1985

Structured Tools and Condiitonal Logic: an Empirical Investigation

Iris Vessey
University of Queensland

Ron Weber
University of Queensland

Structured Tools and Condiitonal Logic: an Empirical Investigation

Iris Vessey and Ron Weber
University of Queensland

ABSTRACT

An important outcome of recent work on the psychology of programming has been the recognition that we have a poor understanding of how various programming practices—indenting, commenting, naming, etc.—facilitate or inhibit the programming process. After a fairly extensive series of studies, many results obtained are contradictory and counterintuitive. The major probem seems to be that we have poor theoretical bases to drive the empirical research. In particular, we have little knowledge of the psychological constructs that programmers bring to bear when they perform various programming tasks, and we have little knowledge of what is "natural" for programmers.

This research tested the propositon that the effectiveness of a programming practice is a function of the extent to which it provides a close cognitive fit with a programmers' problem solving strategy when he or she performs a programming task. The proposition was tested in the context of two psychological processes that appear to be used by programmers when they design and code conditional logic: (a) taxonomizing—identifying the conditions that evoke particular actions; and (b) sequencing—converting the taxa to a linear sequence of program code. Three structured tools—structured English, decision tables, and decision trees—were investigated in a laboratory setting to determine how they facilitated these two processes. It was hypothesized that decision tables and decision trees would facilitate the taxonomising process because they allow conditions and actions to be easily identified, and that structurd English would facilitate the sequencing process because it provides a linear representation of logic that can be mapped easily into programming code.

To test the hypotheses, 124 volunteer information systems and computer science students undertook three experiments. In the first experiment they were given a narrative description of some conditional logic and asked to represent the logic using one of the three types of structured tools. In the second experiment they were given conditional logic already represented via one of the tools and asked to convert it into COBOL code. In the third experiment they were given a narrative description of some conditional logic and asked to convert it into COBOL code after having first represented the logic using one of the three types of structured tools. Their perfomance was assessed in terms of the number of syntactic errors they made, the number of semantic errors they made, and the time taken to perform the experimental tasks.

In general, the results confirmed the propostions investigated. When the taxonomizing task had to be undertaken, decision trees outperformed strutured English, although surprisingly structured English outperformed decision tables. When the sequencing task had to be undertaken, structured English outperformed decision tables, but decision trees evoked the same level of performance as structured English. Across all tasks, decision tables evoked relatively poor levels of perfomance. On the other hand, decision trees evoked high levels of performance across all tasks. It appears that the graphical tree structure allows taxon information to be represented poignantly. At the same time it appears relatively easy to trace a branch to its leaf node to perform the sequencing task. The superiority of decision trees seems to confirm the desirablity of graphically revealing the structure inherent in processes rather than using symbolic languages. Moreover, the results suggest that the syntax of current programming languages may be unnecessarily restrictive. Perhaps programming languages should provide decision trees as part of their syntax instead of providing only unidimensional, linear syntax to represent conditional logic.

Dubin, R., *Theory Building*, Revised ed., New York: The Free Press, 1978.

Ericsson, K.A. and Simon, H.A., "Verbal reports as data", *Psychological Review*, Volume 87, May, 1980, pp. 215-251.

Ericsson, K.A. and Simon, H.A., *Protocol Analysis: Verbal Reports as Data*, Cambridge, Massachusetts: The MIT Press, 1984.

Gane, C. and Sarson, T., *Structured Systems Analysis*, Englewood Cliffs, New Jersey: Prentice-Hall, 1979.

Gildersleeve, T.R., *Decision Tables and Their Practical Application in Data Processing*, Englewood Cliffs, New Jersey: Prentice-Hall, 1970.

Gould, J.D., "Some Psychological Evidence on How People Debug Computer Programs", *International Journal of Man-Machine Studies*, Volume 7, March, 1975, pp. 157-182.

Gould, J.D. and Drongowski, P., "An Exploratory Study of Computer Program Debugging, *Human Factors*, Volume 16, May-June, 1974, pp. 258-277.

Heller, J.I. and Greeno, J.G., "Information Processing Analyses of Mathematical Problem Solving", *National Science Foundation Seminar on Applied Problem Solving*, 1978.

Johnson, P.E., Hassebrock, F., Duran, A., and Moller,J., "Multimethod Study of Clinical Judgment", *Organizational Behavior and Human Performance*, Volume 30, 1982, pp. 201-230.

Mayer, D.B. and Stalnaker, A.W., "Selection and Evaluation of Computer Personnel—the Research History of SIG/CPR", *Proceedings of the 23rd ACM National Conference*, 1968, pp. 657-670.

Newell, A. and Simon, H.A., *Human Problem Solving*, New York: Prentice-Hall, 1972.

Nilsson, N.J., *Principles of Artificial Intelligence*, Palo Alto, California: Tioga.

Pennington, N., "Cognitive Components of Expertise in Computer Programming: A Review of the Literature", Technical Report No. 46, University of Michigan, 1982.

Reilly, R. *et al.*, In *The Use of Expert Judgment in the Assessment of Experiential Learning*, CAEL Working Paper No. 10, 1975.

Reitman, W.R., *Cognition and Thought*, New York: Wiley, 1965.

Sheil, B.A., "The Psychological Study of Programming", *Computing Surveys*, Volume 13, March, 1981, pp. 101-120.

Sheppard, S.R., Curtis, B., Milliman, P., and Love, T., "Modern Coding Practices and Programmer Performance", *Computer*, Volume 12, December, 1979, pp. 41-49.

Shneiderman, B., "Exploratory Experiments in Programmer Behavior" *International Journal of Computer and Information Sciences*, Volume 5, June, 1976, pp. 123-143.

Vessey, I., "An Investigation of the Psychological Processes Underlying the Debugging of Computer Programs", Unpublished Doctoral Dissertation, University of Queensland, 1984.

Vessey, I. and Weber, R., "Research on Structured Programming: Empiricist's Evaluation", *IEEE Transactions on Software Engineering*, Volume SE-10, July, 1984, pp. 397-407.

Winer, R.J., *Statistical Principles in Experimental Design*, 2nd ed., New York: McGraw-Hill, 1971.

Young, F.W. and Lewyckyj, R., *ALSCAL 4 User's Guide*, Chapel Hill, North Carolina: Data Analysis and Theory Associates, 1979.

Youngs, E.A., "Human Errors in Programming", *International Journal of Man-Machine Studies*, Volume 6, November, 1974, pp. 361-376.

Yourdon, E. and Constantine, L.L., *Structured Design*, Englewood, New Jersey: Prentice-Hall, 1979.