Natural Command Names and Initial Learning: A Study of Text-Editing Terms

T. K. LANDAUER, K. M. GALOTTI, S. HARTWELL *Bell Laboratories*

Thomas K. Landauer leads a group of cognitive psychologists interested in a variety of topics relevant to human–computer communications, such as command name choice, categorization and menu access to information and database query language issues. T. K. Landauer is a fellow of AAAS and a consulting editor of the Journal of Experimental Psychology.
Kathleen M. Galotti is a graduate student in Psychology and Computer Science at the University of Pennsylvania.
Steve Hartwell is a systems adminstrator in the Electrical Engineering Department at Stanford.

Authors' Present Addresses: T.K. Landauer, Bell Laboratories, 600 Mountain Ave., Murray Hill, NJ 07974; K.M. Galotti, Department of Psychology, University of Pennsylvania, 3815 Walnut St. T-3, Philadelphia, PA 19104; S. Hartwell, Department of Electrical Engineering, Stanford University, Stanford, CA 94305.

***ABSTRACT:** In the first of two studies of "naturalness" in command names, computer-naive typists composed instructions to "someone else" for correcting a sample text. There was great variety in their task-descriptive lexicon and a lack of correspondence between both their vocabulary and their underlying conceptions of the editing operations and those of some computerized text editors. In the second study, computer-naive typists spent two hours learning minimal text-editing systems that varied in several ways. Lexical naturalness (frequency of use in Study 1) made little difference in their performance. By contrast, having different, rather than the same names for operations requiring different syntax greatly reduced difficulty. It is concluded that the design of user-compatible commands involves deeper issues than are captured by the slogan "naturalness." However, there are limitations to our observations. Only initial learning of a small set of commands was at issue and generalizations to other situations will require further testing.*

Novice users of computer systems often complain bitterly about the strangeness and difficulty of learning their new skill. Reports of displeasure and high dropout rates are common from programs such as those in which experienced typists learn to use text editors. A frequent expression of the problem, from the beginner's view, is "why isn't it in English?" This suggests that at least some of the start-up difficulty may lie in the selection of command names. Designers may choose names from a specialized vocabulary that reflects expert knowledge of the system. But this vocabulary will not necessarily correspond well to the way new users think of the task or the language they would find best to employ or understand in its description.

We wanted to find out whether unfamiliar terminology per se was an important obstacle to beginning use of computer systems. Would more natural terms—putting things in "the users' own words"—help smooth the transition from manual to computer-aided methods? The first step was to see whether available methodology could help us to understand how people naturally think about some computerized task and to identify natural terminology for that task.

We chose text editing as a vehicle for study for several reasons. First, it is often the first computer-related skill that nonspecialists acquire. Second, text editors (including the line-oriented and hard-copy based ones) are currently widely used by nonprogrammers and are gaining users (primarily secretaries and typists) rapidly. This means that findings about editing commands as such may have direct applications. Also, and more important for our purposes, it means that there is a large pool of appropriately experienced and motivated subjects to study.

Previous work on natural command names has been scant. Black and Sebrechts [2] had college students provide a one-sentence description of changes indicated on a manuscript by author/proofreader marks. They concluded that people have preconceptions about the names of operations (e.g., add, cross-out, change, etc.) and suggested that using these names would

improve a text editor. Streeter et al. [10] conducted a similar study using homemakers as subjects. Subjects saw two versions of the same text; one contained errors, the other was a corrected version and provided a one-word description of each change.

Comparing the results of these two studies suggests that different ways of eliciting potential command names yield different responses, for example, where "cross out" was popular in Black's study, it was rarely used in Streeter et al. It is not obvious, a priori, what eliciting conditions would provide the most natural or effective command names. However, an hypothesis we wished to test was this: that the words an actual user would employ to describe the actions to be taken to perform the editing task in its noncomputerized form would make initial learning easier. The rationale is that naive users can be expected to think in terms of asking the computer to do what they or someone like them would otherwise do. Thus the words one typist would use to instruct another typist to produce a particular result, under the conditions of an actual editing task, should provide the kind of "natural" command names in which we were interested.

In the studies described here, we explore two particular questions. First, how do typists normally think about and describe editing operations? Second, would incorporating the novice's words into the language needed to operate an editing system make initial learning easier or the system more acceptable?

We preface description of these studies with some disclaimers. Only command names and their assignment were studied, not variations in syntax or construction. Only variations in naturalness of real words was at issue, with naturalness defined as spontaneous applicability by computer-naive users. Behavioral tests involved only initial learning of a small set of editing commands. In our current state of ignorance about what variables are important in command names, it would be hazardous to generalize any conclusions much beyond actual conditions of the reported studies.

STUDY 1: IDENTIFYING NATURAL COMMAND NAMES

This study was designed to address several issues: How do people conceive of the actions needed to make changes in a manuscript? What names do they give to these operations? To what units or objects of text (e.g., blanks, characters, lines, paragraphs) do spontaneously named operations naturally apply? Do people use the same words to describe the same type of change when the unit of text differs?

Methods

Subjects received a manuscript marked with "author/proofreader" marks. These marks, and the types of changes to

TABLE I. Frequency of User-Provided Names

Type of Change	Object									
	Blank		Character		Word		Line		Paragraph	
Insert	Substitute*	0	Substitute*	0	Substitute*	0	Append*	0	Append*	0
(put in text)	Space	15	Change	20	Insert	25	Insert*	19	Insert*	8
	Put	9	Should be	17	Add	20	Add	24	Type	15
			Insert	11	Place	13	Type	17	Add	15
			Add	8	Put	13				
					Type	11			Put	11
	Other	72	Other	40	Other	14	Other	36	Other	47
	(22 types)		(17 types)		(9 types)		(13 types)		(11 types)	
Delete	Substitute*	0	Substitute*	0	Substitute*	0	Delete*	6	Delete*	6
(remove text)	Connect	13	Omit	15	Omit	31	Omit	34	Omit	32
			Spell	12	Take out	10				
	Change	11	Change	11	Delete	9	Take out	10		
							Eliminate	9		
	Other	72	Other	58	Other	46	Other	37	Other	58
	(31 types)		(20 types)		(15 types)		(9 types)		(14 types)	
Replace	Substitute*	0	Substitute*	1	Substitute*	4	Change*	11	Change*	0
(new text in place of old)	Add	13	Change	18	Change	22			Replace	6
	Insert	10			Replace	10				
	Put in	10	Replace	9						
	Change	9								
	Spell	9								
	Other	45	Other	68	Other	60	Other	85	Other	90
	(20 types)		(35 types)		(35 types)		(43 types)		(44 types)	
Move	Substitute*	0	Substitute*	0	Substitute*	0	Move*	2	Move*	8
(change location of text)	Should be	10	Change	16	Place	11	Put	12	Put	13
					Type	10				
			Put	13	Put	9	Type	9	Place	9
			Should be	17						
	Other	86	Other	50	Other	66	Other	73	Other	66
	(43 types)		(25 types)		(36 types)		(40 types)		(36 types)	
Transpose	Substitute*	0	Substitute*	0	Substitute*	0	Move*	0	Move*	3
(interchange locations)	Change	8	Change	17	Reverse	12				
			Should be	17						
			Switch	14	Switch	11			Switch	16
			Spell	10	Type	9	Type	5		
	Other	88	Other	38	Other	64	Other	91	Other	77
	(46 types)		(21 types)		(27 types)		(52 types)		(22 types)	

* Present ED command.

TABLE II. Intrasubject[a]/Intersubject[b] Naming Agreement

Operation	Object					
	Blank	Character	Word	Line	Paragraph	Mean
Insert	0.19/0.10	0.38/0.13	0.52/0.15	0.35/0.13	0.31/0.07	0.350/0.114
Delete	0.29/0.07	0.45/0.08	0.54/0.13	0.50/0.15	0.46/0.11	0.448/0.108
Replace	0.32/0.07	0.42/0.09	0.44/0.09	0.38/0.04	0.40/0.05	0.390/0.068
Move	0.17/0.04	0.50/0.10	0.27/0.07	0.19/0.07	0.27/0.08	0.280/0.072
Transpose	0.23/0.05	0.52/0.09	0.27/0.06	[c]	0.27/0.07	0.323/0.058
Mean	0.238/0.066	0.454/0.098	0.408/0.100	0.355/0.082	0.342/0.076	0.359/0.084

[a] Numbers represent the proportions of individual subjects who used the same main verb in describing the necessary editing action for two occurrences of presumably equivalent author-indicated corrections in a sample text.
[b] Numbers represent the proportion of all possible pairs of subjects who used the same main verb for the operations of the indicated varieties
[c] Because of experimental error these values could not be calculated.

which they referred, incorporated those that we found in a sample of manuscripts brought to Bell Laboratories typists. From these observations, and the operations available on some common computer editors, we constructed a taxonomy of significant operations and objects involved in text editing. These form the rows and columns of Table I. A text was constructed in which changes corresponding to each of the cells in Table I were indicated by author marks in pencil. Each requested change to be made was margin-numbered for reference. Participants were asked to prepare a typed list of brief instructions for someone else who was actually going to make the changes. We had subjects type their responses to further encourage brevity.

There were 50 indicated corrections, comprising 5 types of operations, applied to each of 5 different units of text, with 2 instances of each combination. Half of the texts were line-numbered, the others were not; they were randomly assigned to subjects. Twenty-two secretarial school students and 26 high school students with some typing experience participated.

Results

Responses to each indicated change were sorted into categories, based on the main verb used to form an instruction to perform an editing operation. Frequencies are given in Table I, where each cell represents one category in our taxonomy and contains the data for 48 subjects judging 2 instances of each operation. For comparison, command names for the UNIX™ editor ED are listed first and are indicated by superscript. The total number of responses for each cell is 96. Responses with low frequencies, as well as instances of no response, have been grouped together as *other*.

One striking result is that in 24 out of 25 cells, the present ED command name was not the most frequent spontaneously given name. In fact, in 23 out of 25 cells there are at least 2 other more frequent responses. Thus novices' claim that the system does not use their words is correct.

Another striking aspect of the data is that there was little agreement on names. On the average, the three most popular verbs for each operation together account for only 33 percent of the total number of responses. The mean intersubject agreement, that is, the probability that any two users would use the same verb in response to a particular text correction, as estimated from this data, is only 0.08. Indeed, individual subjects were not even very self-consistent. Overall, the likelihood that a given subject would use the same verb for both instances of what we assumed to be equivalent corrections, was only 0.34. Table II shows the degree of intrasubject and intersubject agreement for each correction type.

Bearing in mind the general lack of agreement between people or consistency between instances for the same person, the data suggest that for typists the following operation names are most natural: "add" for the **insert** operation, "omit" for the **delete** operation, and "change" for the **replace** operation. For the **move** and **transpose** operation, "put" and "switch" are most natural, but the absolute frequency of these responses was low.

If it is assumed that people are more likely to apply consistent names to operation-object entities that correspond to some sort of stable mental representation, the data in Table II ought to reflect something like the "cognitive naturalness" of the experimenter-defined entities. By this index, operations involving the manipulation of blanks, and perhaps those involving moving and transposing, appear least natural, and operations involving characters or deletion appear most natural. It seems plausible that these differences reflect experience with the manual typing environment in which keys can be used essentially only to enter single characters that may or may not have been previously erased.

Another way to gain insight into natural conceptions of editing tasks is to examine the similarities between descriptions of the various nominally different operation-object entities. For this purpose we tabulated an index of similarity for each of the $[(n \times n - 1)/2 = 300]$ possible pairs of predefined correction types. For any two correction types, we determined whether the main verb used in the description of either of the two instances of one was also used in either of the two instances of the other, that is, whether a given subject used, at least once, the same verb for the two different operation-object entities. The resulting binary score—1 for any agreement, 0 for none—was summed over the 48 subjects. The matrix of proximity indices was then subjected to several latent structure analysis procedures. Statistically satisfactory and at least somewhat revealing structures were obtained with both hierarchical clustering and multidimensional scaling methods, exemplary results of which are shown in Figures 1 and 2, respectively.[1] The hierarchical structure shown in Figure 1 describes the data fairly well, accounting for 65 percent of the variance in the observed proximities. Among its interesting features are the following: inserting blanks and transposing lines are both seen as unlike other classes of editing acts; moving, deleting, and transposing blanks form a strong low-level cluster, while replacing a blank with a character is conceived as equivalent to inserting a character. Deletions of words, lines, and paragraphs are described almost identically, while deletion of blanks falls with other opera-

[1] For those unfamiliar with these procedures, Kruskal and Wish [6] provide a good overview. A general form of structure, e.g., a binary ultrametric tree or a cross-cutting dimensional space is assumed. Additional, fairly weak assumptions about underlying scaling properties and constraints are also needed. Given these, a particular "best-fit" structure is determined by iterative numerical methods.

tions on blanks, and deletion of characters is grouped only in a relatively higher order category of acts having to do mostly with operations on characters.

Five fairly clearly interpretable, but less compact, higher level groupings can be seen: (1) transposition (of characters, words, and paragraphs), (2) all operations on blanks and characters, including replacement of a word (which, therefore, is evidently thought of as a character-by-character operation), (3) moving (of words, lines, and paragraphs), (4) insertions, and (5) deletion or replacement (of words, lines, and paragraphs).

The multidimensional scaling analysis also describes the data fairly well (stress = 0.099) as a three-dimensional space. Figure 2 shows the location of the 25 operation-object pairs in this space, as projected onto the surface defined by the first two dimensions (by rotation to principle components). The first dimension clearly distinguishes insert operations from delete operations. Replace operations are also located in a fairly distinct region of the space (except for replacing blanks). However, the insert, move, and transpose operations overlap considerably. The second dimension distinguishes blanks from characters and these from words, lines, and paragraphs. The second dimension thus suggests that typists think of essentially three levels of text (or actions): those having to do with blank space (space bar or erasure), characters (key strokes), and larger text units. Recall that the words whose similarities of use are represented here are verbs (or predicate phrases) only, not object nouns. It is therefore rather interesting that the similarities in their use are so heavily influenced by the object units being described.

Subjects were least consistent in the naming of operations applied to blanks (e.g., deleting a blank, replacing a blank with a character, etc.), and blanks form separate clusters in both the hierarchical and multidimensional scaling based on similarities among descriptions. It thus seems clear that typists do not think of a blank as just another character, the way computer experts do. It seems likely that our subjects' thinking on this matter is somehow tied to the technology of a typewriter. A typist using a typewriter probably regards a

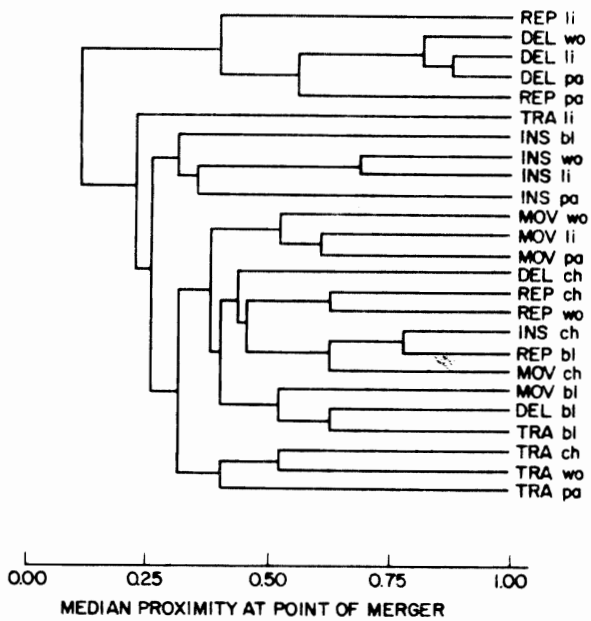

FIGURE 1. Hierarchical tree representing similarities in use of verbs to describe editing operations (REPlace, DELete, TRAnspose, INSert, MOVe).

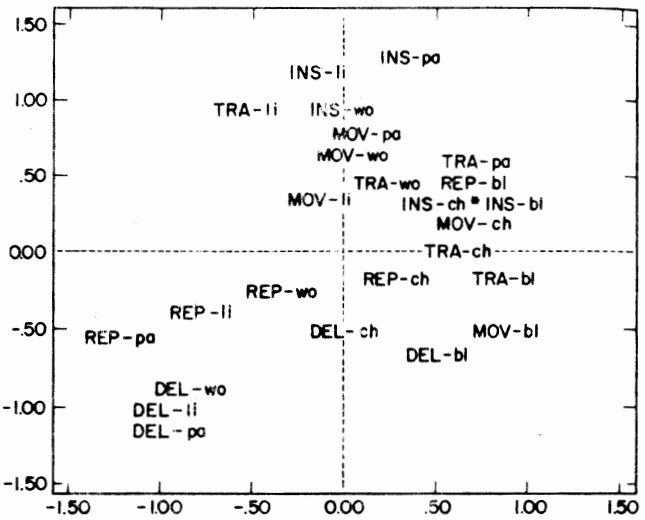

FIGURE 2. Multidimensional scaling representation of similarities in use of verbs to describe editing operations.

blank as the absence of letters, or as preexisting white space. An inexperienced user of a text editor may have to be explicitly instructed that the computer treats blanks as special characters, not as a lack of characters.

A further analysis was done to see under what circumstances subjects considered a line as a unit to be operated upon. Instructions for changes that could be accomplished by operations on whole lines (e.g., replace line, insert line) were considered. In the absence of line numbers, 32 percent of 201 classifiable editing instructions specifically referred to lines. With line numbers present, 68 percent of 204 classifiable instructions referred to lines. This result suggests that typists do not ordinarily think of editing as involving operations that effect whole lines, but that the provision of line numbers tends to induce this mode of thinking.

Table I also shows that the subjects' most popular choices of operation verbs were not much different for operations on line-internal versus whole-line categories. That is, subjects tended to use the same verbs (e.g., *omit*) when instructing someone to delete a character, word, line, or paragraph. In both the hierarchical clustering and multidimensional scaling solutions, words, lines, and paragraphs tend to group together, while characters and blanks form a separate cluster. In the structure of some present text editors, UNIX ED, for example, different commands are used for line-internal operations (e.g., *subsitute*) and whole-line changes (e.g., *delete*). This distinction fails to match the natural language groupings exhibited by our subjects. However, in UNIX ED, for example, line-internal and whole-line operations require different syntactic constructions in the argument portion of the command. It might be advantageous to have different names for commands that use different syntax, independent of natural naming preferences. This issue, among others, is addressed in Study 2.

A possible disadvantage of popular verbs as command names is that they may tend to be too general and thus fail to capture distinctions between different operations. Consider the verb "fix." Our next analysis, then, addresses the issue of how precisely our subjects used verbs to request text corrections.

For each verb used at least twice we tallied how often it was used for each of the five operations. We then determined the most frequent (modal) referent of each verb. Overall,

verbs were used to refer to their modal operation in about 50 percent of all cases. This tells us that even the most knowledgeable recipient (editor) could correctly understand the intended operation of a novice only about half the time, if it always made a single "best guess" when confronted with a naive user's untutored verb.

This may be an important observation for those who would like systems to accept "raw natural language." In this instance, at least, natural terminology is inherently imprecise, that is, it is not easy to guess what users want from what they say. (See Furnas et al. [3] for more on this matter).

Of course, regularly used systems like text editors do not usually rely on natural language input, but instead teach users a specialized vocabulary for the task. However, this does not necessarily remove the problem of imprecision. Even though we provide precise definitions of command words, users will not immediately comply in employing them. Our problem then becomes choosing words that people can most easily learn to use precisely. We know that preexisting association between stimulus and response makes learning easier [5]. Thus the most popular terms given in response to the various indicated text corrections are predictably the easiest to learn as command names to be recalled under similar circumstances (see [2]).

On the other hand, there are at least two reasons to suspect that popular choices might not be optimal as command names. First, as we have just seen, popular terms tend to be referentially imprecise. People will often use them when they are inappropriate and may have difficulty learning to restrict their usage to only correct occasions. Second, as we mentioned earlier, command names need to refer to computer operations and syntax as well as to text-defined editing conditions. These two roles may not be best performed by the same words.

STUDY 2: AN EVALUATION OF NATURAL COMMANDS IN INITIAL LEARNING OF A MINIMAL SYSTEM

Having found that the editing vocabularies of typists and existing text editors differ, we turned to the question of whether incorporating verbs more commonly used by typists would facilitate initial learning and decrease initial negative reaction to a computerized text editor. A recent study showed that one particular text-editing system that employed relatively more familiar, everyday language terms and syntax resulted in more effective use for both experienced and relatively inexperienced users than another system that used more arbitrary terms, punctuation, and argument construction. Ledgard et al. [7] had college students with varying levels of prior computer experience learn to use two text editors. One, which they call the "notational editor," was based on the Control Data Corporation NOS Version I Text Editor. It uses a somewhat arbitrary and complicated command set and syntax. The other, called the "English editor," contained the same operations, but used a simpler syntax and vocabulary chosen by the experimenters to resemble "legitimate English phrases, formed of familiar, descriptive words." Subjects received a manual and were allowed to practice with each editor prior to a 20-minute testing session. Results showed that subjects completed more of the task, made fewer errors, and were more efficient when using the so-called "English editor."

However, the subjects in this study were not representative of a secretarial population, and the two particular editors compared differed more dramatically in syntax and punctuation than in command names. Moreover, the command names were selected according to the designers' personal intuitions rather than by any objective technique such as the "user nomination" method in which we were interested. Therefore, the commands did not reflect users' natural language for the task in a systematic or necessary way. Nevertheless, the results of Ledgard et al. are consistent with the hypothesis that more natural command words might make systems easier to use.

Our second study used findings from the first to investigate whether incorporating natural user terminology from the manual typing environment into a computerized text editor would improve its initial ease of learning. Initial learning is of special interest for several reasons. First, learning curves are usually steepest at the outset, so early learning is expectably a more sensitive indicator of effective variables than is later learning. (We know of few instances in the learning psychology literature in which variables that speed early learning have had detrimental effects on later learning.) Second, it is often claimed that initial introduction to computer systems is the largest obstacle to acceptance; early "novice freak-out" sometimes either preempts further progress entirely or creates persisting attitudinal difficulties. Finally, testing during early learning is, obviously, easier as a practical matter than testing after much experience.

Subjects in this study learned to use a small subset of text-editing operations available on the UNIX™ editor, ED. The three principal operations they learned were "primitive" operations needed by regular users. These were also commands for which spontaneous terminology differed substantially from that in the ED system—namely, the **insert**, **delete**, and **replace** operations. Notice also that these three classes of operations form well separable clusters in the latent structure analyses of Study 1. Consequently, it can be argued that they form an especially natural set of minimal operations to learn first. Subjects also learned three accessory commands: starting an exercise (*start n*); aborting the entry of a line (@); and terminating input for an insert (.), for a total of six commands. Three parameters of commands were studied: *vocabulary*, *scope*, and *length*. Variations were applied only to the three text-altering commands. The *vocabulary* factor had three levels: *old*, the existing ED command set ("append," "delete," "substitute"); *new*, the modal responses found in Study 1 ("add," "omit," "change"); and *random*, words unrelated to the operation but matched to the existing command names in frequency in the English language, length, and word type ("cipher," "allege," "deliberate"). The random condition serves as a baseline against which to test the importance of the choice of names. The factor *scope* had two levels: *old*, in which a line-internal change required a different command name from a whole-line change; and *new*, in which the command was lexically the same for line-internal and whole-line changes. Finally, the *length* of commands had two levels: *short*, using only the first letter of the command name; and *long*, the full name typed out. One might hypothesize that natural meanings would be less influential if the words are severely abbreviated than if they are spelled out in full at each entry. The variations in commands and their associated syntax are shown in Table III. The time required to complete an exercise and the number of text defects remaining at the completion of each exercise served as outcome measurement variables.

Method

Subjects. Sixty-five students from nearby secretarial schools and 56 high school students with some typing skills were paid to participate in an experiment which, they were told, was to evaluate the usefulness of computer text editing. Seventeen

TABLE III. Commands and Associated Syntax Used in Study 2

A. Commands for old and new scope conditions, old vocabulary, long form

Learned in practice exercise	Required operation	Old scope	New scope
2.	Removing a line	Delete	Delete
4.	Removing a word	Substitute/word//	Delete/word/
1.	Adding a line	Append Text .	Append Text .
3.	Adding a word	Substitute/word/new word/	Append/new/word/
5.	Changing a line	Delete Append Text .	Substitute Text .
6.	Changing a word	Substitute/word/new/	Substitute/word/new/

B. Vocabulary difference. Short forms used first letter only.

Vocabulary		
Old	New	Random
Delete	Omit	Allege
Append	Add	Cipher
Substitute	Change	Deliberate

secretarial students and 8 high school students did not finish the task; their data were replaced by that of new subjects for all analyses. However, there was no systematic tendency for these subjects to come from any particular level of any of the experimental factors (e.g., new scope, or old).

Materials and Procedure. Subjects typed at a Texas Instruments 700 computer terminal. This device is similar to a standard typewriter in keyboard layout. It uses only printed paper display, but an automatic outcome-printing feature, described below, gave it some of the presumably favorable characteristics usually associated with full-screen editors.

The text editor used was a "stripped-down" version of the UNIX™ editor ED (1). This was implemented by a preprocessor in front of ED. The preprocessor translated experimental versions of commands into ED commands. It also logged the time and command. The @ command deleted the current line. A new command, *start*, was created to allow the subjects to specify which exercises they were currently working on. They began each exercise by giving the command *start n*, where *n* specifies the exercise number. The preprocessor then issued an internal command to ED to read in a file containing the sample text for that exercise. After every command given, all of the text in the buffer was printed at the terminal. This provided immediate feedback of the results of each operation. (Pilot testing convinced us that such feedback was critical; without it many subjects were unable to make sufficient progress to provide usable data.) Subjects were supplied with manuals covering editing fundamentals to be learned. They were instructed to read and do the exercises in the manual, typing at the terminal. At the end of the session, they were given a short opinion questionnaire. There were three editing operations: adding text, removing text, and changing text; and two objects: lines and words. The first six sections in the manual, after the introduction, provided instruction for the resulting six command types. These were (in the order introduced) putting in lines of text, taking out lines of text, putting in words, taking out words, changing lines, and changing words.

Immediately following each of the first six sections in the manual was an exercise to be performed by the subject at the computer terminal that could be accomplished by using the new command. The remaining four sections, 7–10, introduced no new commands, but instead each called for the use of all six command types. These last exercises are referred to as test exercises.

The text of the manuals was constructed so that differences in command names, scope, and length could be described with minimal substitutions of words and phrases. The variations in levels of the experimental factors produced 12 different manuals. The exercise pages were completely identical and did not specify what command the subject should use.

The opinion questionnaire given at the end of the experiment consisted of 6 questions that asked subjects to express, in the form of 5-point ratings, their attitudes toward the task. Due to extraneous circumstances, complete questionnaire data are available only for the high school student subjects.

Procedure. Subjects participated in groups of three in adjacent isolation booths. They were given a brief introduction to computer text editing, then told to learn how to use the text editor, on their own, by reading each section in the manual carefully and doing the corresponding exercise at the keyboard. They were to complete each exercise, that is, make the sample text correct, before going on. Questions were answered, but as minimally and nondirectively as possible; there were usually not more than five questions per session.

Results

The last four exercises (7–10) were intended to test the operations taught in Exercises 1–6. Although the required changes could be done in a number of ways, we tried, through subtle manual wording, to make it apparent that each of the six operations should be used in each test exercise.

Means of the total times to complete the four test exercises, as a function of the experimental variables, are shown in

Table IV. We believe that total time to correctly accomplish editing tasks is the best dependent measure to assess comparative usability. The ability to correct text completely and rapidly is the desired result of using a text editor. Moreover, most important but more detailed effects, such as errors, typing or problem solving difficulties, and even boredom or frustration, ought to be reflected in increased time.

A four-factor analysis of variance [11], with groups, scope, vocabulary, and length as between-subjects variables was performed. There were no appreciable differences in the time it took secretarial students versus high school students to complete the text exercises. As shown in Table IV, there were mean differences between vocabulary conditions, but they clearly do not favor the new (more "natural") over the old (ED) command names. Indeed, even a post hoc test of the difference between just the best (old, $N = 32$) and worst (random, $N = 32$) vocabulary conditions was not nearly statistically significant (t (62) = 1.0). However, subjects in the old scope condition took significantly less time than subjects in the new scope condition $F(1,92) = 12.05$, $p < 0.001$). All our editors allowed operations on either character strings or lines as units with a different syntax to specify the operands in the two cases. Under these conditions, different command names for line-internal versus whole-line operation were distinctly advantageous even though, as found in Study 1, typists do not spontaneously describe line and word changes as different operations. This effect was independent of vocabulary choice; having "allege" and "deliberate" effect whole-line and within-line deletion, respectively, was superior to using "allege" for both, to the same extent that using "delete" and "substitute," respectively, was superior to using "delete" for both cases.

Somewhat surprisingly, the time differences between subjects in the short versus long conditions were not statistically significant (see Table IV). Apparently the number of keystrokes is not a dominant factor in editing time during initial learning (but see further analysis below). However, there was a significant interaction between effects of length and scope ($F(1,72) = 4.22$, $p < 0.05$), reflecting a greater effect of scope for long than short command names. Possibly, the meaning differences involved in the scope effect are attenuated when the commands are reduced to single letters. None of the other interactions was statistically significant; for example, although vocabulary effects were actually slightly *smaller* for spelled-out commands, the differential was well within the expected range of chance variation.

Because there may have been a tradeoff between a particular subject's speed in completing an exercise and the number of errors that were made while doing it, we did two other analyses. First, we repeated the data analyses with total number of characters typed as the measurement variable. Total characters must reflect the number of incorrect command entries, and additional commands needed to correct such errors as well. This analysis yielded no large or significant differences except for the effect of length. Predictably, subjects in the *short* condition typed fewer characters than those in the *long* condition ($\bar{X} = 566$ and 910, respectively, for exercises 7–10 combined; standard error of means = 34.71 and 73.52, respectively).

A second reanalysis used a combination of time and residual errors. Ten seconds for each residual error was added to the time it took subjects to complete an exercise. We reasoned that by doing this we would better estimate the time that would have been required to produce a completely clean text, for all subjects, and consequently reduce noise in the data. However, results were essentially the same as the analysis shown in Table IV.

We looked also at trends over exercises in the data. The slope of the practice function (learning curve) was steeper in the *short* condition than in the long condition ($F(1,72) = 3.93$, $p < 0.07$), and there was a larger quadratic component for subjects in the *long* condition than for subjects in the *short* condition ($F(1,72) = 3.92$, $p < 0.07$). Thus, while the two groups were about equal after the initial training exercises, subjects in the *long* condition increased their speed less rapidly with practice and appeared to approach a lower asymptotic performance level than did subjects in the *short* condition. A plausible interpretation is that spelled-out commands carry a mnemonic advantage that offsets the disadvantage of more keystrokes, but only until their meanings have been mastered.

The high school student subjects answered a questionnaire at the end of the task. Of six 5-point rating scales, two revealed significant differences. Question 2 asked subjects how difficult they found the task. Lower numbers indicate greater difficulty. Subjects in the old scope condition thought the task significantly easier than did subjects in the new scope condition ($\bar{X} = 3.42$ and 3.17; s.e.m. = 0.20 and 0.20, respectively; $F(1,36) = 5.16$, $p < 0.05$). The main effect for vocabulary was also reliable; subjects in the old and new vocabulary conditions thought the task easier than did subjects in the random vocabulary condition ($\bar{X} = 3.38$, 3.44, and 3.06; s.e.m. = 0.17, 0.16, and 0.19, respectively; $F(2,36) = 5.22$, $p < 0.03$).

A similar pattern of answers was given to Question 4, which asked subjects how difficult it would be to learn to work with such a system in the future. Subjects in the *old scope* condition anticipated less difficulty than did subjects in the *new scope* condition ($\bar{X} = 3.88$ and 3.58; s.e.m. = 0.14 and 0.12; $F(1,36) = 6.85$, $p < 0.025$). Subjects in the *old vocabulary* condition thought that learning to use such a system would be easier than did subjects in either the *new* or *random vocabulary* conditions ($\bar{X} = 3.88$, 3.63, and 3.69; s.e.m. = 0.13, 0.16 and 0.12, respectively; $F(2,36) = 8.89$, $p < 0.01$), whose times did not differ significantly from each other.

Subject's perceptions of difficulty were not unfounded. Correlation coefficients (Pearson r) between ratings on Questions 2 and 4 and the total time taken on all exercises were $r = -0.44$, and $r = -0.49$, respectively, both statistically significant ($p < 0.001$).

TABLE IV. Mean Time to Complete Test Exercises (standard error of means)

Groups (80)	
Secretarial students	1886
High school students	1866
Vocabulary (108)	
Old	1799
New	1801
Random	2029
Scope (88)	
Old	1657
New	2039
Length (88)	
Long	1816
Short	1936

DISCUSSION

The two major observations emerging from Study 2 were that it made little difference what particular words were chosen for command names, but that the mapping of differences in command names to differences in command syntax (and function) had a very large effect. Task completion time, total characters typed, residual text errors, and combined time

were essentially the same for beginners who "alleged" text segments, as compared with those who "deleted" or "omitted" them. Indeed, none of our secretaries, and only a few of the high school student typists, spontaneously commented on the oddity of the random names. There were differences in rated difficulty of the task between vocabulary conditions, but they did not favor the new terminology obtained from previous typists' spontaneous descriptions over one particular system designer's set of words. Overall, the old vocabulary words, which were seldom nominated by typists in Study 1, created the greatest impression of ease and learnability (although by a slim margin). Perhaps words chosen by designer intuition represented more precisely the system characteristics of which the user must be made aware. These words are probably part of typists' recognition vocabulary even if they do not produce them spontaneously. Overall, the randomly chosen words created the least favorable impression. However, the differences were small, and it must be remembered that while effects of vocabulary variations on performance were generally consistent with subjective opinions, only the attitude effects were statistically reliable.

There are important qualifications of the conclusion that vocabulary effects are small. First, note that actual words or their initial letter abbreviations were always used as commands. Thus, our results cannot be taken as grounds for choosing commands that are unpronounceable or meaningless strings. Moreover, only subjects with no previous knowledge of text editing were used; a different population might conceivably be more sensitive to some kinds of vocabulary differences. It should also be noted that we studied a very small command set that was used only for one 2-hour session. Considerations of precise semantic distinction and confusability may be more important in larger sets. Meaningful names may also be more beneficial when users need to remember commands over longer periods. Indeed, Barnard et al. [1] have obtained (unpublished) results favoring specific over general terms for editing commands in a similar experiment to ours. They used a larger command set that had to be remembered over several days. However, significant effects were found only for number of errors, not for total time to produce correct text. Recently, Gomez et al. [4], as part of a study of individual differences in learners, have repeated our "old" versus "random" vocabulary comparison in an almost identical experiment, but with an additional test after a week's interval. While their typist subjects could more easily recall the "old," meaningful command verbs than the (different) set of random verbs when asked merely to remember them, the typists performed text editing with identical facility using "old" and "random" vocabulary. Unfortunately, we know of no experimental research, to date, on natural vocabulary effects for very large command sets.

In Study 2 there was substantial advantage in learning time of the old over the new scope, despite our attempt to better map the new scope onto that implicit in the way subjects described the operations. Subjects became proficient more quickly using a system that distinguished line-internal versus whole-line operations through the use of different command names. Subjective ratings also significantly favored the old scope conditions. One explanation we can offer for these results is as follows. In the editors studied, within-line operations required a syntactic construction (command/old/new/) different from that used in whole-line operations (command). The use of different command names when different syntactic constructions are required may facilitate learning and remembering to execute the proper construction in the proper context. This advantage apparently outweighs any disadvantage of violating prior concepts and naming preferences imported from the noncomputer editing environment. One can only guess the outcome if the constructions for whole-line and within-line operations had been the same, for example, if both had made use of physical pointers to beginning and endpoints. Under such circumstances, which might themselves better match preconceptions of the task, using the same words for both operations might possibly be better. However, the lesson we believe should be taken from the present results is that it is probably better to match the language to the structure of the task being learned than to blindly mimic the language used in prior similar tasks. It needs to be remembered that commands are not only names for the effects desired by users but also cues to the users about what they must do next. Indeed Ledgard et al. [7] produced an easier-to-use editor, not by making its language match users' preconceptions, but, apparently, by trying to make its commands and syntax easy to comprehend as descriptions of their specialized functions.

CONCLUSIONS

We have summary comments to offer on two general issues: research methodology and principles of command name choice. First, we have found that novice users do not use the same language as system designers to describe operations done in a text-editing task. Indeed, for the most part, one potential user does not even use the same words as another, or even the same words on different occasions. One important lesson from this is that intuitive guesses as to what is a "common" or "natural" name for a command are likely to be hazardous. One person's obvious name may not be another's. (See, e.g., [8] and *Communications of the ACM*, June, 1981, pp. 404–406 for some spirited debates about what words are most natural.) This also means that comparing two systems, one of which has features intuitively chosen to be natural by its designers, for example, [7], is probably primarily a test of the designer's personal art, not of the principle of naturalness. Such a test requires a systematic method for empirically determining what natural users naturally use. We feel the observational and analytic methods employed for this purpose in Study 1 produced sufficiently clear results to warrant further application and refinement.

The last issue we wish to discuss is the one we started with: how to choose command names. We began our investigations with the hypothesis that commonly used words for similar operations in the noncomputer environment would be best. Such words are familiar and have at least approximately correct known meanings, so they should demand minimal new learning. This line of reasoning has a wide following among computer scientists and human-factors psychologists alike [2,7,8]. It now appears to be a somewhat naively undifferentiated view. There is no question that familiar words, and words with known meanings related to the objects they are to name, are easier to learn as associates (see [5, pp. 101–110] for a review of some of the extensive experimental literature). The catch is that command names require other properties than ease of learning as isolated associative responses. One is precision of application. Familiar, easy to think of words are not necessarily precise. In fact, one of the factors that can make a word common is that it can be used for many purposes. For example, dividing the words used in Study 1 into those most and least frequently used, shows the more popular ones to be the ones less consistently applied, that is, their use is more evently spread over the five operations. (For this analysis we considered only words used at least ten times, to make negligible an obvious statistical artifact.) Clearly, constructing a command set of words whose natural meanings tend to lead to misuse is not desirable.

Moreover, the learning of a set in which the names are interchangeable to some degree will be retarded by the interference between conflicting responses to the same stimuli, a learning principle that is perhaps even better established than the benefit of previous association [5].

A reasonable compromise between the benefits and potential disadvantages of common words might be achieved by choosing relatively low frequency words that nevertheless are recognized as having appropriate meanings [2]. A matching test could be employed. Such words would have high enough preexisting associative value and familiarity to make learning easy (the learning rate versus frequency function is negatively accelerated), but not lead to inappropriate spontaneous use. We have no direct evidence of the correctness of this conjecture. In fact, the results of Experiment 2 give little support to the belief that word-choice is of much importance at all. However, as stated previously, we are reluctant to conclude that word choice would not be influential in larger command sets used for longer periods.

An apparently more important property of command names is that they differentiate appropriately among the actions the system requires of the user, for example, the arguments or syntax to be provided with the command. Independent of what particular words are used, however, our results urge close attention to the mapping of differences in command names to differences in syntactic and functional distinctions. When varying constructions are required on the part of the user, or varying effects are obtained, different command names should be used. Untrained users cannot be expected to spontaneously generate terms that are appropriate for a system they do not know, but system designers at least have the necessary information. Programmers are certainly likely to choose different names for different system actions. Moreover, there seems a fair chance that the words designers choose in trying to be precise, rather than systematically popular, will also meet the criteria of low but sufficient familiarity and recognizably apt meaning. Together, then, this set of hypotheses may explain why the command names chosen by ED system developers produced as rapid initial learning as did command names based on the quite different editing vocabulary employed by manual typists.

The final conclusion from our results we think is clear: rational design of commands and command names for usability requires deeper understanding than is captured in the slogan "make the language natural."

REFERENCES

1. Barnard, P., Hammond, N.V., and Morton, J. Learning and remembering interactive commands. In Proc. ACM 1st Int. Conf. Human Factors in Computer Systems, Gaithersburg, Md., March 1982.
2. Black, J.B., and Sebrechts, M.M. Facilitating human–computer communications. *Applied Psycholinguistics 2* (1981), 149–177.
3. Furnas, G.W., Landauer, T.K., Gomez, L.M., and Dumais, S.T. Statistical semantics: Analysis of the potential performance of keyword information systems. *Bell Syst. Tech. J.* To appear.
4. Gomez, L.M., Egan, D.E., and Bowers, S. Personal communication, 1981. Gomez and Egan are at Bell Labs, Murray Hill, N.J. 07974.
5. Goss, A.E., and Nodine, C.F. *Paired-Associate Learning. The role of Meaningfulness, Similarity and Familiarization.* Academic Press, New York, 1965.
6. Kruskal, J.B., and Wish, M. *Multidimensional Scaling* (Sage University Paper Series on Quantitative Applications in the Social Sciences, 07–011). Sage Publishers, Beverly Hills, Calif., 1978.
7. Ledgard, H., Whiteside, J.A., Singer, A., and Seymour, W. The natural language of interactive systems. *Comm. ACM 23*, 10 (Oct. 1980), 556–563.
8. Norman, D.A. The trouble with UNIX. *Datamation 27* (1981), 139–150.
9. Reisner, P. Formal grammar and factors of design of an interactive graphics system. *IEEE Trans. Softw. Eng. SE-7*, 2 (Mar. 1981), 229–240.
10. Streeter, L.A., Ackroff, J.M., Taylor, G.A., and Galotti, K.M. Personal communication, Bell Telephone Laboratories, Holmdel, N.J. 1979.
11. Winer, B.J. *Statistical Principles in Experimental Design.* McGraw-Hill, New York, 1971.

CR Categories and Subject Descriptors: H.1.2 [**Models and Principles**]: User/Machine Systems—*human factors, human information processing*; H.4.1 [**Information Systems Applications**]: Office Automation–*word processing*

General Term: Human Factors

Additional Key Words and Phrases: user interfaces, human–computer interaction, dialogue

Received 10/81; revised 1/82; accepted 8/82

Correction. In the Technical Note "On the Synthesis of Decision Tables" by B. Srinivasan [February 1983, pp. 135-136], the affiliation of the author was incorrectly given as the National Institute of Technology, Singapore. Srinivasan's correct affiliation is Indian Institute of Technology, Kanpur. The error is regretted.