VLB ARRAY MEMO No. 308

CALIFORNIA INSTITUTE OF TECHNOLOGY
MS 105-24, Pasadena, CA 91125
Telephone 818-356-4970
Telex 675425 CALTECH PSD, Telecopier 818-796-8751

To: VLBA Memo Series Date: 16 Jan. 1984

From: Martin S. Ewing

Subject: VLBA Programming Languages

Gareth Hunt (VLBA Memo 307) is concerned that (1) there should be a common language for the whole VLBA, (2) the life-cycle cost, including maintenance, should be considered. Hunt has proposed Fortran, probably Fortran-77, as the standard.

While agreeing that languages should not proliferate needlessly, I feel that a strong emphasis on a standard language may be inappropriate. This is a viewpoint that came from a recent correlator working group meeting, supported by the following observations:

(1) There is a large investment in Forth and Pascal programs for control and fringe processing, respectively, in the Block II Correlator; we hope to transfer these as much as possible to the VLBA.

(2) It is not clear that any one language is suitable for the full range of VLBA work. Fortran does well for AIPS processing, but would not be preferred (by us) for real-time multiprocessing. Forth, C, and Pascal are all more attractive than Fortran for this work. (Data structures, pointer variables, and interactiveness are important in this work.)

(3) Native VAX Forth, as developed by JPL, is a civilized language; it is much easier to work with and maintain than older 16-bit versions. (Dave Rogstad is preparing listings for Craig Walker and others to examine.)

(4) The VLBA is a large project: it is not expected that the same programmers will maintain all VLBA code. It is reasonably efficient to have both non-Fortran and Fortran programmers working on the project. The user community's knowledge of Fortran is not especially relevant, since users would not be expected to deal with antenna or correlator control code in any way.

(5) The cost of maintenance has more to do with the complexity of the system than with whether a standard language is used. In fact, selecting a non-standard language that easily expresses the particular computational problem can REDUCE the maintenance effort.