VLB ARRAY MEMO No. 278

General Organization of VLBA Control Software

B. G. Clark
October 1983

This memo sets out some general principles upon which the software for the control and monitor system will be constructed. It will not excessively concern itself about the hardware upon which the system will run, and for the moment will leave in limbo a number of controversial matters that still must be properly discussed. Having listed the limitations, it is surprising that anything can still be found to be discussed.

The memo will, contrary to the remarks above, start off with a section of comments and desiderata on hardware, and then proceed to a discussion of general software principles, and then to a brief description of the specific programs envisioned.

1. Remarks on Hardware

Let us begin with a bit of notation. Each VLBA station will have in it a computer (size and other properties yet to be specified) whose job is (among other activities) to establish communication between the various hardware devices with which the station is equipped and the outside world. This computer will be referred to as the 'station computer' or 'station control computer'. The 'outside world', referred to above, is a single computer at the array operations center. This computer will be called the 'array control computer' or 'central control computer'. If there is distributed intelligence at the station other than the station control computer, these devices will not be dignified with the name 'computer', but will be called controllers or control units (eg Antenna Control Unit).

If possible, two or three copies of the station control system should be purchased as soon as it is specified. In this way, software development, interfacing, and installation of the computer into the first antenna can proceed in parallel with a minimum of interference.

The philosophy used at the VLA seems to me to be the correct one, that the station control computer will output "commands" at appropriate times, and expect that the devices to which they are addressed to be ready to accept and execute them. I also incline to the VLA custom that, if the device does not "roger" or "wilco", the computer should ask for human help, instead of blindly retrying in hopes of inducing compliance.

A cardinal principle of device design should be that the station control computer should be able to examine both the last command sent to the device and the complete status of the device itself. Both have their uses in debugging.

All communications, possibly except those within a single room and static parallel lines, should be protected by parity, checksum, or CRC, and there should be a mechanism of reporting parity errors to the station computer.

The station control computer should have access to the system clock. It should be interrupted at a regular rate. The particular rate depends on what we have the computer doing. If it does the conversion to az-el coordinates, the 20 Hz rate used at the VLA is about right--it is sufficiently above the antenna resonant frequencies that updating antenna pointing at this rate should produce no problems. If it does not have this task, a slower rate is acceptable; however, I would not like to see it as slow as once per second. On the other hand, if it has the job of blanking pulsar data, a 200 Hz rate would be more appropriate (incidentally, a particularly elegant method of blanking would be to substitute an even paritiy pattern during unwanted times). It seems to me unacceptable that the station computer not have access to the system clock (it being connected instead to the pointing controller). The loss of flexibility for adding future time scheduled items might be crippling. The interrupt times will be referred to as ticks. The station computer should be able to read the clock, in ticks as well as in the conventional hours, minutes, seconds. It is possible that we might want the computer to be able to write the clock as well, though only hours, minutes, seconds, NOT ticks. Setting the clock is a bit more touchy than the sort of thing I usually would entrust to a computer, but it seems to me an attractive way to handle leap seconds. The other possibilities--having all ten station maintainers on the job at midnight UT to push buttons, or keeping track of when the buttons were in fact pushed at each station and requiring the correlator to run some stations with an integer second offset until all of the clocks were reset--also have fundamental disadvantages. Of course, if the station computers must set the clocks at the stations, the array control computer must also have a clock, accurate to 0.1 second or so, so it can tell the stations what time it is.

A final remark about hardware. One of the constant annoyances at the VLA is having technicians leaving an antenna with some critical device in manual control. Something should be done at the stations to make this a bit less likely. Possibilities include having manual control switches be timers, so that manual control expires after an hour; having manual control switches be key switches, with an SOP that specifies that the keys are to be on the same ring as the car keys; having manual control switches ground a logic bus that causes a large yellow light (perhaps 200 watts) to flash at the entrance to the control room. I favor the latter.

2. Software General Principles.

The VLA Modcomp software update procedures are very effective and appropriate for an on-line system. The first job after choosing the hardware on which the system will run will be to get an appropriately modified version of this software update procedure on line.

Communication software should bought instead of written where at all possible. I would hope that we could take it to the point that we could have multiple logical links operating on one physical link, with error correction and alternate path routing concealed from the user interface. It seems acceptable to me for these features to be implemented in software within the station CPU or in an external statistical multiplexor. This sort of capability is necessary, is available on the market, and is most unlikely to cost anything like as much money as it would take to develop it ourselves. The importance of this point cannot be overemphasized.

A capability of rebooting over the communications link is highly desireable. The alternatives are to commit it to ROM, which makes updating somewhat cumbersome, or to give the communications link the capability of causing the computer to boot from a local disk. In my eyes the most desireable is to have both the first and third alternatives.

Uninteruptable power for the station computer RAM is probably necessary for some hours. We are inclining to leaving monitor data lying around the station computer until called for by the array control computer. It would be most undesireable if major disasters capable of causing blown fuses would automatically erase all monitor data that might tell why they happened.

The station computer operating system must be a multitasking system with several priority levels. The convenience of writing tasks without having to worry about what is happening in other tasks is well worth the cost.

I propose to continue the general philosophy used at the VLA, that the program system is to be organized around a fairly large common data area available to all programs, and that one program will have the responsibility to set each datum, but all others can read and use it. The common area will consist of four primary parts. The first will be called the common control area. This will contain things having to do with time (for instance, the formula for LAST-UTC, trig functions of sidereal time, perhaps precession and nutation matrices), a few observation independent control bits (for instance the focus heater on/off bit, recorder control information, and tape use information), and a collection of pointers into various parts of the common area. The second part will be called the observation control block, and will essentially be the VLA control blocks concatenated into one unit. Several copies of this will be provided (I think four or five might be about right), so that a simple cycle of observations can be executed without having to go through the process of organizing one of these core blocks. The third part will be an in-core table of all monitor point values. The considerations that dictated doing this in the VLA (limited monitor system bandwidth and its autonomous nature) do not really apply here, but this organization has been very successful as a means of separating functions into simpler, quasi independent blocks. Finally, the fourth part of common will consist of buffers containing observation requests received from the array control computer and monitor data to be sent to the array control computer.

As much as possible, there programs running in each station control computer should be identical. Differences between the various stations should, as much as possible, be handled by tables of equipment present, maintained centrally and sent to the station computer by the array control computer as part of the process of initializing the station computer. This will probably suffice for most specializations, but a provision for loading special tasks, or non-standard versions of standard tasks, on an station by station basis, should be provided, while hoping that we never have to use it.

3. Software Organization

In this section I shall list the programs which we shall clearly need, grouped by the computer in which they will run.

3.1. Programs in The station Control Computer

THE ANTENNA DRIVER. This program might send az-el to the antenna controller at a 20Hz rate, or might send more sophisticated stuff at a slower rate. The VLA uses a linear extrapolation for ten seconds, with the full spherical triangle solved only every 10 seconds. It would seem more likely that the spherical triangle would be solved at the full 20Hz rate for the VLBA antenna. This eliminates one level of tasking and the concomitant handshaking. The price is about ten or fifteen percent of the CPU, either for a CPU with floating point hardware, done in Fortran, or for a CPU without floating hardware, done in assembler scaled binary fixed point.

THE NEW SOURCE EXECUTOR. This program would be responsible for switching between two observation control blocks, mentioned above. The program would primarily make sure that all of the receiver switches are thrown to the correct position.

NEW SOURCE ORGANIZER. In order to permit more observation requests to reside in the in-core buffer of requests received from the array control computer, the latter would be sent in some rather condensed form, and expanded to make an observation control block. There is a lot to be said for sending the observation requests as text, rather than binary. It makes it possible to examine them from anywhere in the system in a rather easy fashion, and, if care is taken, to constructively interfere with them.

MONITOR DATA INHALER. This program would maintain the core image of monitor data, as mentioned above.

MONITOR DATA LOGGER. This program would sample the monitor data core image at appropriate intervals and store it into a buffer, with appropriate identifying information, still in the station computer.

TAPE SYSTEM CONTROLLER. This program would do the bookkeeping about how much tape is available, when to switch tapes with minimum disruption, and generally supplies any information that the tape subsystem needs.

PHASE CALIBRATION EXTRACTOR. This I know from nothing, but somebody says something should be done about it.

DATA SENDER/RECEIVERS. These include Observation request receiver, monitor log sender, possibly a fast monitor data sender (see below), a fringe check data sender, and a real-time remote debugger. Off hand, I do not see why the station computer would ever need to initiate communication, which might simplify things a bit.

3.2. Programs which could be written for station computer or array control

DEVICE CONTROL PROGRAMS. These are the equivalent of the VLA Modcomp DMT overlays. I feel that they could run in either the station computer or in the array control computer, so long as a terminal can access them from the other computer using the dedicated computer link. I do not feel that the station staff/visiting maintenance staff would be adequately supported by having the programs run in the array control computer and accessed by a separate, dial up, modem.

DATA FLAGGER. This would automatically set two flags--antenna off source (including subreflector not set), and LO chain malfunction--based on the monitor data. The latter should be overrideable; the VLA experience is that the monitor equipment is at fault nearly as much as the LO chain itself. My favored implementation would be to run this program in the station computer and have it insert these two bits into the Mk III type data header blocks. The correlator would then recognize these bits, and not correlate data when they are set (with the LO chain bit overrideable by input from the correlator control computer). It is a bit of a bother to have the data flagger merely make entries in the array control computer log, which is then sent to the correlator control computer for execution.

MONITOR DATA CHECKER. This program would notice out-of-range monitor points and call them to the attention of the array operator. At the VLA, this program and the above are combined into a single package. The advantages of this have been less than expected, and do not, for instance, constitute a constraint that the programs must run in the same computer. The VLA version of this program is written as an interpreter, which does have some advantages, but I wouldn't do it again on a Modcomp (though the call is close enough that the decision can be swayed by the nature of the computer which is to run it).

3.3. Programs for the array control computer

SENDER/RECEIVERS. These programs match the programs which run at the station computers and accomplish the transmission of data. Depending on which particular program, they would be initiated on a regular cycle or by the operator.

MONITOR DATABASE FILLER. This program would put monitor data sent from the various stations into a conveniently accessible disk file for access by the array maintenance engineers.

MONITOR DATABASE PRUNER. Because of the lack of the capability of putting a chart recorder on a monitor point directly that we have, and use, at the VLA, I would make a very much denser monitor data base for the VLBA, an order of magnitude more voluminous. This dense database I would keep only for a couple of days, and then make a roughly VLA style monitor database that prunes the original down to a manageable size.

ARRAY LOG WRITER. This program is likely to be a bit more complicated than at first appears, since it should be able to cope with any station being out of communication for hours, and still be able to produce a nicely time ordered log file ready for submission to the correlator control computer.

STATION COMPUTER INITIALIZER. Unless the station computer runs from ROM this would include the down-line load. It would also send such stuff as date, time, location, tables of equipment present, etc.

MONITOR DATA PLOTTER/LISTER. Equivalent to the VLA programs Monplt and Monlst. Probably should be available over the link to the stations, though in a real pinch, accessability over a dialup might be acceptable.

REAL TIME FRINGE CHECK. Receives data from the fringe check buffers from the array and does the correlation. On a VAX 750 this program would probably require several seconds per baseline and per lag. This is no real problem, unless we lose a station and have to go hunting for it over many tens of lags, or unless we decide we want to do this on such weak sources that all baselines must be processed and global fringe fitting done. Incidentally, this program will be providing a delayed input for the log writing program, which will again complicate its life.

OBSERVATION PLANNING AID/OBSERVATION REQUEST GENERATOR. The VLA has been threatening a major revison in Observ for a year now. It seems likely that this could serve as a basis for the program for the VLBA scheduling. It is also clear that additional auxiliary functions will be needed, such as display of rise/set times at various stations, (u,v) tracks, etc.

4. Other considerations

4.1. Remarks on languages

Despite the fact that this is just the sort of application for which Ada was designed, Fortran 77 is the language of choice during the current phase of activities. This decision will be reevaluated when actual hardware and communication software have been chosen. Software for the array control computer should be almost entirely in Fortran, with a few assembler subroutines (for instance screen formatters and, for the fringe checker, a correlation routine). If the station computer has no floating point hardware, substantial parts of its program must be in assembler (or language other than Fortran) because of the slowness of software floating point and the clumsiness of Fortran for scaled fixed point binary operations. In any event, one of the requirements of the station computer supplied software should be a Fortran with an easy escape to assembler.

4.2. Main memory

The main memory requirements for the station computers seem to be surprisingly moderate. For instance, it might go with 32K for a real time operating system, 32K for programs (including their internal buffers), 32K for common control blocks, 32K for observation and monitor data buffers. Thus, 128K, expandable to 256K, looks right for the station computer. If the manufacturer does not provide a reentrant Fortran library, the overhead of including the Fortran library in each task would probably call for increasing this to 160K, expandable to 320K. Still, a rather modest computer by today's standards. It should be noted that these are ball park estimates only, and should be looked at more carefully when the particular station computer is being chosen.

A wild guess at the memory requirement of the array control computer would be 100K for operating system and networks, plus 100K for each user, up to the maximum the system might be expected to support. I suggest that that number might be two programs bringing monitor and log data back from array stations, an array operator, two people calling in from stations to run diagnostic programs, two programmers, two observers preparing source files, and a batch job. One MByte looks right, probably with a requirement for expandability to two.

4.3. Peripherals

The proper equipment of the station computer is yet to be decided. I throw out the following for discussion.

- 1 miniWinchester drive
- 1 floppy drive (possibly dual)
- 1 screen (say VT 100 equivalent)
- 1 spare screen interface
- 1 hard copy terminal (say Quietwriter equivalent)
- 1 modem/dedicated line

o

1 modem, dialup
interfaces to station hardware as required

The programs can be organized so that both incoming and outgoing buffers are both left in core and are written to the Winchester disk. The disk going down would then affect primarily only the time the station can survive without telephone service. If DMT type overlays are run in the station computer, they would probably reside on the disk, so losing the disk might also lose the overlays. The floppy drive would be primarily used for software interchange and for maintenance (eg garbage collection) of the Winchester. It might also be a backup residence for DMT software. The modem on the dialup line should have password protection--lets not have the 414 Gang stealing our antennas. The spare screen interface would be for the use of a visiting maintenance engineer, who might want to bring something special. A wild thought is that enclusion of a voice synthesizer might be a handy thing for coordinating tests between array control and the station, and for handling unauthorized intrusions.

The array control computer should have two or three tape drives, for log and monitor data archiving/retrieval, observing file backup/restore, software maintenance, and disk maintenance. Data rates are relatively modest, so they need not be dense or fast. Disk requirements might total 100 MBytes for system and software development needs, 100 Mbytes for observing programs (old as well as current) and log files, and 200 MBytes of monitor data. I suggest eight local screens (including at least one graphic), a dialup modem reserved for contacting stations which have lost their dedicated line, three other dialup modems for remote preparation of observing files, and at least 4 spare ports. We must have a printer/plotter on this system.

4.4. Maintenance

It looks to me as if the station computer is a sufficiently simple machine that routine preventive maintenance could be done "hot", without stopping the machine. This would mostly be filter changes, fan maintenance, physical inspection, disk file garbage collection. It would appear to me appropriate to schedule a day or two a year for something more thorough. Otherwise, it would probably be adequate to have only on-call maintenance from the manufacturer's service organization nearest the station. CPU failure quick fix would be to ship a spare from the array operation center by air, as would the special interfaces. Failure of any other component does not put the station off the air, and can be borne for days. In fact, the station computer is so small and simple, it is not clear to me that we shouldn't take it in to the service man, rather than having him make a house call.

Whatever the array control computer is, it is likely the observatory will have another reasonably near at hand (that is, the DEC-10 and the IBM do not look like good candidates). With this as a backup to provide essential services, it seems to me that no special provision need be made for a quick response to an array control computer failure.

5. Order of implementation

The first item is to investigate communication links, since these are such an expensive operations item. Once the choice of link is made, we should be in a position to choose the communications software package, which may well dictate what CPU it would like to run on. Only after the choice of CPU can significant amounts of software be written, though I would think at least a third could be well specified independently of the CPU. The emphasis from the first should be to have a remotely operating system, so the communications handlers will need to receive early attention. Within that framework, though, the lowest level (ie closest to the station harware) programs should be done first. Antenna pointing and DMT type programs would be developed early on, and a full blown set of new source prorams later. Monlst would come later still, and last of all, a version of Observ particularized to the VLBA.