SOLOWAY83
Soloway E., Bonar J., and Ehrlich K.; Cognitive Strategies and Looping Constructs: An Empirical Study; CACM 26, 11 (November 1983).

SYNOPSIS

In this paper the authors test the hypothesis that people will find it easier to program correctly when the language facilitates their preferred cognitive strategy. In particular they compare the performance of programmers using different looping constructs, the 'while' and 'repeat' of Pascal versus the 'do .. exit when .. od' construct.

The authors use as their test case the following problem, "Write a program that repeatedly reads in integers until it reads the integer 99999. After seeing 99999, it should print out the correct average. That is, it should not count the final 99999." and identify two different strategies :

```
1) process/read:    READ ( first value )
                  WHILE ( test i'th value )
                     PROCESS ( i'th value )
                     READ ( i+1'st value )

2) read/process:    DO
                     READ ( i'th value )
                     EXIT WHEN ( test i'th value )
                     PROCESS ( i'th value )
                   OD
```

They feel that the process/read strategy puts an extra cognitive burden on the programmer since the 'process' and 'read' within the loop are out of sync. However the process/read strategy fits well with the Pascal while loop, whereas the read/process strategy can only be coded in Pascal (without gotos) by using additional code. To gather data about programmer performance a group of students was given a two part test, first write an English plan for solving the above problem, second write a program to solve the problem. In the second half of the test half of the students used standard PASCAL and the other half a language Pascal L which is Pascal without 'while' and 'repeat' but including a 'exit when' type loop construct. The authors used this data to test the following hypotheses.

1) Which strategy do people naturally use?

80% were read/process according to their plans written in English.

2a) Will people write correct programs more often when using the language that facilitates their preferred strategy?

Yes, 52% wrote correct programs in Pascal L whereas only 33% wrote correct programs in Pascal.

2b) Irrespective of whether a strategy is preferred or not, will people write correct programs more often when using the strategy facilitated by the language?

Yes.

3) Does preference for strategy vary with experience?

Yes, experienced Pascal programmers tended to pick the process/read strategy when programming in Pascal.

The authors then cite some other studies which suggest that the readability of the two different constructs is the same. E.g. a pure loop construct (with the exit at the top or bottom) is just as readable as a loop construct which allows exits in the middle.

COMMENTS

The paper clearly adds evidence to the hypothesis that programming language constructs should be closer to how people naturally specify problem solutions. The results can also be applied to the older languages (i.e.; FORTRAN, COBOL) and attempts to impose a structured style on these languages. For example in COBOL the preferred read/process strategy cannot be easily coded without gotos. The process/read strategy (which can be coded without gotos) requires an extra read statement before the loop. Perhaps one of the reasons that the data processing industry pays only lip service to structured programming in COBOL is that structured programing in COBOL requires abandoning a naturally specified problem solution.