COGNITIVE PSYCHOLOGY 12, 32–74 (1980)

Computer Text-Editing: An Information-Processing Analysis of a Routine Cognitive Skill

STUART K. CARD AND THOMAS P. MORAN
Xerox Palo Alto Research Center

AND

ALLEN NEWELL
Carnegie–Mellon University

An information-processing model is presented that describes how a person uses an interactive computer text-editing system to make modifications to a manuscript. It is demonstrated that the behavior of an expert user can be modeled by giving his goals, operators, methods, and selection rules for choosing among method alternatives. The paper assesses the predictions of such a model with respect to (1) predicting user behavior sequences, (2) predicting the time required to do particular modifications, and (3) determining the effect on accuracy of the detail with which the modeling is done (the model's "grain size"). Chronometric task protocols from several users are examined in some detail. Users' choices between alternative methods are predicted about 80% of the time by a few simple rules. Accuracy of the model is little affected by the detail of modeling. The manuscript-editing task is discussed as an example from the larger class of tasks called "routine cognitive skills."

The current attempt to understand man as a symbolic information processing system has concentrated on certain domains of behavior: recall and recognition tasks, which reveal the mechanisms of learning and the structure of short-term and long-term memory; discrete symbolic puzzles and mathematical exercises, which reveal the nature of search in problem solving; discrete symbolic induction tasks, which reveal elementary concept acquisition; tasks of elementary sentence comprehension, decision, and arithmetic, which reveal the nature of the immediate processor; and simple tasks that occur in child development. Most of this work is summarized by Anderson and Bower (1973), Lindsay and Norman (1977),

The authors are listed in alphabetical order. We would like to acknowledge George Baylor, University of Montreal, for helping us formulate the method selection issue and for running some pilot experiments. We would like to thank Donald Norman, Richard Young, Daniel Bobrow, and Jerome Elkind for comments on an earlier report (Card, Moran, & Newell, 1976) on which this paper is based. We would also like to thank Janet Farness for assistance for running the experiments and Richard Lyon for help with the matching algorithm.

Reprint requests should be sent to S. K. Card or T. P. Moran at Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA. 94304.

0010-0285/80/010032-43$05.00/0

Klatzky (1975), and Newell and Simon (1972). There remain, however, important domains of behavior for which we do not yet have any reasonable detailed theory nor any verification that the theory of man as a symbolic information processor provides an appropriate theoretical base.

We wish in this paper to consider an example task from one such domain of behavior which we shall call *routine cognitive skill.* Such behavior occurs in situations that are familiar and repetitive, and which people master with practice and training, but where the variability in the task, plus the induced variability arising from error, keeps the task from becoming completely routine and requires cognitive involvement. The example task is the *manuscript-editing task:* making corrections from a marked-up manuscript in a text file stored in a computer system through an on-line editor.

Our motives for studying the manuscript-editing task are both to extend the theory of man as a symbolic information processing system and to apply the theory of human information processing in practical task domains. The consideration of tasks with real application is important as a check on the often-noted tendency for psychological research to become paradigm-bound, as a way of testing the power of theoretical ideas against the complexity of real-world behavior, and as a means of fostering progress in the application domain.

This paper focuses on the basic structure of behavior in a close laboratory analogue of the natural manuscript-editing task environment: We consider the behavior of people under laboratory conditions (free from interruptions) employing a computer text-editing program commonly used in their daily work. They work from manuscripts with legible instructions to perform straightforward, single-line, but otherwise typical, modifications. We shall be concerned with several issues: Is it possible to describe the behavior of a person in a reasonably complex workaday task, like computer text-editing, as the repeated application of a small set of basic information processing operators? Is it possible to predict the actual sequence of operators a person will use and the time required to do any specific task? Finally, in attempting to describe behavior in this way, the issue of the level of analysis is critical. How does the model's ability to describe and predict a person's behavior change as we vary the grain size of the analysis?

THE MANUSCRIPT-EDITING TASK

It will be helpful for our later discussion to describe the manuscript-editing task in concrete detail. A person (the "user") sits before an on-line computer terminal, which has a keyboard for input and a CRT display for output. In the computer is a text file. To the user's left is a manuscript, a printout of the text-file, marked with modifications. The user, working

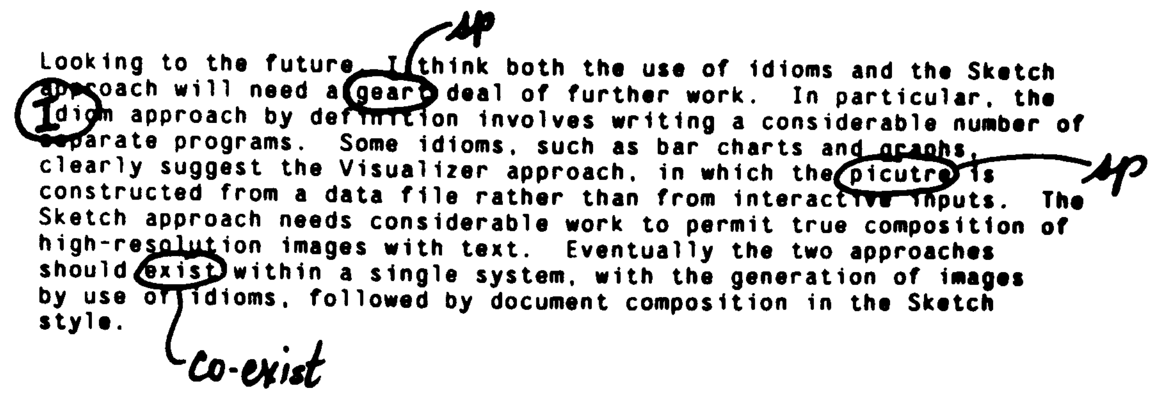
sp
Looking to the future, I think both the use of idioms and the Sketch
approach will need a geart deal of further work. In particular, the
1 idiom approach by definition involves writing a considerable number of
separate programs. Some idioms, such as bar charts and graphs,
clearly suggest the Visualizer approach, in which the picutre is sp
constructed from a data file rather than from interactive inputs. The
Sketch approach needs considerable work to permit true composition of
high-resolution images with text. Eventually the two approaches
should exist within a single system, with the generation of images
by use of idioms, followed by document composition in the Sketch
style.
co-exist

FIG. 1. Sample fragment of the manuscript used in experiment.

through a text-editor computer program, is to effect each of the modifications marked on the manuscript in the text file, thus producing an updated file. The task is daily routine for thousands of workers using equipment ranging from small "word-processors" to large general-purpose time-sharing computers. Variations of the task occur with variations in the nature of the computer, the editor, the terminal, the size of the manuscript, the type and number of corrections, the physical layout, and the familiarity of the user with the manuscript and with the editor.

To bring the details of the task into focus, let us consider the problem of making the corrections indicated in Fig. 1 with the system we will use for our analysis—a system called POET (Russell, 1973). The first instruction on the manuscript indicates the word "great" has been mistyped in the manuscript as "geart." A typical dialogue between the system and the user is as follows.[1]

SYSTEM:	#	*(Prompts user for a command)*
USER:	"geart"	*(The user indicates the line he wants by typing between quotation marks some characters that identify the line)*

The system responds internally by finding the next line that contains the sequence of characters "geart" and making that its "Current Line."

SYSTEM:	#	
USER:	/	*(This command requests the system to display its Current Line. The user issues the command to make sure the system is attending to the line he intended.)*

The system displays the Current Line on the bottom line of the display screen.

[1] Ellipsis ". . ." is used to shorten long typeouts by the system; carriage returns typed by the user are indicated as ⟨CR⟩, linefeed characters by ⟨LF⟩; the dialogue proceeds by adding new lines to the bottom of the CRT, "scrolling" the rest of the lines up and eventually off the top.

SYSTEM: **approach will need a geart deal of further work . . .**

The user decides to make the change by using the system's Substitute command to substitute the characters "grea" for the characters "gear."

SYSTEM:	**#**	
USER:	**s**	*(Tells system to use the substitute command)*
SYSTEM:	**ubstitute**	*(System completes typing of command name)*
USER:	**grea⟨CR⟩**	*(Types new text followed by a carriage return)*
SYSTEM:	**(for)**	*(Prompts user for second argument)*
USER:	**gear⟨CR⟩**	*(Types text to be replaced followed by a carriage return)*
SYSTEM:	**[OK]**	*(Asks user to confirm that the command is correct before system will execute it)*
USER:	**⟨CR⟩**	*(Types carriage return to confirm)*

The system makes the substitution as many times as it can and responds by printing the number of substitutions it made—in this case only one is made.

SYSTEM: **1**
SYSTEM: **#**

Again, the user wishes to look at the Current Line to see if, in fact, the modification was made as he intended.

USER: **/**
SYSTEM: **approach will need a great deal of further work . . .**
SYSTEM: **#**

The modification was correct. The user now proceeds to the next task.

In the next task, the word *idiom* is to be capitalized. The user decides he will do this by substituting *Idi* for *idi*. This time he uses the "linefeed method" rather than the "quoted string method" to locate the line to be changed.

USER:	**⟨LF⟩**	*(This is the command to print the next line and to make it be the Current Line)*
SYSTEM:	**idiom approach by definition involves writing a . . .**	
SYSTEM:	**#**	
USER:	**s**	
SYSTEM:	**ubstitute**	
USER:	**Idi⟨CR⟩**	
SYSTEM:	**(for)**	
USER:	**idi⟨CR⟩**	
SYSTEM:	**[OK]**	
USER:	**⟨CR⟩**	
SYSTEM:	**1**	
SYSTEM:	**#**	
USER:	**/**	
SYSTEM:	**Idiom approach by definition involves writing a . . .**	
SYSTEM:	**#**	

In this manner, the user proceeds, modification by modification, through the manuscript.

THE GOMS MODEL

The account we shall give of behavior in the manuscript-editing task is an application of the general theory of humans as symbolic processors to this particular task environment (for more general treatment, see Card, Moran, & Newell, in preparation). The models we shall describe hypothesize that the user's cognitive structure consists of four components: a set of Goals, a set of Operators, a set of Methods for achieving the goals, and a set of Selection rules for choosing among a goal's competing methods. For short, we shall call a model specified by such components a GOMS model.

Example Model

As an example of the basic concepts of a GOMS model and the notation used, let us consider a particular model, called M4B, of the manuscript-editing task. According to the model, the user begins with the top-level goal:

GOAL: EDIT-MANUSCRIPT

It is a characteristic of manuscript-editing that the larger task of editing the manuscript is composed of a collection of small *unit tasks* which are almost completely independent of each other. The obvious method for accomplishing the top level goal is to do each of the individual unit tasks one at a time:

GOAL: EDIT-MANUSCRIPT
- GOAL: EDIT-UNIT-TASK *repeat until no more unit tasks.*

The indentation above indicates that EDIT-UNIT-TASK is a subgoal of EDIT-MANUSCRIPT, and the annotation in italics says that the subgoal is to be invoked repeatedly until no more unit tasks remain.

To edit a unit task, the user must first get the unit task from the manuscript and then do what is necessary to accomplish the unit task:

GOAL: EDIT-UNIT-TASK
- GOAL: GET-UNIT-TASK
- GOAL: DO-UNIT-TASK

Each subgoal will itself evoke appropriate methods. There is a simple method for getting a task:

GOAL: GET-UNIT-TASK
- GET-NEXT-PAGE *if at end of manuscript page*
- GET-UNIT-TASK

The operator GET-NEXT-PAGE is invoked only if there are no more edit instructions on the current page of the manuscript. The bulk of the work toward the goal—looking at the manuscript, finding an editing instruction, and interpreting the instruction as an edit task—is done by the operator GET-UNIT-TASK.

To do a unit task in POET there is a two-step method:

GOAL: DO-UNIT-TASK
- GOAL: LOCATE-LINE
- GOAL: MODIFY-TEXT

The Current Line of the POET editor must first be set to the line where the correction is to be made. Then the appropriate text on that line can be modified.

To locate POET at a line, there is a choice of two methods:

GOAL: LOCATE-LINE
- [select: USE-LF-METHOD
  USE-QS-METHOD]

To use the LF-METHOD, the linefeed key is pressed repeatedly, causing the editor to advance one line each time. To use the QS-METHOD, a string in quotation marks is typed which identifies the line. Usually the LF-METHOD is selected when the new unit task is within a few lines of the current unit task, and the QS-METHOD is selected when the new unit task is farther away.

Once the line has been located, there is a choice of how to modify the text:

GOAL: MODIFY-TEXT
- [select: USE-S-COMMAND
  USE-M-COMMAND]
- VERIFY-EDIT

That is, either POET's Substitute command or Modify command can be used to alter text on a line. An example has already been given of the Substitute command. The Modify command allows the user to invoke a series of subcommands for moving forward and backward and for making modifications within a line. In either case a VERIFY-EDIT operation is evoked to check what actually happened against the user's intentions.

Putting all the pieces together into one tree structure, we have:

GOAL: EDIT-MANUSCRIPT
- GOAL: EDIT-UNIT-TASK *repeat until no more unit tasks*
  - GOAL: GET-UNIT-TASK
    - GET-NEXT-PAGE *if at end of manuscript page*
    - GET-UNIT-TASK
  - GOAL: DO-UNIT-TASK

```
if (i≠0 and (j=0 or (Score[i-1,j] > Score[i-1,j-1])) then
  PredSeqResult[k] ← PredSeq[i];
  ObsSeqResult[k] ← "X";
  k ← k+1; i ← i-1;
else if (j≠0 and (i=0 or (Score[i,j-1] > Score[i-1,j-1])) then
  PredSeqResult[k] ← "X";
  ObsSeqResult[k] ← ObsSeq[j];
  k ← k+1; j ← j-1;
else
  PredSeqResult[k] ← PredSeq[i];
  ObsSeqResult[k] ← ObsSeq[j];
  k ← k+1; i ← i-1; j ← j-1;
%Match ← Score[PredLength, ObsLength] / (k-1);
return(%Match, PredSeqResult, ObsSeqResult); end;
```

REFERENCES

Abruzzi, A. *Work measurement.* New York: Columbia Univ. Press, 1952.

Abruzzi, A. *Work, workers, and work measurement.* New York: Columbia Univ. Press, 1956.

Anderson, J. R., & Bower, G. H. *Human associative memory.* Washington, D.C.: Winston, 1973.

Bartlett, F. *Thinking.* New York: Basic Books, 1958.

Card, S. K. *Studies in the psychology of computer text-editing systems.* Ph.D. Dissertation, Department of Psychology, Carnegie-Mellon Univ., 1978.

Card, S. K., Moran, T. P., & Newell, A. *The manuscript editing task: A routine cognitive skill* (Technical Report SSL-76-8). Palo Alto, CA: Xerox Palo Alto Research Center, 1976.

Card, S. K., Moran, T. P., & Newell, A. Hillsdale, NJ: Erlbaum, book in preparation.

Hirschberg, D. S. A linear space algorithm for computing maximal common subsequences. *Communications of the ACM,* 1975, **18**, 341–343.

Kinkead, R. Typing speed, keying rates, and optimal keyboard layouts. *Proceedings of Annual Meeting of the Human Factors Society,* 1975, 159–161.

Klatzky, R. L. *Human memory: structures and processes.* San Francisco: Freeman, 1975.

Lindsay, P. H., & Norman, D. A. *Human information processing: An introduction to psychology.* New York: Academic Press, 1977. 2nd ed.

Long, J. Visual feedback and skilled keying: Differential effects of masking the printed copy and the keyboard. *Ergonomics,* 1976, **19**, 93–110.

Maynard, H. B. *Industrial engineering handbook.* New York: McGraw–Hill, 1971. 3rd ed.

Mills, R. G., & Hatfield, S. A. Sequential task performance: Task module relationships, reliabilities, and times. *Human Factors,* 1974, **16**, 117–128.

Newell, A. Production systems: Models of control structures. In W. G. Chase (Ed.), *Visual information processing.* New York: Academic Press, 1973.

Newell, A., & Simon, H. A. *Human problem solving.* Englewood Cliffs, NJ: Prentice–Hall, 1972.

Russell, D. S. POET: A page oriented editor for TENEX. University of Utah, Computer Science Division, 1973.

Sakoe, H., & Chiba, S. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing,* 1978, ASSP-**26**(1), 43–49.

Shaffer, L. H. Intention and performance. *Psychological Review,* 1976, **83**, 375–393.

Welford, A. T. *Fundamentals of Skill.* London: Methuen, 1968.

(Accepted May 14, 1979)