THE INTERACTIVE VISUAL FORTRAN INTERPRETER

Douglas P. Witmer

Stuart C. Shapiro

Computer Science Department

Indiana University

Bloomington, Indiana 47401

TECHNICAL REPORT NO. 6

THE INTERACTIVE VISUAL FORTRAN INTERPRETER

DOUGLAS P. WITMER

STUART C. SHAPIRO

NOVEMBER 1, 1973

Appeared as Chapter 3 of: Stuart C. Shapiro and Douglas P. Witmer, Interactive visual simulators for beginning programming students, Fourth Symposium on Computer Science Education, SIGCSE Bulletin 6, 1 (February 1974), 11-14.

The Interactive Visual FORTRAN Interpreter

Douglas P. Witmer

Stuart C. Shapiro

Computer Science Department

Indiana University

Introduction

Learning to program in a higher level language such as FORTRAN can be a formidable task for the beginning student. He must not only memorize the key words and statement syntax associated with the language, but must learn to assemble meaningful programs from these building blocks. In this regard, it is essential for the student to be able to visualize the step-by-step operation of his programs. The Interactive Visual FORTRAN Interpreter (IVF) was designed as an aid to such a visualization.

Basic Description

The IVF accepts a FORTRAN program and graphically simulates its execution on an interactive time-sharing terminal with a cathode ray tube (CRT) screen. During the simulation the executable statements are displayed on the screen. An arrow is caused to point at each statement in the order in which execution would normally take place. Current values of variables are displayed in the unused space to the right of the program, and are updated with each move of the arrow. Thus the immediate state

of the program is visually apparent at all times in terms of current values of variables and flow of control.

```
      I = 0                          I:  1
  4   I = I + 1
->    X = (I ** 2) / 2               X:  0.4999999999
      IF(I .LT. 10) GO TO 4
      END
```

Figure 1

Figure 1 shows a short program as it would appear during simulation on the IVF. The arrow indicates that execution of the statement "X = (I ** 2) / 2" is being simulated. The current values of "I" and "X" are displayed on the right side of the screen. In this example the arrow will next move to the "IF" statement. Simulation of the "IF" statement succeeds and the

arrow moves to the statement whose label is "4". Now the value of "I" is incremented and the new value is displayed on the right. The simulation proceeds in this manner until execution of the "IF" statement is simulated with the current value of "I" equal to 10. At this time the arrow moves to the "END" statement and the simulation is terminated.

Upon termination of the execution phase, the IVF displays an appropriate message and the user is given the option to exit the IVF or attempt another simulation. The FORTRAN program just simulated is not lost in either case. The user may have it simulated again (as discussed later) or dispose of it in some other fashion.

The IVF assumes a naive user, and is therefore self-explanatory. Instructions may be received at the beginning of each session by responding appropriately to the program's inquiries. In addition, all portions of the program requiring a user response will recognize errors and prompt as necessary.

Although the IVF is primarily an aid to the visualization of FORTRAN program execution, it is also an interactive FORTRAN statement accepter with a number of error checking facilities. The IVF will accept FORTRAN lines and continuation lines until a complete statement has been entered. It then checks for syntax errors, errors in "DO" loop nesting, and multiply defined statement numbers. If the IVF detects an error, the statement is rejected and an appropriate error message is

displayed. The user is thus made aware of his mistake immediately and is given another opportunity to enter a correct statement.

Additional Features

A user may optionally bracket a portion of his FORTRAN program by placing a "+" in the continuation field of a non-continuation line to turn the "STEPPED MODE" on. The "STEPPED MODE" is turned off in an analogous fashion with a "-". Within the bracketed section the user must press the "RETURN" key after the execution of each FORTRAN line has been simulated. This permits a slowing of the simulation over a portion of the user's program for more careful study.

The IVF saves a copy of the FORTRAN program simulated, and makes it available on a file reserved for that purpose. The user may optionally submit complete programs to the IVF rather than enter them one line at a time. Error checking is performed on each statement as with on-line entry. The complete program may have originated from any source, including a session with the IVF.

FORTRAN programs submitted to the IVF may be edited with an internal editing feature, and subsequently simulated. Transfer to the editor may be evoked during an IVF session whenever the user finds it necessary to correct errors in his programming logic.

Implementation

The IVF is written in the SNOBOL4[1] programming language. Each FORTRAN statement accepted by the IVF is translated into a sequence of SNOBOL statements. Steps are taken to adjust operator precedence and make type conversions as necessary. This is bracketed by statements which move the arrow and update displayed information during the simulation. This block of SNOBOL statements is concatenated into a string of such blocks for later compilation, utilizing the CODE function.

Simulation of a FORTRAN program is accomplished by first displaying its executable statements on the CRT screen, and then executing the compiled SNOBOL statements discussed above. During simulation the IVF displays output from the FORTRAN program on the bottom line of the screen, and accepts input on the line above. The position of the executable FORTRAN statements on the CRT screen is never changed. The CRT's cursor (which underlines the next printable position) is moved about as necessary to prevent erasure of valuable information and to facilitate input and output.

---

[1] Griswold, R.E., Poage, J.F., Polonsky, I.P., The SNOBOL4 Programming Language, second edition, Prentice-Hall, Englewood Cliffs, N.J., 1971.

Summary

Visualization of the execution of his programs is of utmost importance to the beginning programmer. It is hoped that the IVF will serve as an effective aid to such visualization. It is further hoped that the IVF will be found flexible in its features and yet simple enough for use by a beginning programmer.

The IVF currently runs under the KRONOS Time Sharing System on a CDC6600 computer using an Applied Digital Data Systems, Inc. ADDS Consul 880 terminal.

It can readily be adapted to run on a number of computing systems which support time-sharing terminals and provide interactive execution. The primary restrictions are the need for a CRT display at the time-sharing terminal and a sufficiently large character set to support control of the display.

The IVF is written in a straightforward manner with numerous comments. An experienced programmer should, therefore, be able to expand the scope of the IVF to suit the needs of his local users. He might even cause the IVF to simulate some language other than FORTRAN.