



**PROCEEDINGS**  
**INTERNATIONAL CONFERENCE**  
**ON**  
**CYBERNETICS AND SOCIETY**

**October 8-10, 1980**

**Sponsored by**

IEEE Systems, Man and Cybernetics Society

Hyatt Regency, Cambridge, Massachusetts

# HUMAN MEMORY PROCESSES FOR COMPREHENDING COMPUTER PROGRAMS

A. F. Norcio  
Applied Science Department  
United States Naval Academy  
Annapolis, Maryland 21402

While programmer behavior research has supported the importance of logic segments in the human memory organization of computer programs, no direct attempt has related comprehending computer programs to logic segment organization. The purpose of this study was to examine experimentally logic segment chunking and program comprehension. Using the Cloze procedure which requires subjects' supplying missing statements, the results suggest that it is easier to supply correct statements within a logic segment than at the beginning. The results also suggest that internal program documentation and statement indentation tend to enhance the memory/comprehension processes. It is concluded that highlighting logic segments may improve these processes.

## INTRODUCTION

In his classic paper concerning the magical properties of the number seven, Miller introduced the term chunking into psychological literature. This term is used to describe the human memory processes of "organizing or grouping input into familiar units or chunks."<sup>1</sup> A good example of this process is the natural cognitive function of linguistically recoding letters into words, words into sentences, sentences into paragraphs and paragraphs into ideas. The chunking hypothesis further suggests that the first element of a chunk provides the key to the contents of the entire unit.

Subsequent studies have supported Miller's contention that memory organization/linguistic recoding is the essential ingredient in the thought processes. There is ample evidence that a sentence is organized and recoded in human memory by its phrase structure segments. Johnson has shown that more recall errors occur at phrase structure boundaries than within phrase segments.<sup>2</sup> Foder and Bever have also noted the significance of syntactic boundaries as linguistic segments for deep structural associations.<sup>3</sup>

The importance of human memory organization in computer programmer behavior has also been suggested by several recent studies.<sup>4,5,6</sup> Using a memory recall approach, Norcio found that subjects tended to recall computer program statements according to the logic hierarchy of the program's algorithm.<sup>5,6</sup> From this, he has suggested that a program's logic segments play a similar role in the memory organization of a program as phrase structure segments perform in the linguistic recoding processes natural language sentences and paragraphs.

Internal program documentation and statement indentation have been proposed as important factors in the design of well written programs. However, controlled experimental studies have not observed strong evidence which decisively supports their benefits on programmer performance. With respect to program statement indentation, studies have found that novice programmers prefer indented programs. Surprisingly, results from several different studies have

found no significant indentation effects on program recall tasks.<sup>5,7,8</sup>

These same studies have observed moderately significant documentation effects on program recall. In general, it appears that documentation at the beginning of a program module produces better recall than documentation interspersed between statements throughout the program.

While computer programmer research has been insightful, no direct attempt has been made to relate the comprehension of computer programs to the chunking of the programs' logic segments. The purpose of this study was to examine experimentally the relationship between chunking of logic segments and the comprehension of the program's algorithm.

While the memory recall methodology is appropriate for examining chunking processes in programming, a different tact may be more appropriate for examining comprehension. Norcio has suggested the Cloze procedure as possibly a valid and reliable technique for measuring comprehension of a program's algorithm.<sup>9,10,11</sup> With this technique, blank lines are substituted for specific program statements. The subjects' tasks then become supplying correct statements for the blank lines. Using this procedure, it was predicted that it would be easier to supply correct statements within logic segments as compared to statements at the beginning of segments. Since statement indentation and internal documentation provide clues to a program's logic organization and its meaning, groups with these aids were compared to groups without them. Two separate experiments were conducted. The first experiment examined the effects of internal documentation on memory organization and comprehension. The second investigated the effects of that program statement indentation on these two factors.

## EXPERIMENT 1

### Subject

Fifty computer programming students participated in groups of ten students each: 1) one panel, and 2) four experimental conditions. The four experimental conditions represented two documentation levels (ND-no documentation; BD-beginning documentation) and two location levels (BS-beginning segment; WS-within segment).

### Materials

Five Fortran programs which varied from twenty to thirty-three statements were used. These programs were appropriately complex for student programmers. A sample program is included in the appendix. Copies of all the programs may be received from the author.

Procedure

The panel was used to define the logic segments in each program. A consensus of opinion decided the logic segments in each program.

Internal documentation was incorporated at the beginning of each program given to the BD groups. The ND groups had no documentation in their programs. The BS groups had blank lines substituted for program statements at the beginning of the programs' logic segments. The WS groups had blank lines substituted for program statements within the programs' logic segments.

Subjects were randomly assigned to each of the four experimental conditions. Programs were presented to the subjects in random order to that order effect would be balanced. The subjects were instructed to study the first program and supply correct versions of the missing statements. They were further instructed to repeat this process until all five programs had been examined. Subjects were tested individually and had thirty minutes to complete the experiment.

The numbers of correctly supplied statements in each program was recorded for all subjects.

Results and Discussion

A 2 x 2 multivariate analysis of variance (MANOVA) was computed with documentation and blank line location as the independent variables and the numbers of correctly supplied statements for each of the five programs as the five dependent variables. The results generally indicated that groups with initial documentation and blank lines within segments supplied significantly more correct statements. Specifically, there was a significant multivariate interaction effect ( $F=4.0, p<.05$ ) in favor of the BD/WS group. There was a significant documentation effect ( $F=6.25, p<.05$ ) in favor of the DB groups. And there was a significant multivariate location effect ( $F=4.0, p<.02$ ) in favor of the WS groups. There were also several significant univariate effects between the various programs.

The results of the first experiment suggest that programmer comprehension is enhanced when the logic segments are well organized in memory. Psychological theory indicates that the first element of a chunk provides the key to its entire contents. The findings of this experiment support this contention in the context of computer programs. The two groups whose missing statements were within segments supplied significantly more correct statements than did the other groups. This suggests that groups whose initial statements were missing had more difficulty in comprehending the functions of the segments.

Table 1

Average Number of Correct Statements

	<u>BS</u>	<u>WS</u>
ND	10.6	8.6
BD	8.8	14.0

Subject

Forty computer programming students participated in groups of ten students each in four experimental conditions. The four conditions represented two indentation levels (NIN-no indentation; IND-indentation) and two location levels (BS-beginning segment; WS-within segment). These subjects did not participate in Experiment 1.

Materials

The five programs which were used in Experiment 1 were also used in this experiment.

Procedure

The same logic segment definitions which were used in Experiment 1 were also used in this experiment.

The programs given to the IND groups had their statements indented according to the logic hierarchy of the definitions. The programs given to the NIN groups had no indentation at all. The BS groups had blank lines substituted for program statements at the beginnings of the logic segments. The WS groups had blank lines substituted within logic segments. In this experiment no groups were given internal documentation.

Subjects were assigned randomly to the groups and the experiment was conducted in the exact same manner as Experiment 1.

Results and Discussion

Both the analysis and results of this experiment paralleled Experiment 1. A 2 x 2 MANOVA was computed with indentation and blank line location as the independent variables and the numbers of correctly supplied statements in each of the five programs as the dependent variables. The results generally indicated that groups with indentation and blank lines within segments were able to supply significantly more correct statements. Specifically, there was a significant multivariate interaction effect ( $F=3.8, p<.05$ ) in favor of the IND/WS group. There was a significant multivariate indentation effect ( $F=6.40, p<.02$ ) in favor of the IND groups. And there was a significant multivariate location effect ( $F=15.0, p<.001$ ) in favor of the WS groups. There were also several significant univariate effects between the various programs.

The results of the second experiment parallel the first experiment's results. Again the groups with missing lines within segments supplied significantly more correct statements than the other groups. Again these results are consistent with psychological theories of memory organization.

The indentation seemed to serve as an indication of logic segment organization which the subjects were able to transfer to memory. As a result, indentation probably facilitated the comprehension processes.

Table 2

Average Number of Correct Statements

	<u>BS</u>	<u>WS</u>
IND	11.6	15.4
NIND	10.2	11.2

## GENERAL DISCUSSION

These experiments have extended the findings of other studies concerning the memory and comprehension processes involved in computer programmer behavior. The present study has provided evidence that memory organization of a computer program's logic segments plays an important role in the comprehension of a program's algorithm, the results suggest that internal documentation may provide some assistance in comprehension. More importantly, the findings indicate that statement indentation may offer substantial assistance in comprehending computer programs.

In appreciating this study and its results, it is important to distinguish between memory organization and comprehension. While both are fundamental cognitive processes, they are nonetheless quite distinct, memory organization is the natural human ability to propensity to categorize information in memory according to a consistent scheme. Comprehension is the process of extracting semantic concepts underlying organized information. Presumably, it is quite possible for the human to organize information in memory without comprehension occurring. On the other hand, information must be organized in memory according to some scheme before comprehension can be successful.

Further it is important that the methodological approach employed in human/computer interface studies be consistent with the cognitive construct that is under investigation. While the memory-recall approach has been deemed appropriate in memory organization studies, the Cloze procedure provides a more realistic alternative for examining comprehension.

Using the Cloze procedure the pattern of results from both experiments suggest the importance of logic segment memory organization in the comprehension of computer programs. In both experiments, significantly more correct statements were supplied within segments than at the segment beginning. This finding is entirely consistent with the chunking hypothesis which specifies that the first element of a chunk provides the key to the contents of the entire unit. The results which have also supported the relationship between program begin segments and human memory organization.

With respect to comprehension the results suggest that statement indentation provided more assistance to the comprehension processes than documentation. This is especially surprising since studies, using a recall approach, have not observed significant indentation effects.

However, a plausible explanation can be advanced for these apparent discrepancies. The recall methodology is an experimental paradigm for measuring memory organization. Its suitability for measuring comprehension can be questioned. The Cloze procedure presumably provides a cleaner measure of comprehension.

If memory organization is a natural process, it is reasonable that visual cues as indentation may not be helpful. However, indentation may provide enough visual information which facilitates the comprehension processes.

A practical outcome of this investigation is that program logic segments should be highlighted as much as is practical in order to facilitate the comprehension processes. Indentation provides the earliest and most direct method to accomplish this goal.

## REFERENCES

- <sup>1</sup>Miller, G. A. (1956). The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information. Psychological Review, 63, 81-97.
- <sup>2</sup>Johnson, N. F. (1965), The Psychological Reality of Phrase Structure Rules. Journal of Verbal Learning and Verbal Behavior, 4, 468-475.
- <sup>3</sup>Foder, J. and Bever, T. (1965). The Psychological Reality of Linguistic Segments. Journal of Verbal Learning and Verbal Behavior, 4, 414-420.
- <sup>4</sup>Shneideman, B. (1976). Explanatory Experiments Into Programmer Behavior. International Journal of Man Machine Studies, 5, 123-143.
- <sup>5</sup>Norcio, A. F. (1978a). Memory Organization and Computer Program Logic. Paper presented at the Annual Meeting of the Human Factors Society, Detroit, MI.
- <sup>6</sup>Norcio, A. F. (1978b). Chunking and the Recall of Computer Programs. Paper presented at the Annual Meeting of the ACM, Washington, DC.
- <sup>7</sup>Love T. (1977). Relating Individual Differences for Computer Programming Performance to Human Information Processing. Ph.D. Thesis, University of Washington, 1977.
- <sup>8</sup>Shneideman, B. and Mayer, R. (1979). Syntactic/Semantic Interactions in Programmer Behavior: Analysis and Results. International Journal of Computer and Information Sciences.
- <sup>9</sup>Norcio, A. F. (1979a). The Cloze Procedure: A Methodology for Analyzing Computer Program Comprehension. Paper presented at the Annual Meeting of the ACM, Dayton, Ohio.
- <sup>10</sup>Norcio, A. F. (1979b). Factors Affecting the Comprehension of Computer Programs. Paper presented at the National Computer Conference, New York, NY.
- <sup>11</sup>Rankin, E. F. (1957). An Evaluation of the Cloze Procedure as a Technique for Meaning Reading Comprehension. Ph.D. Thesis, University of Washington.

## APPENDIX

### Sample Program

```
DIMENSION X(100)
READ 10,N
10 FORMAT(13)
READ(5,15) (X(I),I=1,N)
15 FORMAT(F5.2)
M=N-1
DO 20, X=1,M DO20 I=1,M
    J=I+1
    DO 30 K=J,N
        IF(X(I).GE.X(K)) GO TO 30
        T=X(I)
        X(I)=X(K)
        X(K)=T
    CONTINUE
30
20 CONTINUE
```

```
100 MED=M
    IF (M.EQ.1) GO TO 50
    M=N/2
    MED=(X(M)+X(M+1))/2
    GO TO 100
50 MED=X(M)
    WRITE 60,MED
60 FORMAT(F5.3)
    END
```