National Radio Astronomy Observatory

COMPUTER ADVISORY GROUP DOCUMENTATION

REVIEW OF RECENT DEVELOPMENTS IN COMPUTING

August 1984

Table of Contents

1. INTRODUCTION....................................page 3

2. AIPS PROJECT REVIEW..................................3
2.1 Introduction...3
2.2 Project History......................................3
2.3 Technical Overview...................................5
2.4 User Written AIPS Software...........................7
2.5 Portability..7
2.6 Development Plans....................................8

3. WHATEVER HAPPENED TO THE PIPELINE?..................11
3.1 Introduction..11
3.2 Genesis...11
3.3 Prototype...12
3.4 Demise of the Prototype.............................13
3.5 Full Implementation.................................14
3.6 Shooting at a Moving Target.........................16
3.7 Complexity..18
3.8 Current Status......................................22
3.9 Lessons Learned.....................................23

4. SUMMARY OF THE VLA ON-LINE COMPUTER DEVELOPMENT
FROM MID-1982 TO MID-1984...............................25
4.1 Overview..25
4.2 Changes to the Existing System......................25
4.3 Upgrade of the On-Line System.......................26

5. SPECIAL PROJECTS....................................27
5.1 Introduction..27
5.2 Mass Storage Unit...................................27
5.3 Image Storage Unit..................................28
5.4 Digital Switch......................................28

6. GENERAL COMPUTING...................................29

1. - INTRODUCTION

Since the last meeting of the Computer Advisory Group in March, 1982, NRAO has made a number of important advances in computing capability. The AIPS system has been extended and developed, the Pipeline system at the VLA has come into regular operation and a plan has been formulated for the future development of the VLA on-line system. Implementation of the on-line system upgrade has actually begun with the delivery in mid-1984 of the first two CPUs necessary for the upgrade. Several special projects related to computing are also underway.

This section is intended to review the current status of NRAO computing and serve as an update to the documentation prepared for the 1982 meeting of the Computer Advisory Group. We begin with a description of the history and current status of the AIPS project.

2. - AIPS PROJECT REVIEW

2.1 - Introduction

NRAO's Astronomical Image Processing System (AIPS) is a body of programs and subroutines which are presently the main image forming, image processing, and image analysis software for the aperture synthesis interferometers operated by NRAO, especially the VLA. AIPS is a mature project (more than 20 man-years investment, more than 350,000 lines of code produced) whose goals from its inception have been to produce aperture synthesis mapping software within a general purpose, user-friendly, portable system framework. AIPS has evolved to include most of the necessary postcalibration software, to be attractive and intelligible to users and to be modular and reasonably portable (and ported) to a variety of computers and peripherals.

2.2 - Project History

The development of AIPS began in about 1978 for the following reasons. First, the software development and computer power available at the VLA in the late 1970's for map analysis were obviously insufficient; AIPS was intended to be used to analyze images produced by the Pipeline. Secondly, users of the VLA desired a software package which could operate in their university computing systems. This is because achieving the highest quality in VLA data reduction generally requires several iterations of the reduction procedures, and often it is not practical for users to stay at the VLA long enough to finish the process. Export of the existing VLA software (DEC-10 and Pipeline) to university sites was not practical.

Although the AIPS system was originally envisioned as only a map analysis system, it was soon realized that AIPS would need to be able to manipulate visibility data and to be able to make maps from the data. There were three reasons for this need:

i) the VLA site mapping systems were over-burdened,

ii) for an overwhelming majority of projects it is necessary to return to the visibility data in order to either edit out bad data or to revise the calibration and then to construct improved maps, and

iii) it was desirable that AIPS should be able to make maps from Very Long Baseline Interferometry (VLBI) data.

The calibration revision procedures are called "selfcalibration"; the availability of these algorithms in AIPS has resulted in a spectacular improvement in the quality of VLA maps in recent years. The interactive nature of the AIPS design is a key element in the selfcalibration process; AIPS users have become accustomed to exploiting the flexibility which this style permits. The VLBI capabilities of AIPS have gradually increased; in 1984 NRAO decided that AIPS will be the basis for the map making software of the Very Long Baseline Array (VLBA). Thus, as a result of the early decision to support map making as well as map analysis, AIPS has evolved to overlap most of the Pipeline software capabilities, and to form the basis for the software support of the next generation of NRAO instrumentation.

AIPS was designed in Charlottesville in part because the detachment from the everyday problems of the VLA was considered attractive. The project began with only two people who spent almost a year investigating various options. Several other people then joined the effort and the initial implementation was made on a ModComp Classic CPU. A VAX-11/780 CPU was acquired in 1980, and the code was ported to it. The ModComp and the VAX computer systems in Charlottesville contained identical Floating Point Systems AP-120B array processors and IIS image displays and since 1980 the same AIPS code has always run on both of these machines. The resulting necessity to produce portable code has had a profound influence on the AIPS project in many areas, such as design, implementation, policies, and philosophy. NRAO later purchased two more VAXes with APs and displays in order to run AIPS at the VLA.

During 1980 the CLEAN algorithm which had been developed at the VLA for the Pipeline project (in AP microcode) was adapted to the needs of AIPS. Several other mapping, deconvolution and self-calibration algorithms which use the array processors were implemented subsequently. AIPS computers depend heavily on their APs but with a simpler hardware configuration than the Pipeline. AIPS machines consist of a single general purpose computer with an attached array processor. This configuration is very flexible and allows AIPS to take advantage of new developments in data processing hardware and software.

A newsletter for AIPS users, called the "AIPSletter", was started in the fall of 1981 and by 1984 its mailing list had grown to contain over 300 names. A "Cookbook" was developed in 1981 and has become the most popular user documentation for AIPS. A system was developed for automating the handling of software problem reports, called "gripes".

This page is missing from the only print copy available for scanning.

they are satisfactory. The user then issues the AIPS command "GO", which initiates the task as a subprocess, and passes the input parameters to it. When the task has begun, and has acquired the parameters, it allows program AIPS to resume talking to the user. The task is then free to execute in the background, often for periods of tens of minutes or even hours with current computers. The AIPS user can initiate more tasks and perform various interactive operations such as examining the results of previous computations. It is not uncommon for a single user to be interacting with the digital image display (using program AIPS) while three tasks are in progress, with all actions occurring concurrently. There is a prohibition against initiating a second copy of a task while the first copy is still executing. Two or more users may be executing copies of program AIPS and initiating tasks. Thus, multiple copies of the same task may be in execution at the same time, one for each user. It is typical for a machine of the VAX-11/780 class to support two interactive AIPS users fairly well, and five tasks in action at the same time is typical.

A number of AIPS tasks (currently 15) are designed to utilize an AP-120B when one is available. The behavior of these tasks is completely analogous to CPU-only tasks. AIPS tasks which use the array processor do so as a pipelined vector arithmetic unit; data are passed from the host to the array processor, some operation is done on the data, and the result is returned to the host. Because AIPS AP tasks were developed using AP-120Bs, the way in which the AP is called uses FPS conventions. AIPS makes use of microcoded routines in the FPS standard libraries plus a set of custom microcoded and "vector function chainer" routines developed by NRAO programmers. Generally there is only one AP attached to an AIPS computer. Therefore the tasks which depend on the AP are coded to time-share it with a time quantum of a few minutes.

In order to allow the use of AIPS array processor tasks on machines which do not have AP hardware, NRAO has developed the concept of a "pseudo-AP", in which a Fortran common is used as the array processor memory and Fortran or assembly language routines operate on data in this common. These routines have exactly the same names, arguments, and functionality as the corresponding routines in the libraries for the AP-120B. There is only one version of each of the tasks which use the AP; the choice of whether a task uses the true AP or the pseudo AP is made by link editing it with the appropriate subroutine library. The tasks do not know or care, except in the most subtle ways, if they are using a true AP or a pseudo-AP. The "pseudo-AP" concept in AIPS actually amounts to defining a "virtual device interface" for vector processing.

The physical memory size of the AP-120B is effectively limited to 64k floating point words, but this limitation has not been coded into either the AIPS application code or the pseudo-AP code. Instead, the size of the AP memory is a system parameter and most AIPS AP tasks are coded to utilize any and all AP memory which is available.

AIPS needed interactive capability to enhance user efficiency and to permit data analysis techniques in which user intervention is required (e.g. data editing). But AIPS also needed a batch processing capability to enhance computer efficiency for non-interactive applications.

Utilities are provided in the interactive system which facilitate the creation and management of batch jobs. A special version of the command language processor processes the queue of batch command files. The command syntax used in these files is identical to that entered by the users in interactive sessions. In both interactive and batch operations AIPS users need to know very little about the host operating system.

2.4 - User Written AIPS Software

One of the original specifications for the AIPS project was that it be done in a language and a style in which users could write their own applications software. The reason for this was twofold: (i) occasionally a scientist wants a very specialized operation done on his data so he/she should be allowed complete access to the data, (ii) users frequently have good ideas about new, but potentially generally useful, ways of processing their data; in order to tap this pool of talent, these users need to be able to write AIPS software. These reasons are still relevant; in addition, AIPS has become widely adopted for data analysis outside of traditional NRAO areas of research so that non-NRAO programmers are writing AIPS software.

Unfortunately, the requirement of ease of programming conflicts with the requirement of efficient operation and the resultant compromise was biased heavily towards efficiency. For this reason, programming in AIPS has been limited to a fairly small number of programmers and knowledgeable users. This has been a disadvantage. In order to reduce the difficulty of programming in AIPS, three general steps have been taken:

i) Programmer level documention ("Goin AIPS") has recently been make available.

ii) A package of simplified input/output routines has been implemented ("Easy I/O").

iii) A number of paraform tasks have been written for which the user provides a subroutine to operate on his data and can ignore the messier aspects of AIPS programs.

2.5 - Portability

As explained in the historical section above, the AIPS strategy, from its inception, included an assumption that the software would need to execute in more than one hardware/software environment. There were three motivations for this policy:

i) NRAO possessed more than one kind of computer system (ModComp, VAX, IBM, etc.),

ii) NRAO had no assurance that user sites which wanted AIPS would possess computer systems identical to any of those possessed by NRAO, and

iii) NRAO wanted to be able to migrate AIPS to new computers easily in the future.

A variety of examples of all three cases have occurred over the years. Portability has enhanced the value of the mapping software by allowing it to run on a variety of existing machines and it has protected the investment in the software by assuring that it will run on future machines, such as the proposed supercomputer.

AIPS users on a variety of CPUs under various host operating systems all use the same command syntax to invoke the same portable mapping programs in both interactive and batch modes. The portable user command interface is a vital element of the plan to install AIPS on a supercomputer: it means that not only can the mapping programs be implemented easily, but also that the users will be able to use them immediately, because the interactive interface (which is also basis of the batch system) will be exactly what they are accustomed to using. Several hundred astronomers are now trained in using AIPS, and the investment in their training is a substantial part of the whole software investment which NRAO has made in developing AIPS. AIPS has a portable command language precisely so that the skills of its users will be portable.

Recognizing the increasing importance of Unix in the computing industry and recognizing that a university site had produced a prototype installation of AIPS under Unix, in 1982 NRAO formally began a project to develop a generic Unix capability. The goal of the project was and still is "to support AIPS under Unix as well as we support it under VMS". First distribution of Unix installation kits will be for the 15JUL84 release of AIPS (pre-release kits were sent to four test sites in 1983 and 1984).

Several of the image processing peripherals originally chosen by NRAO in 1979-80 are now either no longer manufactured or have become suboptimal choices. More and more user sites have wanted to use AIPS with devices other than the ones which NRAO chose five years ago. Both of these problems have led to a need to interface the AIPS software to a variety of new peripherals. The existence of device independent software interfaces in AIPS has made it possible to prepare, export, and support software interfaces for new devices (support for two new image displays and a laser printer/plotter were added in 1983). Other experimental device interface software exists at various non-NRAO AIPS sites and even within NRAO. Some of this code may eventually become formally supported by NRAO. The development of support for new devices will be a major activity of the AIPS project indefinitely into the future.

2.6 - Development Plans

Several areas in which development work is planned are:

i) improved calibration and editing of visibility data (to be supplied by the VLBI effort in AIPS);

ii) additional spectral line software;

iii) testing and implementation of better deconvolution and self-calibration algorithms;

iv) more sophisticated TV displays, especially for very large images and hypercubes;

v) implementation of table data structures (analogous to spread sheets);

vi) completion of the generic Unix support project;

vii) conversion to the Graphical Kernel System (GKS) for improved portability of graphics device interfaces.

3. - WHATEVER HAPPENED TO THE PIPELINE?

3.1 - Introduction

This is a short review of the history and of the current status of the VLA computer system known as the "Pipeline". The combination of hardware and software referred to by that name was first suggested nearly a decade ago but has just recently come into limited productive use.

The Pipeline system has had a very long development period, has been plagued with hardware, software and management problems and has changed drastically from its initial concept. It has only recently come into use and is not yet fully developed. It is also complex and fragile and requires constant attention from VLA computer staff.

In spite of the development difficulties, the current Pipeline is being used on a regular basis; it is producing good science which otherwise could not have been done; and it has relieved a great deal of the burden on the DEC-10 and AIPS systems at the VLA. It will continue to be a major part of the VLA computing capacitv for the next several years.

Some important lessons can be learned from a review of the Pipeline history. They should be taken into account in the VLA long range computer plans.

3.2 - Genesis

As early as 1973, a planning committee decided that the VLA mapping (imaging) facility would consist of a network of machines rather than a single conventional computer. The nature of the network was not specified. The committee recognized that no digital machine available at that time could cope with the flood of data expected from the completed telescope. It hoped that, before the data inundation began, suitable analog optical devices would be available to handle most of the imaging problem. An important assumption upon which the planning committee operated was that the data emerging from the on-line system would be flagged and calibrated (at least to first order). Although the clean algorithm for removing instrumental effects from the data was known and in use elsewhere at that time, the committee assumed that it would not be needed for VLA data.

A DEC-10 was purchased as an intermediate development facility which could handle, by itself, the mapping of data received from as many as ten antennae. (From 10 antennae one can derive 45 baselines compared with 351 obtainable from the full complement of 27 dishes.) Beyond the 10-antenna level, the DEC-10 was expected to serve as a node in the anticipated (but unspecified) net. A PDP-11/40, supporting various display devices, was connected to the DEC-10 through a terminal port and became, thereby, the first additional node on this net.

The notion that VLA data should be mapped by means of a digital Pipeline synthesized from minicomputers was put forward in a memo written by Barry Clark in 1975 (VLA Computer Memorandum No. 127). During the next two years much effort went into evaluating alternatives, including the use of an optical device. A note by Ehnebuske et al. to the VLA planning committee (1976) proposed a 12-machine minicomputer net which they felt would be needed by the VLA even if the optical processor was successful. The function of each minicomputer was clearly stated in the Ehnebuske proposal but there was no mention of the method of inter-CPU communication.

In 1977 Clark issued a second memo (VLA Computer Memorandum No. 137) recommending a revised form of Pipeline which relied heavily upon array processors controlled by just a few minicomputers to achieve the required computational power at a relatively low cost. (The communication method was still unspecified.) This is the form in which the Pipeline eventually materialized. Not until 1980, however, was the first Pipeline hardware purchased and the first Pipeline software written.

VLA Computer Memorandum No. 137 is remarkably brief; including block diagram, budget and action plan it is only 8 pages long. Its most important aspect, therefore, is the range of subjects that it does NOT discuss. For example, it offers no estimate of the programming man-years that would be required by the project. Likewise, it does not mention the relative role of the Pipeline and the AIPS project that was being planned concurrently (see VLA Computer Memoranda Nos. 140 and 141, 1977).

All in all, it is clear that a very large amount of deliberation and investigation preceeded the construction of the Pipeline. But almost all of it went into systems that were never implemented (e.g. the optical processor). The array processor made the Pipeline so much cheaper and simpler than other alternatives (such as the network proposed by Ehnebuske et al.) that design details were never committed to paper. In addition, the problems associated with communication between units appear to have been consistently overlooked.

3.3 - Prototype

The Clark memorandum suggested that a prototype Pipeline be constructed from a single minicomputer (PDP-11/70) and an array processor (AP120B). The lack of programming staff forced a delay of somewhat more than a year but in 1978 the prototype system, named MAPPER, was brought into production. Within the context of its objectives it was a complete success.

Approximately two and a half man-years of programming went into the initial MAPPER system. Table 1 shows the distribution of this effort by task. The central program was written in FORTRAN IV. The array processor was controlled by calls to subroutines, almost all of which were supplied by the vendor. Only a few special array processor routines were microcoded (by Clark himself) but one of those (CLEAN) was the most critical element in the system. The largest software effort was required for the communication elements whereby data and control information was transfered between the DEC-10 and the PDP-11/70. This component, called

HARVEY, had to be written in assembly language. HARVEY also provided communication with the PDP-11/40 dedicated to display tasks thereby incorporating that machine into the prototype system.

Table 1
Prototype Pipeline Software Effort 1978-1979

Person	Time	Task
Al Braun	9 months	Communications
Dave Ehnebuske	6 months	Communications (DEC-10 interface)
Barry Clark	3 months	Design. Microcode. CLEAN Algorithm
Bob Payne	6 months	Principal programmer
Jim Torson	6 months	Graphic display
Total	30 months	(about 2 1/2 man-years)

3.4 - Demise of the Prototype

For more than three years the prototype Pipeline was used as the primary means of producing images from VLA data. It became a crucial part of VLA computing partly because its hardware was inherently much faster than the DEC-10 but mostly because Clark's CLEAN algorithm was much more efficient than the counterpart on the DEC-10. In some ways the success of the prototype was responsible for delaying the full Pipeline. Because the prototype was so important as a production tool, pressures to improve it on a piecemeal basis diverted manpower from proper implementation of the full system.

The prototype Pipeline remained an essential part of VLA software until the installation of the first AIPS system, at which time it was essentially abandoned. The abruptness with which the MAPPER system was discarded foreshadowed problems that would be faced by the full Pipeline system. Astronomers preferred the AIPS system because:

i) Its communication with the DEC-10 was simple yet reliable (magnetic tape carried by hand). The HARVEY communication software that linked the DEC-10 and the PDP-11s in the MAPPER system had been plagued with bugs (arising mostly from the DEC-10 operating system). More than anything else, the unreliability of the MAPPER system alienated its users.

ii) AIPS incorporated a new computer technique (selfcalibration) for extracting more information from a given set of data. In this respect, AIPS represented a logical progression from the MAPPER system whose initial popularity was due in part to the great improvement of its CLEAN algorithm over the counterpart on the DEC-10.

iii) AIPS was an interactive single-user system. The MAPPER system was a batch system that queued requests from multiple users. Astronomers clearly prefered the interactive mode of operation.

iv) The graphics facilities on the AIPS system were superior to those connected to the prototype Pipeline. In particular, it was easier to obtain larger maps (images) on AIPS. Furthermore, the display facility was an integral part of the AIPS system so that there were no communication problems involved as there were between the mapping and the display components of the prototype Pipeline.

v) The AIPS system was exportable to other institutions so that an observer could begin processing the data for a quick "first look" at the VLA then carry it home for further processing without having to convert the data (or himself) to a new system.

In an ideal world, the full Pipeline would have been redesigned after the introduction of AIPS in order to merge its best features with the advantages (mostly throughput) that a Pipeline could offer. There was, indeed, some review. The most important decision was to adopt DECnet software in place of the homemade communications software. Unfortunately, that decision was undercut by the failure of DEC to deliver a usable DECnet for the operating system (TOPS10) in the central Pipeline node. Furthermore, the fundamental problems that arise from dealing with an inadequate operating system were not fully appreciated.

3.5 - Full Implementation

The main reason for the delay between the design of the Pipeline (VLA Computer Memorandum No. 137, 1977) and the beginning of work on a full version was the lack of programmers to create the software. In 1981, Dr. Wim Brouw, Director of Netherlands Foundation for Radio Astronomy, devoted a year's leave of absence (spent at the VLA) to the sole purpose of writing the full Pipeline mapping software. It was expected that, with some modest amount of help, he could complete the task within the time available.

Unfortunately Brouw had to return to his normal duties before the full Pipeline could be placed in production. At his departure, the only incomplete portion seemed to be the communications link with the synchronous computers. In order to debug his code, Brouw had put together a temporary scheme for reading data from the DEC-10 data base. It was also known that "a few trivial glitches" remained in the code. A member of the permanent VLA programming staff (Bob Payne - who had programmed a large part of the prototype Pipeline) was assigned to complete the project.

As things turned out, the programming effort required to bring the full Pipeline into production has been greater than 10 man-years. The distribution by task is shown in Table 2. There were several reasons for this many-fold increase of actual over estimated programming time. The first reason was that the initial estimate was unrealistically low. The other reasons will be described in detail below.

Table 2
Pipeline Software Effort 1981-1984

Person	Time	Task
Al Braun	2 months	Operating system patches
Wim Brouw	14 months	AP Microcode, Main FORTRAN framework
Barry Clark	3 months	Interface to synchronous system
Phil Dooley	12 months	Special Hardware Construction
Bob Duquet	18 months	Utility routines, Documentation
Miller Goss (et al)	3 months	Validation
Eric Graham	12 months	Task control, Queueing
Bob Kummerer	1 month	DEC-10 interface
Bob Payne	36 months	Principal programmer (all aspects)
Jim Torson	20 months	Graphic display
Total	121 months	(more than 10 man-years)

The length of time required to bring the Pipeline into routine use was bad enough; even worse was the apparent unpredictability of the project. Completion of at least a partial Pipeline was repeatedly announced and repeatedly delayed. The result of such uncertainty was a widespread impression that the Pipeline project was a monumental failure.

Many factors contributed to the difficulty of creating the full Pipeline:

i) Intermittent faults in "home-brew", one-of-a-kind, hardware (the transpose memory) were mistaken for software errors. More than a man year of the most frustrating and demoralizing programming effort went into chasing non-existant bugs.

ii) A concurrent defect in a part of the conventional hardware (the APs) greatly confused the issue. The problem was one of timing and its effects were data dependent. Furthermore, the effect could not be reproduced consistently even with a given data set. Not until the Pipeline had already been severely delayed was this condition identified as a separate problem.

iii) Crucial parts of the Pipeline software involved microcode rather than high level language. Validating this massive and complex code was made abnormally difficult by the absence of the original programmer and by the lack of any flow chart, structure diagram, or almost any other system documentation.

iv) In the microcode refered to above, a strong emphasis was placed on efficiency even at the cost of complexity. As it turned out, the efficiency could not be achieved because of the hardware failures described in item ii) but the complexity remained (and even worsened as the microcode was rewritten to bypass the hardware limitations).

v) Program development tools were inadequate. During the early days of the project the only version of FORTRAN that was available did not contain features to allow structured programming. Despite the presence of reasonable amounts of physical memory the software was restricted to grossly inadequate address space. This added to the system complexity by forcing numerous levels of program overlays.

The combination of these factors led not only to a painfully protracted development period for the Pipeline but also to a demoralizing waste of human resources.

The unfortunate problems with intermittent hardware that were encounterred in the Pipeline can be construed as a piece of bad luck. It was certainly that, but it was more than that too. The Pipeline was not the first computer project designed to achieve a major astronomical objective by being very clever with very small resources. About the time the Pipeline was suggested, an astronomer in Groningen was finally solving a sporadic hardware problem that, for three years, had frustrated his attempt to CLEAN images by doing map arithmetic on an IIS*device. (The problem turned out to be a ground loop on one of the IIS boards.)

The pertinence of this history to the NRAO long range computer plan has to do with the often heard claim that a "clever" approach to computational problems would eliminate the need for costly computational resources. What the claimants inevitably overlook is the indirect costs of such cleverness. The Pipeline history clearly illustrates some of those costs:

i) Vulnerability to unforseen problems.

ii) Uncertainty in scheduling.

iii) Absorption of scarce human resources.

iv) Inflexibility in the face of changing requirements.

The last point, inflexibility, is so very important that it is the subject of a separate section which follows next.

3.6 - Shooting at a Moving Target

In six crucial areas, assumptions made in the design of the Pipeline have been invalidated by later events. These six areas are:

i) The requirement for data flagging.

ii) The demand for user interaction.

iii) The need for sophisticated displays.

iv) The desirability of image enhancement through iteration.

v) The need to interface with many disparate systems.

vi) The usefulness of map arithmetic.

The most important difference between the Pipeline described in Barry Clark's 1977 memo and the system presently in use is the "hands-on" involvement of the observer. As originally envisaged, the Pipeline was an extension of the on-line data-acquisition system which automatically converted raw numeric data into maps (i.e. images). The observer, at a later time, might create modified and enhanced images using a post-processing system such as AIPS but the original Pipeline maps would never be recomputed (much as the correlator output is currently considered sacrosanct).

This vision of the Pipeline assumed that bad data (e.g. interference) could be detected automatically and discarded whenever appropriate. In fact, no algorithm for recognizing contaminated data has yet been developed which can satisfactorily replace an observer's trained eye. Furthermore, certain types of data flaws can only be detected AFTER that data has been transformed into an image and subjected to considerable processing (i.e. CLEANed). The Pipeline has therefore been converted from an automatic batch system into one in which interaction with the user (and with other systems accessed by the user) is an important aspect.

The prototype Pipeline failed to point out the importance of user access to the data. Since all the data (not just calibrators) resided in the DEC-10, it was directly accessible to the user. In the full Pipeline only calibrator data is accessible - the rest is isolated on the Pipeline's own disks.

The second major difference between the Pipeline design and actuality is the existence of algorithms for extracting more information from the data than was previously possible. The Pipeline was designed at a time when the dynamic range in VLA images was expected to be about 100:1; that figure is now 10,000:1. The improvement has been achieved through numerical processing techniques that increase the computing load for image formation by more than an order of magnitude. More important, the new algorithms involved a type of iteration for which the fundamental design of the Pipeline is poorly suited.

The third unforseen circumstance that affects the Pipeline is the existence of an alternate system (including both hardware and software) for creating images from raw correlator data. In some circumstances the most valuable current role of the Pipeline is that of interface between the raw data and the alternate map-making package (AIPS).

The net effect of these three differences is that, in the current operational Pipeline system, at least half of the software performs tasks that were not part of the initial design. An indirect consequence is that the Pipeline is much more complex than initially intended. In fact, the programming effort for the main part of the full Pipeline (Dr. Brouw's work) was a bit more than 1 man-year but 3 additional man-years were required to debug and revise that task, and 5 additional man-years were required for new or auxiliary tasks. (Details were shown in Table 2).

3.7 - Complexity

Data flowing from the VLA correlator through the Pipeline to a final image in AIPS must pass through a bewildering mishmash of computer hardware. The principle components of the Pipeline and the data format in each are shown in Table 3. The interconnection between units is shown schematically in Figure 1.

Table 3
Pipeline Hardware and Data Format

Vendor	Device	Word Length
ModComp	ModComp	16 bit integer (*)
Century	Disks and Controller	
DEC	DEC-10	36 bit (packed 18-bit integer)
DEC	ANF-DECnet converter	
DEC	PDP-11	16 bit integer (*)
Century/Emulex	Shared disk/Controller	
FPS	AP-120B	38 bit floating point
Dataram	DATARAM	24 bit scaled integer
Telex/IPS	Tape drives/Controller	16 bit scaled integers
DEC	VAX 11/780	32 bit floating point

(*) Byte order is opposite in these two 16-bit machines

The contrast between the block diagram in Memorandum No. 137 and Figure 1 is fairly indicative of what has happened to the Pipeline concept. While the earlier diagram contained 7 blocks, the current diagram contains 30.

The magnitude of the problem of maintaining Pipeline hardware can be inferred from the mix of vendors listed in Table 3. There are actually more vendors listed in Table 3 than there were blocks in the original diagram. All of the CPU's (except the ModComps) are under contract for maintenance by DEC but this does little to solve the problem. Whenever anything goes wrong, a substantial amount of the VLA's own very limited resources (especially system programming time) must be expended on problem determination before the provisions of that contract can be invoked.

Appearances to the contrary, the Pipeline hardware is NOT the result of trying to see how many vendors could be represented in a given system. Each purchase was dictated by the unavailability of a counterpart from the original computer vendor (DEC) or by funding inadequate to the purpose. For example, DEC did not supply a 6250bpi tape drive for PDP-11 machines until quite recently and the disk space required by the Pipeline would have cost twice as much had it been obtained from DEC.

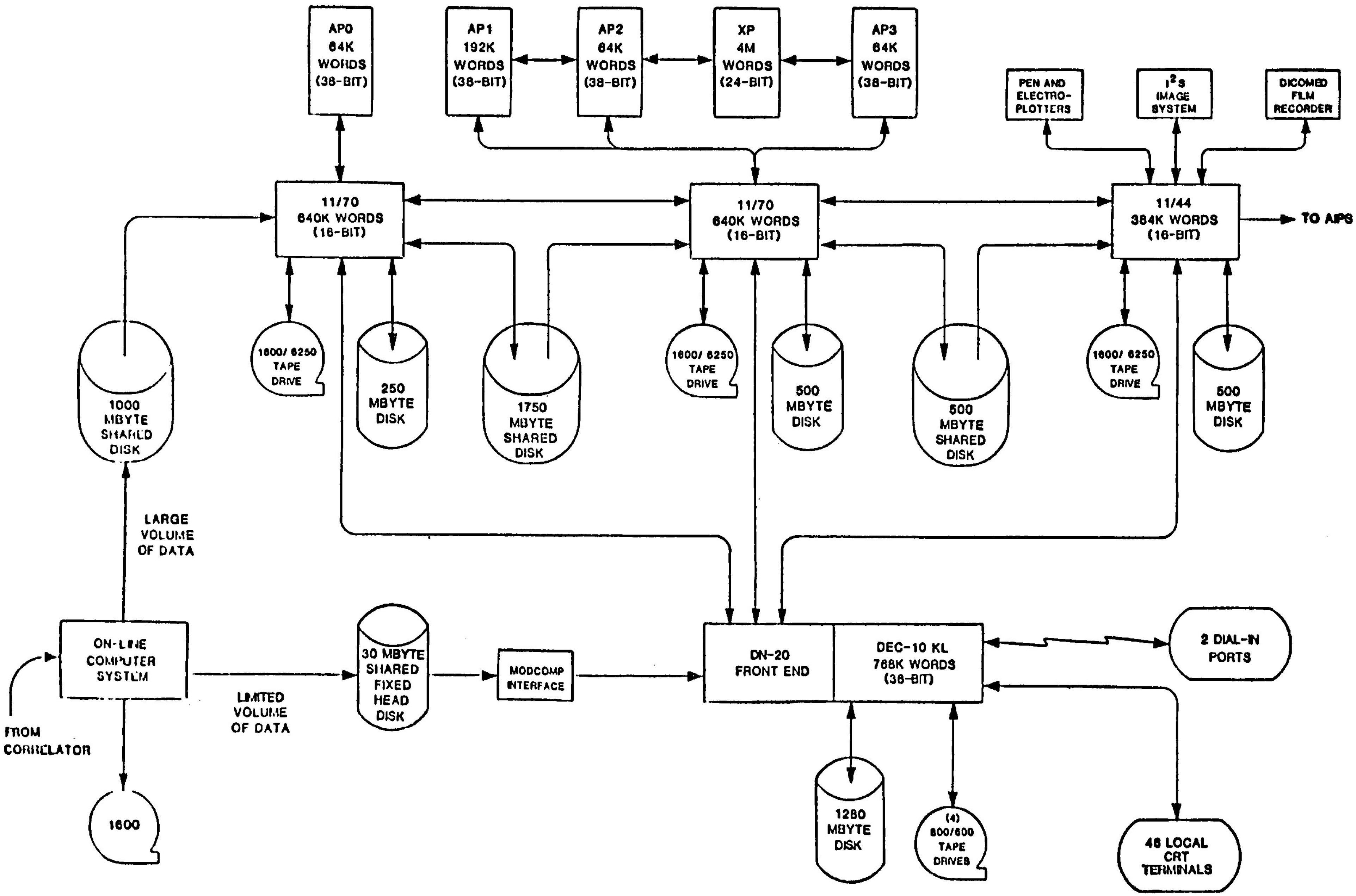
AP0
64K
WORDS
(36-BIT)
AP1
192K
WORDS
(36-BIT)
AP2
64K
WORDS
(36-BIT)
XP
4M
WORDS
(24-BIT)
AP3
64K
WORDS
(36-BIT)
PEN AND
ELECTRO-
PLOTTERS
I^2S
IMAGE
SYSTEM
DICOMED
FILM
RECORDER
11/70
640K WORDS
(16-BIT)
11/70
640K WORDS
(16-BIT)
11/44
384K WORDS
(16-BIT)
TO AIPS
1000
MBYTE
SHARED
DISK
1600/ 6250
TAPE
DRIVE
250
MBYTE
DISK
1750
MBYTE
SHARED
DISK
1600/ 6250
TAPE
DRIVE
500
MBYTE
DISK
500
MBYTE
SHARED
DISK
1600/ 6250
TAPE
DRIVE
500
MBYTE
DISK
LARGE
VOLUME
OF DATA
ON-LINE
COMPUTER
SYSTEM
FROM
CORRELATOR
LIMITED
VOLUME
OF DATA
30 MBYTE
SHARED
FIXED
HEAD
DISK
MODCOMP
INTERFACE
DN-20
FRONT END
DEC-10 KL
768K WORDS
(36-BIT)
2 DIAL-IN
PORTS
1600
1280
MBYTE
DISK
(4)
800/600
TAPE
DRIVES
48 LOCAL
CRT
TERMINALS

Figure 1

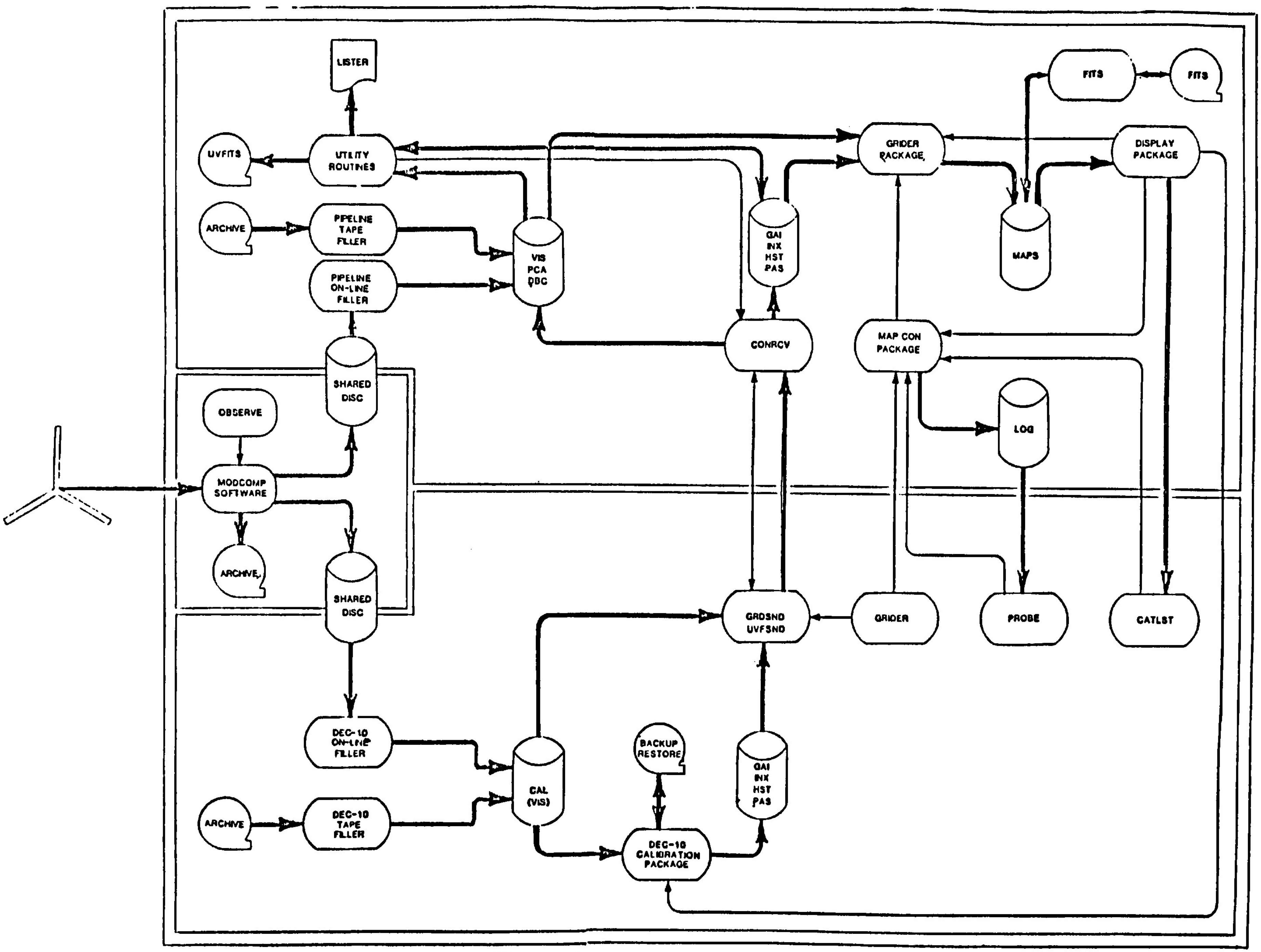
LISTER
FITS
FITS
UVFITS
UTILITY ROUTINES
GRIDER PACKAGE
DISPLAY PACKAGE
ARCHIVE
PIPELINE TAPE FILLER
VIS PCA DBC
GAI INX HST PAS
MAPS
PIPELINE ON-LINE FILLER
CONRCV
MAP CON PACKAGE
SHARED DISC
LOG
OBSERVE
MODCOMP SOFTWARE
ARCHIVE
SHARED DISC
GRDSND UVFSND
GRIDER
PROBE
CATLST
DEC-10 ON-LINE FILLER
BACKUP RESTORE
CAL (VIS)
GAI INX HST PAS
ARCHIVE
DEC-10 TAPE FILLER
DEC-10 CALIBRATION PACKAGE

Figure 2

3.8 - Current Status

Early this June (1984) the Pipeline was made the default path through which VLA spectral line data would be processed in the absence of specific instructions to the contrary from the observer. This decision was prompted more by necessity than by desirability. It was anticipated that the observations scheduled for this summer would completely swamp all other computing facilities. In the sense that the current Pipeline is filling an essential need, it is a success.

The Pipeline has turned out to be very good at the task for which it was initially intended: making large numbers of images. It is an efficient mapping engine which can create maps 3 or 4 times faster than the only alternate system available (AIPS). It takes the Pipeline approximately 8 minutes to produce a typical group of 8 spectral line maps, 512 pixels on a side, from approximately 100,000 visibility records. (This time is totally dominated by I/O.)

The ability to make maps quickly (if not in the greatest achievable refinement) is especially valuable for a "quick look" at spectral line data and snapshots where many maps are involved in a given experiment. Part of the value of the Pipeline to spectral line observers comes from avoidance of cumbersome database conversions that are required on the DEC-10 (e.g. the SPECTR program).

Many desirable features are still lacking in the Pipeline. A "wish list," compiled at the time the Pipeline was made the default data path, contains four pages of one-line items some of which involve many man-months of effort. The other side of the coin is that the Pipeline already provides some capability absent from other systems - for example a graphic presentation of raw data (BTMAP) in which errors can be detected more readily than by any other means. The Pipeline is also the only available means of processing large (i.e. 4k x 4k) maps.

Quite apart from what the Pipeline does or does not offer in its own right, it has been a success in decongesting the DEC-10 computer which had become a painfully tight bottleneck for all VLA processing. This relief came in three forms: CPU requirements on the DEC-10 were reduced (because spectral line data did not need to be transformed to meet the antiquated requirements of the DEC-10 database structure), the crunch on disk space was reduced (because the Pipeline provided additional space and used it more effectively), and I/O bottle necks were relieved (because the Pipeline offers a more efficient gateway to the AIPS system and to the outside world in general).

Hindsight reveals, with its customary clarity, that the Pipeline design was far too optimistic in the following expectations:

i) It overestimated the extent to which I/O between the APs and their host machines could be overlapped with, hence hidden behind, computing time.

ii) It did not take into account the practical limitations on massive data transfer between machines. Especially, it overlooked the effects of limited buffer space and the correspondingly small I/O block size.

iii) It expected to utilize too great a percentage of the theoretical power of the Array Processor. It overlooked the problem of keeping the AP busy when only a small amount of memory was available to hold data.

As a result of these miscalculations the performance of the completed system is much less than anticipated. But loss of throughput is a relatively minor problem; a running system can be accelerated by simply adding more or faster hardware. The more significant problems are those that arise from system instability.

The fragility of the current Pipeline system is a direct result of its complexity. There are too many hardware components which must all be running at the same time, there are too many elaborate database systems in which exceptional conditions must be recognized for proper data transfer, there are too many points of user access at which correct input must be supplied and there are too many user-controlled parameters which mesh into too many possible internal modes of operation. A very large part of the wish list, referred to above, consists of measures to "ruggedize" the system but adding "fixes" to a system that is fundamentally over-complex can only increase that complexity.

3.9 - Lessons Learned

Three basic lessons that should be remembered while drawing up long range computer plans emerge from our examination of the Pipeline experience. They are these:

i) Requirements Grow.

ii) Systems Degenerate.

iii) Complexity Costs.

4. - SUMMARY OF THE VLA ON-LINE COMPUTER DEVELOPMENT FROM MID-1982 TO MID-1984

4.1 - Overview

The on-line system is designed to perform observations with the VLA under the control of the array operator. The observations begin after the operator is supplied with the current observing list. The system performs the necessary calculations to determine the configuration of the hardware and appropriate commands, such as antenna coordinates and wavefront delay, are sent to the relevant modules. While the observation is in progress, monitor information is continuously being returned to the computer and this is used to check the state of each component of the instrument. This information is used to make an automatic assessment of the data quality as well as to inform the operators and the responsible technicians of any malfunction. Errors are reported immediately both on hard-copy and color video displays, and they can be further investigated using an interactive display system.

The signals from the antennas are converted to correlation coefficients by the correlator, which may be regarded as a special purpose computer whose configuration and performance is controlled and monitored by the on-line system. This data is passed from the correlator through an array processor where some preliminary corrections are made, to the on-line system, where quality information is appended, and then output to an archive tape. The data is also left on mail-box disks for retrieval by the calibration and mapping system.

The status of the hardware and software of the on-line system at the time of the last Computer Advisory Group Meeting is briefly summarized in documentation distributed to the group in March, 1982. There have been significant changes during the last two years in the on-line system development. They fall into two categories: modest changes to the existing system and the development and implementation of a specific long range plan.

4.2 - Changes to the Existing System

There have been a small number of significant changes made to the existing software system to accommodate enhanced observing capabilities. Only those tasks which were considered critical and which could be accommodated were implemented. The list of features implemented is by no means sufficient for getting the most science out of the instrument nor is it complete.

Some of the major milestones are:

i) The amount of data processed has been doubled by turning on two more receiver outputs and increasing the on-line processing to double the spectral line capacity;

ii) A new receiving system has been brought on-line;

iii) The control software has been integrated into one package (the continuum and spectral line system are now one) greatly facilitating one aspect of the array operator's duties;

iv) A substantial increase in the number of possible hardware options selectable by the observer has been made allowing many more different types of spectral line experiments;

v) A number of small features to aid VLA operating staff in control and monitoring of the hardware performance have been installed; and

vi) conversion to floating point numbers.

4.3 - Upgrade of the On-Line System

The NRAO has developed a plan for, and begun implementation of, the next generation on-line control system. The limitations of the current hardware and software system, the additional current and future requirements and the possible options have been outlined in documentation presented to the Computer Advisory Group at its last meeting and in two VLA internal memoranda: "Upgrade of the VLA On-Line Computer" by G.Hunt and K.Sowinski (VLA Computer Memorandum No. 166, February, 1983) and "VLA On-Line System Upgrade" by G.Hunt and K.Sowinski (VLA Interoffice Memo, November 14, 1983). The decision has been made to remain with ModComp as the manufacturer and replace the existing computers with the newest and fastest that ModComp offer; i.e. the Classic series.

The current hardware system consists of 8 computers networked together to provide both real-time control and monitoring of all the relevant hardware and real-time data collection. The networking software was developed within NRAO. The long range plan is to replace the four principle ModComp Model II computers with modern ModComp Classics which have more speed, I/O capacity and addressable memory. A more modern operating system (and compilers) will be available for this hardware. Most of the current software will be transferable to the new hardware with a minimum of effort (although non-trivial).

The first installment of the hardware (two ModComp Classics with appropriate peripherals) has been received. The additional CPUs and peripherals will be purchased next year with the goal for completion set for 1987. Work has begun both in the area of converting the existing software to the new system and in converting as much as is practical of the software to FORTRAN 77, a necessary step to make the system more maintainable.

5. - SPECIAL PROJECTS

5.1 - Introduction

NRAO, through its Electronics Division, maintains a small but state-of-the-art digital electronics capability. This manpower is used both for the development of various telescope-related electronics and for the ongoing support of digital apparatus including computers. This latter category includes both maintenance and interface work. On occasion, special computer-related devices are designed and built either because they are not available in the marketplace or because the device is a simple modification of an existing NRAO designed apparatus. A few such devices are described here. Their description serves to give a feel for the general level of the Observatory's limited involvement in computer hardware development.

5.2 - Mass Storage Unit

NRAO has for years been involved in the development of high density digital magnetic recording through its involvement in VLBI. The Very Long Baseline Array (VLBA) project requires digital recording of 100 megabits per second continuously at each of 10 antennas along with a similar playback capability. Modest error rates are tolerated (1 per 10,000). For this purpose systems using both instrumentation recorders and modified video recorders have been developed. In the latter approach several recorders are used, each at a lesser data rate. Also, the read/write electronics is replaced to effect a much higher recording density.

This same device has been developed for the storage of VLA and other data. The error rate is improved by using a number of digital techniques. In such a way, it is possible to place 2.5 gigabytes of high quality data on a single video cassette. This is equivalent to nearly 60 1600bpi conventional tapes or 20 6250bpi tapes. This development is currently being tested and should be in use before the end of the year.

The development has been kept somewhat unsophisticated because we are not in the mass store business and because it is considered somewhat of an experiment. Also, it is recognized that optical disk technology will probably replace it in a few years.

5.3 - Image Storage Unit

The standard hardware configuration for the AIPS-based image processing systems is a host CPU with its conventional peripherals, an AP, and an image display/processor. The image display normally works from the host's disk, and has two to four 512 x 512 image planes eight bits deep. When working with images of 512 x 512 pixels, no more than a few can be held in the processor itself at one time. For a new image, it is necessary to access the host disk. This can involve a load time of several seconds depending on the host I/O load.

A number of applications occur, however, where movie type displays are useful. To achieve this, we have developed an image storage unit with a high transfer rate to the image display. The storage unit is made up of two Winchester technology disks with a special controller designed to combine the data stream from the two disks in order to double the transfer rate. A 512 x 512 x 512 image cube can be stored and images transferred to the host at a rate of four images per second.

The hardware for this device has been completed along with simple software integration. Application software, however, is not very far advanced.

A special control panel is also being developed, for use with this device, which will enable the user to have simple analog control without the use of a keyboard.

5.4 - Digital Switch

A digital switch which supports switching crt terminals to various hosts or to the intersite digital communication system has been developed. This is similar to switches which are manufactured and sold for this purpose but is somewhat more tuned to the Observatory's needs. The switch performs well, and is less costly and more flexible than available commercial units. The switch is currently in use at our Charlottesville site; copies are now being developed for use at the other sites. The system is very well matched to our needs. Nevertheless, it is clear that developments of this type should only be undertaken after considerable investigation of commercial units available.

6. - GENERAL COMPUTING

Historically, all NRAO general computing was done in the Observatory's computer center. This system was originally located in our Green Bank facility and later moved to Charlottesville. The system has always made use of an IBM mainframe and most of this time fit the stereotype of a batch-oriented closed shop.

General computing began to diversify in the late 1970s. The development of the VLA in New Mexico and the existence there of the DEC System-10 was one cause. The IBM was clumsy for local use at the VLA both because of its distance and lack of local peripherals and because of its batch orientation. The DEC-10 by comparison was close and had a very nice interactive system. The trend toward distributed general computing continued as VAXes came on the scene. The VAXes were considered by many users a more attractive machine than either the DEC-10 or the IBM. With the development of AIPS (the Astronomical Image Processing System) on VAXes, new investment in hardware and software was concentrated on these machines. Also, once the astronomer became familiar with the VAX for his image processing, he naturally continued to use it for his other work.

The result of all this is that today general purpose computing is distributed among the IBM, the DEC-10, and 3 or 4 VAXes, as well as a few smaller machines. The following table gives a rough breakdown of the distribution of general computing. Scientific computing here refers to miscellaneous computing by staff and not to standard data reduction systems.

Type of Work	Location of work	Machine
Scientific computing	Charlottesville	IBM, Charlottesville VAX
Scientific computing & Local Eng. Support	VLA Site	DEC-10, VLA VAX
Scientific computing & Local Eng. Support	Tucson	Tucson VAX
Scientific computing & Local Eng. Support	Green Bank	IBM, Mass Comp
Engineering Structures	Charlottesville	IBM
Electronics Central Lab.	Charlottesville	IBM, Charlottesville VAX
Electronic Mail	All Sites	All VAXes
Technical Word Processing	All Sites	VAXes, Various PCs

If the NRAO developed a large computer facility this system would almost certainly replace the existing IBM, the DEC-10 and the Pipeline. From a purely operational point of view, it would be attractive to require this same system to absorb all of the Observatory's general purpose or miscellaneous computing. At present, however, it is not clear to what extent this is possible. The remoteness involved and the corresponding cost and difficulty of working over communications lines are factors. Another factor is the general batch orientation of these large machines. Except for its limited capacity and I/O, a VAX makes a very good image processor. If VAXes or other superminis are kept for image processing they will undoubtedly be used for these other activities. On the other hand, it is clear that the cost of operating a very large system will be so large that it is unlikely the NRAO will be able to afford to keep many of the smaller systems.

This area needs more study once the characteristics of the large computer and the communication system are better specified.