VLB ARRAY MEMO No. 307

Interoffice

National Radio Astronomy Observatory
Very Large Array

30th December 1983

To: Hein Hvatum

From: Gareth Hunt

Subject: Programming Language for the VLBA project

I am concerned about rumors that I have heard concerning the computer language(s) that will be adopted for the programming of the correlator computer for the VLBA. We made the decision some time ago at the VLA that all future software development should be in standard FORTRAN where possible. I strongly propose that the VLBA project should make the same decision for all phases of its operation: the on-line system, the correlator computer, and the data reduction computer.

Here is a brief summary of some of the major arguments for and against some of the major options.

1. One well known fact of programming is that the productivity of a programmer, measured in lines of debugged documented code, is almost independent of the language in which he is programming. In higher languages such as FORTRAN, PL/1, ALGOL (SAIL), C, PASCAL, etc. fewer lines of code are necessary, in general, to produce the same functionality when compared with lower level languages such as Macro Assemblers.

2. It must be easy for NRAO to maintain and enhance any software over the lifetime of the VLBA project. Languages such as FORTH and Macro Assemblers (and APL) are much more difficult to maintain than others because they are more difficult to read. Everyone forgets what they wrote a month or so ago, let alone a few years ago. Programs are easier to maintain, therefore, if they are written in an easily readable language not only by the programmer himself, but also to someone else called upon to take it over at a later date.

3. Most radio astronomers have to program computers to reduce their data. In the vast majority of cases, the language that they use is FORTRAN. A few of the younger generation have been brought up to use C (especially under the UNIX operating system) and this may well be the way for the future, but NRAO must support the large majority.

4. I do not know of any major programming system that has been written in PASCAL or ALGOL. This may just be a statement of ignorance, but certainly none is available in astronomy. PL/1 is used very heavily on IBM computers but, although it is available on some other computers, it is not used heavily elsewhere. My knowledge of C is

limited, but I have a feeling it is better for system work and casual programming than for use in a major programming system.

5. There is a great advantage in using a language which was designed for the job, namely scientific programming such as FORTRAN or PL/1. COBOL was designed for business data processing, C for systems work, PASCAL as a teaching language, and FORTH as a stand-alone operating system.

6. Since we will never be in the business of writing and supporting our own operating systems, it is extremely important that we use a language that is well supported by the hardware vendor using the vendor's operating system. SAIL is not; nor are, in general, PL/1 (except on IBM computers), ALGOL, PASCAL, or C. FORTH is not even compatible with use on most operating systems. FORTRAN is, however, generally available on all machines that would be considered for use with the VLBA.

I would also like to emphasize some of the points made above by a brief summary of NRAO's experience in several areas.

A. The VLA on-line (synchronous) system. Here some programming was done in FORTRAN but the majority of the code is in Macro Assembler. On the ModComp II computers we were unable to write the majority of the code in FORTRAN mainly because of the limited memory size and also, in some cases, the execution speed. Obviously the really time-critical tasks and such things as device drivers should be written in Macro, but in a modern machine with unlimited address space, the majority of a real-time system should be written in a high level language. There will be a greater use of FORTRAN on the upgraded on-line system.

B. The VLA asynchronous system. Here we chose a language (SAIL) that was preferred by the programmers mainly for two reasons a) the availability of many of the features of PL/1 and b) the rather poor version of FORTRAN then available on the DEC-10 computer. The SAIL compiler is not a standard product and is no longer maintained by anyone on a routine basis, so some of our problems can never be addressed. The fear of an unknown language by the scientific staff has led to there being virtually no direct programming input from these people into the system and I believe this has significantly hurt the project. In retrospect, the choice of FORTRAN would have been worth the early frustrations.

C. AIPS. The success of AIPS has been remarkable, and here FORTRAN was chosen. It has been made to run on many machines and the scientific staff have written programs to run under the system. AIPS is written in FORTRAN 66 not the current standard, FORTRAN 77, but it will, almost certainly, have to be upgraded to a more modern standard at some date. It would be my mild personal preference to invest the effort to upgrade AIPS to FORTRAN 77 before the huge addition of code necessary for VLBA support. I understand, however, that the decision is likely to be to wait for the next standard (FORTRAN 8X).

D. Tucson on-line system. Here we chose FORTH. Although it has been used successfully, we no longer have the compiler for it, and reading the code is tortuous. If the present staff were to leave, we would have an almost impossible job to maintain or upgrade it. The reduction system is written in FORTRAN and is being moved from a PDP-11 to a VAX. The other package (NOD2) being implemented in Tucson is also exclusively in FORTRAN.

I am therefore left with FORTRAN as the recommended choice. AIPS is already in FORTRAN, and since astronomers will be involved in the programming of the correlator system, (3.) would indicate that this too should be on FORTRAN. In view of (2.) I think there is a very good case for extending the development stage of the project, if necessary, to reap the benefits of long term maintenance.

Copies:
Bob Burns
Ron Ekers