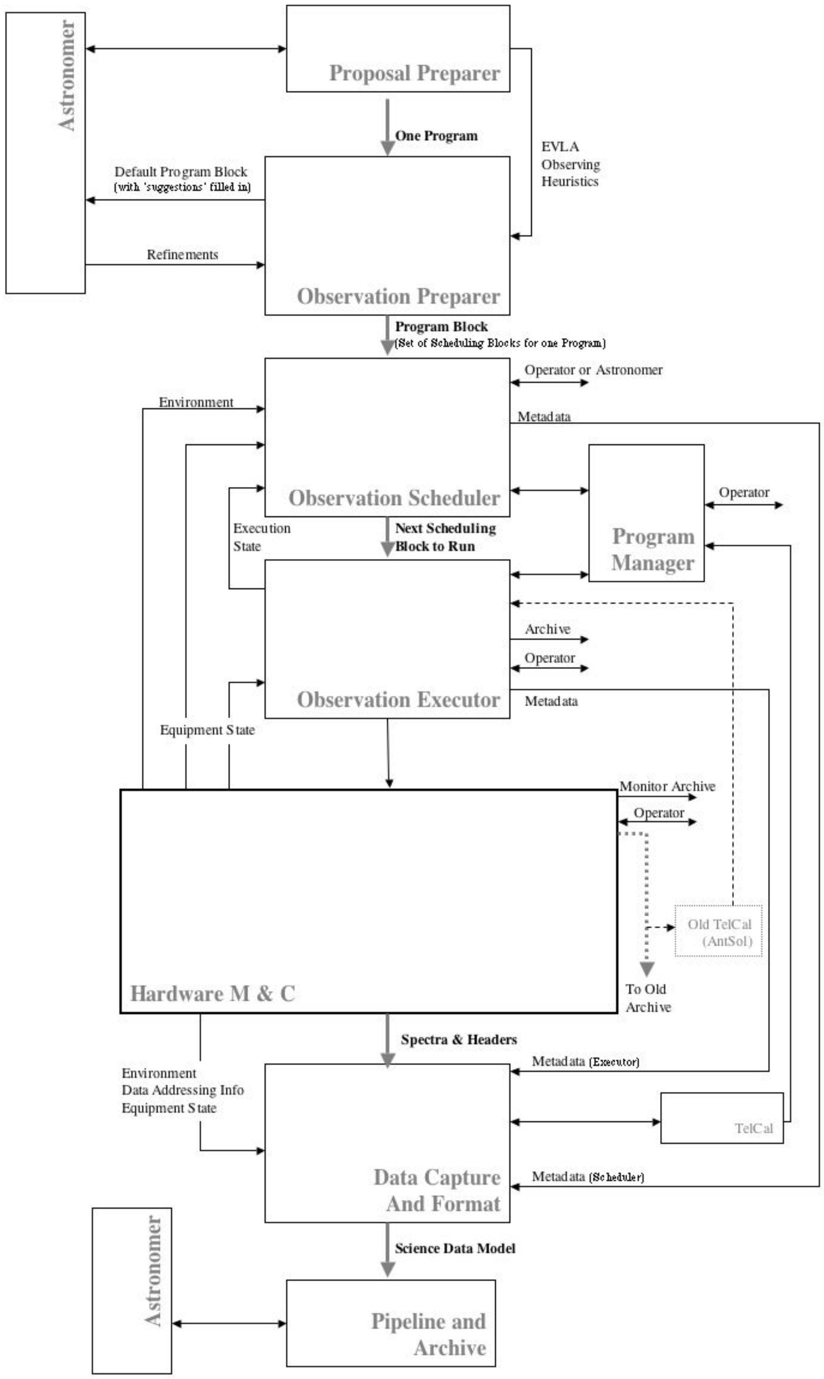

Figure 2. The high level design for EVLA software.

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.

6.2. 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).

6.3. 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.

6.4. 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 6.6.2). Where it makes sense, time histories of all the quantities need to be displayed.

6.5. 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 (metadata); calibration data; environmental conditions (meteorological and related) data; instrumental monitor data; and image cubes which result from the pipeline data processing (see section 6.6.3). 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.

6.5.1. 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 6.7.1) at that time, combined with what is thought is a reasonable range of scientific programs and their expected data production.[5] 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 and Needed Computing Power

Date	Data Rate MB/s	Average Compute Rate TFlop	2000 Equivalent Gflop
2009	25	0.5	8
2012	250	5.0	19
2017	1600	32	8

6.6. 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 6.7), 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*.

6.6.1. 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.

6.6.2. 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).

6.6.3. 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.

6.7. 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[6] 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++[7] has been developed as the modern successor to AIPS. Following an extensive review and design effort a new, scalable data processing system is being engineered in which the science data processing will be based primarily upon the AIPS and AIPS++ processing modules, but the underlying framework will be much more general. In this way, the power of a modern framework will be combined with the proven data processing of the existing packages, resulting in a processing system which should carry NRAO into the future operation of the GBT, ALMA, and EVLA.

6.7.1. 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. There have been several recent studies of this,[8–10] concluding that the required computing for imaging of the bulk of EVLA data is feasible given expected increases in computing power. Table 1 shows the estimated computing power needed for the EVLA, illustrating how it increases as the data rate for the EVLA is allowed to increase over time. The last column in that table shows that computing power normalized to the year 2000, assuming Moore's law holds through the associated date (possible through 2012, unlikely through 2017). 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).

6.7.2. Algorithm development

The developments needed for EVLA (and future interferometers in general) can be generally grouped into three categories: doing things faster (parallelization,[11–13] grid computing[14]); doing things that we are already doing, but better (for example, multi-scale CLEAN[15]); 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[16,17];
- exotic imaging algorithms (pixons,[18,19] wavelets,[20] other Bayesian methods, etc...);
- combination of single-dish and interferometric data[21];
- wide-field full-polarization imaging,[22] especially with different primary beams per antenna and time variable primary beams[23] (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[23,24];
- RFI excision and subtraction[25];
- ionospheric corrections, including "phase screen" derivation.[26]

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.

7. 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.

8. CURRENT STATUS

The first overall design for EVLA software is near completion, with review to occur this summer. After this review, the more detailed design at the subsystem level will begin. For some elements of the system, this detailed design has already begun - notably for the proposal preparation software (driven by needs at the GBT, but taking input from all of the telescopes), and the archive search and retrieval software (which has been implemented for the current VLA archive).

9. SUMMARY

The EVLA software is designed to allow the EVLA to reach its full scientific potential. In particular, those areas of the software with which the astronomer interacts directly are being designed to be easy to use, yet powerful, flexible, and extensible, while at the same time meeting the fundamental requirements of the astronomers. By combining the experience of the VLA with modern design and implementation techniques, the software will meet these lofty goals. In an NRAO-wide design effort, the EVLA software will have interfaces common with the other main NRAO telescopes: ALMA, VLBA, and GBT. In this way, it will be much easier for the astronomer to use the wide array of powerful instruments offered by NRAO to achieve the scientific goals which are at the heart of the instruments and NRAO itself.

ACKNOWLEDGMENTS

Documents produced by the e2e oversight committee of NRAO (and especially by Allen Farris), and the overall design team of the EVLA were greatly appreciated.

REFERENCES

1. B. Butler, C. Brogan, C. Chandler, B. Clark, S. Myers, P. Napier, F. Owen, R. Perley, and M. Rupen, "EVLA e2e Science Software Requirements," *EVLA Computing Memo Series* **26**, 2003.
2. S. Myers, B. Butler, C. Chandler, B. Clark, F. Owen, M. Rupen, C. Brogan, R. Perley, and P. Napier, "DRAFT - EVLA Data Post-Processing Software Requirements," *EVLA Computing Memo Series* **28**, 2003.
3. S. Durand, B. Butler, B. Clark, R. Hayward, J. Jackson, and B. Sahr, "EVLA Engineering Software Requirements," *EVLA Computing Memo Series* **29**, 2003.
4. R. Moeser, P. Perley, J. Campbell, C. Chandler, K. Sowinski, and P. van Buskirk, "EVLA Array Operations Software Requirements," *EVLA Computing Memo Series* **27**, 2003.
5. R. Perley, "Wide-Field Imaging with the EVLA: WIDAR Correlator Modes and Output Data Rates," *EVLA Memo Series* **64**, 2004.
6. E. W. Greisen, "AIPS, the VLA, and the VLBA," in *Information Handling in Astronomy - Historical Vistas*,
7. T. Cornwell and M. Wieringa, "The Design and Implementation of Synthesis Calibration and Imaging in AIPS++," in *ADASS VI*, G. Hunt and H. E. Payne, eds., *ASP Conference Series* **125**, p. 10, 1997.
8. T. Cornwell, "Computing for EVLA Calibration and Imaging," *EVLA Memo Series* **24**, 2001.
9. R. Perley and B. Clark, "Scaling Relations for Interferometric Post-Processing," *EVLA Memo Series* **63**, 2003.
10. T. Cornwell, "EVLA and SKA computing costs for wide field imaging," *EVLA Memo Series* **76**, 2004.
11. K. Golap, A. Kemball, T. Cornwell, and W. Young, "Parallelization of Widefield Imaging in AIPS++," in *ADASS X*, J. F. R. Harnden, F. A. Primini, and H. E. Payne, eds., *ASP Conference Series* **238**, p. 408, 2001.
12. W. Young and D. Roberts, "Parallelization Test Results," *AIPS++ Note* **232**, 2000.
13. D. Roberts, "Parallelization of AIPS++," *AIPS++ Note* **203**, 1996.
14. G. B. Berriman, D. Curkendall, J. C. Good, J. C. Jacob, D. S. Katz, M. Kong, S. Monkewitz, R. Moore, T. A. Prince, and R. E. Williams, "Architecture for access to a compute intensive image mosaic service in the NVO," in *Virtual Observatories*, A. S. Szalay, ed., *Proc. SPIE* **4846**, p. 91, 2003.

15. S. Bhatnagar and T. J. Cornwell, "Scale-sensitive (multiscale) deconvolution of astronomical interferometric images," in *Astronomical Adaptive Optics Systems and Applications*, R. K. Tyson and M. Lloyd-Hart, eds., *Proc. SPIE* **5169**, p. 331, 2003.
16. C. L. Carilli and M. A. Holdaway, "Tropospheric phase calibration in millimeter interferometry,"
17. C. Cai and T. Cornwell, "Parameterized Phase Screen Modelling for the VLA," *in preparation* , 2004.
18. T. Cornwell, "Pixon imaging in AIPS++," *AIPS++ Note* **242**, 2001.
19. R. C. Puetter and A. Yahil, "The Pixon Method of Image Reconstruction," in *ADASS VIII*, D. M. Mehringer, R. L. Plante, and D. A. Roberts, eds., *ASP Conference Series* **172**, p. 307, 1997.
20. K. Maisinger, M. P. Hobson, and A. N. Lasenby, "Maximum-entropy image reconstruction using wavelets," *MNRAS* **347**, p. 339, 2004.
21. S. Stanimirovic, "Short-Spacings Correction from the Single-Dish Perspective," in *Single-Dish Radio Astronomy: Techniques and Applications*, S. Stanimirovic, D. Altschuler, P. Goldsmith, and C. Salter, eds., *ASP Conference Series* **278**, p. 375, 2002.
22. T. J. Cornwell, "Full Primary Beam Stokes I, Q, U, V Imaging," *EVLA Memo Series* **62**, 2003.
23. T. Cornwell, "High dynamic range radio synthesis imaging with time variable primary beams," *Presented at 2002 URSI meeting, Boulder* , 2002.
24. T. J. Cornwell, K. Golap, and S. Bhatnagar, "W Projection: A New Algorithm for Non-Coplanar Baselines," *EVLA Memo Series* **67**, 2003.
25. R. Perley and T. Cornwell, "Removing RFI Through Astronomical Image Processing," *EVLA Memo Series* **61**, 2003.
26. O. Smirnov, "Ionospheric Corrections in AIPS++," *AIPS++ Note* **235**, 2000.