Int. J. Man-Machine Studies (1985) **22**, 403–423

The transfer of text-editing skill

MARK K. SINGLEY AND JOHN R. ANDERSON
Department of Psychology, Carnegie-Mellon University, Pittsburgh, Pa. 15213, U.S.A.

(*Received 31 August 1984*)

Computer-naive subjects were taught to use either one or two line editors and then a screen editor. Positive transfer was observed both between the line editors and from the line editors to the screen editor. Transfer expressed itself in terms of reductions in total time, keystrokes, residual errors, and seconds per keystroke. A simple two-component model of transfer is proposed that allows for the differential practice of general and specific components when learning a skill.

Introduction

Despite its relatively short research history, text editing is well on its way to becoming one of the most thoroughly-understood cognitive skills. There are many possible reasons for this, but perhaps one is that research has progressed in a fairly orderly fashion. Newell & Simon (1972) point out that there is a certain research agenda imposed on cognitive psychologists by logical necessity. First, a performance theory must be worked out in detail; only then can issues of learning be addressed. The underlying logic is that one must understand the endpoints of a transition before one can understand the transition itself.

Fortuitously, this has been exactly the course taken in text-editing research. Card, Moran, & Newell (1976, 1980*a*, *b*, 1983) have been working for some time on the details of a performance theory, and the result has been a series of information-processing models at various levels of detail that account for an impressive percentage of error-free, expert behavior. As the work of Card *et al.* progressed, other researchers took up the study of the more complex, second-order phenomenon of learning (Roberts, 1979; Egan & Gomez, 1982; Mack, Lewis, & Carroll, 1983). At present, issues of learning are perhaps receiving most of the attention of researchers in the field. Although a coherent theory of learning has not yet emerged, progress has been steady. Among the more significant accomplishments have been the identification of novice misconceptions based on faulty analogies with typewriters (Bott, 1979; Mack *et al.* 1983; Rumelhart & Norman, 1981; Douglas, 1983), the role of individual differences in learning (Egan & Gomez, 1982), and the measurement of learning rates for different editors (Roberts, 1979).

The aim of our research is to take another step in the research program for text editing, and to begin exploring issues of transfer. In a sense, transfer of skill is a higher-order phenomenon than learning. To understand it, one must not only understand performance and learning in one skill, but also in another. One might object that current theories of learning in text editing are not strong enough to support the weight of a theory of transfer. However, even if a defensible theory of transfer were not possible at this time, studying it now would still be justified.

0020-7373/85/040403+21$03.00/0

First of all, the study of transfer can provide important constraints on the learning theory. This is an important elaboration of the Newell & Simon position presented earlier. By applying the theory of lower-order phenomenon to the understanding of a higher-order phenomenon, much about the lower-order phenomenon may be revealed. This exchange of information between levels is quite similar to the interchange of ideas between basic and applied science. Sometimes, it is only through application that certain theoretical flaws surface. This does not imply that the study of the higher-order phenomenon should be done first, only that it should trail closely behind.

Aside from the theoretical motivations for studying transfer, there are a host of practical ones. Issues of transfer become increasingly important in a society where rapid technological change keeps conditions of life and work in constant flux. For example, computer science curriculum designers today are faced with the thorny problem of what to teach their students, knowing that current technology will be outdated within the decade. The issue is clearly how to maximize transfer to new technology. In these circumstances, curriculum designers are forced to choose some representative set of experiences for their students and hope that students can adapt to the actual situations encountered. We hope that work on transfer will allow educators to make a more informed choice.

Interacting with these educational questions are questions of design. We share with Nakatani (1983) the perception that, in comparison with simpler machines (automobiles, copiers), transfer among different kinds of computer systems is relatively difficult. Perhaps transfer should be considered more explicitly in the design process. Of course, even if it were possible, designing for optimal transfer might sometimes overly restrict the functionality of a new technology. This is just one more example of a trade-off inherent in the design process (Norman, 1983).

PREVIOUS RESEARCH ON TRANSFER

Despite the obvious importance of studying transfer of complex skills, the subject has received little attention in the psychological literature. This is for obvious reasons: studying the acquisition of a single skill requires a substantial investment; studying the acquisition of two or more requires twice as much.

Perhaps the first psychologist to express interest in transfer was Thorndyke (Thorndyke & Woodworth, 1901). Thorndyke took issue with the prevailing opinion concerning education during his time, namely, the *Doctrine of Formal Discipline.* This doctrine claimed that studying such otherworldly subjects as Latin and geometry were of significant value because they served to "discipline" the mind. After finding no substantial support for this claim, Thorndyke proposed an alternative view, the *Theory of Identical Elements.* The theory stated that training in one kind of activity would transfer to another only if the activities shared common elements. Of course, without an explanatory theory it was difficult to decide just what these elements were, and there has been some debate on this point (Ellis, 1965). However, it is generally true that Thorndyke saw transfer as being more limited in scope than those who held to the Doctrine of Formal Discipline.

Transfer of declarative knowledge was of significant interest to the verbal learners in the middle of the century. A typical study involved the learning of a list of paired associates followed by the learning of a new list that differed from the first in some theoretically meaningful way. A significant result from these studies was that, if the stimuli in the second list were similar in meaning or sound to the stimuli in the first

list, transfer was positive (Yum, 1931). However, if the similarity was in the responses rather than the stimuli, transfer was negative (Bruce, 1933). Although these studies may have some bearing on the study of cognitive skill, it is clear that they deal primarily with facilitation and interference in declarative memory.

The most recent expression of a theory of transfer was that proposed by Moran (1983). He presents a task analysis technique called ETIT that conceptualizes the use of a device as a mapping from the user's goals and intentions (the external task) into the operating principles or semantics of the device (the internal task). One can use this technique to predict transfer among devices by merely gauging the differences between the mappings for different devices. One performs set computations on the two sets of mapping rules, and the degree of overlap predicts the degree of transfer. As Moran points out, this work is at a very early stage, and chances are good that predicting transfer with any degree of accuracy will require a more complex theory. However, ETIT has already been used to account for errors made by novices learning a screen editor (Douglas & Moran, 1983) and the fact that novices can perform certain tasks they have not been taught (Douglas, 1983).

While Moran's attempt looks promising, it seems premature to model transfer without a stronger empirical base. Given the dearth of research on transfer of complex cognitive skills (Anderson, 1980), any theoretical account now must necessarily be under-constrained. At present, it is still largely an open question as to whether transfer occurs at all. Indeed, a growing body of literature on analogical reasoning (Hayes & Simon, 1977; Gick & Holyoak, 1983) suggests that in some situations people are quite poor at transferring knowledge to new situations. In addition, research on text editing has shown that, when people do transfer, they often select knowledge that is inappropriate in the new context (Bott, 1979; Mack *et al.*, 1983) and performance may suffer. This view that analogical transfer is limited at best and harmful at worst (Halasz & Moran, 1982) contrasts with work in artificial intelligence (Winston, 1979; Carbonell, 1983), where achieving what might be thought of as minimal human competence at analogical transfer has been highly prized and is exceedingly difficult. However, given the available evidence, we must conclude that transfer is uncertain in most situations.

THE PRESENT STUDY

As we have tried to point out, the conditions and effects of transfer of cognitive skills such as text editing and programming are largely unknown, both qualitatively and quantitatively. Theoretical progress on this topic has been limited by the scant empirical database on transfer. Accordingly, the purpose of the present study was to observe and characterize transfer in the domain of text editing. The study involved teaching groups of computer-naive subjects to use one or two line editors and then a screen editor. The following were of particular interest.

(a) *The magnitude of transfer between different editors.* Would transfer be positive, negative, or nonexistent? How would transfer between the line editors compare with transfer from the line editors to the screen editor? Although we did not do an in-depth formal analysis of the structural similarity of the editors in our study, an informal analysis suggested that, whereas the line editors were quite similar, the line editors and the screen editor shared no obvious features. Perhaps the magnitudes of transfer would reflect this difference.

(b) *The shape of learning curves before and after transfer.* A ubiquitous finding in cognitive psychology is that learning curves of all kinds can be fitted to power functions

(Anderson, 1982; Card *et al.*, 1983). Given a particular subject population, a particular power function defines the time course of the acquisition of a particular skill. Given this fact, it may be possible to characterize positive transfer from one skill to another as a displacement of the standard power function for the second skill. In other words, it may be possible to characterize positive transfer solely in terms of prior trials (Rosenbloom & Newell, in preparation).

(c) *The possible advantage of learning to use two line editors instead of one line editor.* We thought subjects might show greater transfer to a screen editor having learned to use two line editors rather than one line editor. This speculation was founded on the belief that those who learned to use two line editors would have a more generalized skill that would apply more broadly to a new editor (Anderson, 1982).

Method

SUBJECTS

Subjects were 24 women between the ages of 18 and 30, from a local secretarial school. None of the subjects had any computer experience, but all could type proficiently. Subjects were balanced across various conditions of the experiment for typing speed ($\bar{X} = 41$ wpm) and performance on a standardized cognitive test of spatial memory, the Building Memory Test (Ekstrom, French & Harman, 1976). This test was found by Egan & Gomez (1982) to be a fair predictor ($r = -0.58$) of initial performance on a text editor.

MATERIALS

Subjects learned from a set of three commercially-available text editors. Two of these editors, UNIX "ED" (Kernighan, 1981) and VMS "EDT" (Digital Equipment Corporation, 1982) belong to the genre known as line editors, whereas the third editor, UNIX "EMACS" (Gosling, 1981), belongs to the genre known as screen editors. Line editors differ from screen editors in basic editing strategy. Line editors display the contents of the file only upon request and force users to enter abstract commands that specify edits on a line-by-line basis. Screen editors, on the other hand, fill the screen with the contents of the file and allow users to edit the contents explicitly by moving to a particular location by means of a cursor. Screen editors are generally seen as a significant advance over the older line editors, and, in fact, have been found superior on measures of learnability and expert performance (Roberts, 1979; Gomez, Egan, Wheeler, Sharma, & Gruchacz, 1983).

Subjects were taught a minimum core set of commands for each editor. These commands were totally sufficient for the kinds of edits our subjects had to perform. In the line editors (ED and EDT), the core set included commands for:

- printing, deleting, inserting and replacing lines and
- substituting strings within lines (the substitution command also provided for string insertion and deletion).

In the screen editor (EMACS), the core set included commands for:

- moving the cursor forward, backward, up and down, and
- deleting characters, words, and strings.

TABLE 1
*Command summary for three editors**

Command type	Editor	Command	Action
Locative	ED	1,$p	Prints all lines of the file
		3p	Prints the third line
		.p	Prints the current line
		.=	Prints the line number of the current line
		[return]	Prints the line following the current line
	EDT	t whole	Prints all lines of the file
		t 'dog'	Prints the first line following the current line that contains 'dog'
		t-'dog'	Prints the first line before the current line that contains dog
		t	Prints the current line
		[return]	Prints the line following the current line
	EMACS	↑f	Moves cursor forward one character
		]f	Moves cursor forward one word
		↑b	Moves cursor backward one character
		]b	Moves cursor backward one word
		↑a	Moves cursor to beginning of line
		↑e	Moves cursor to end of line
		↑p	Moves cursor to previous line
		↑n	Moves cursor to next line
Mutative	ED	.a	Inserts lines after the current line ('.' exits the insert mode)
		.d	Deletes the current line
		.c	Replaces the current line ('.' exits the insert mode)
		s/a/b/p	Substitutes the first occurence of 'a' with 'b' on the current line and prints the line
	EDT	i	Inserts lines after the current line (↑z exits the insert mode)
		d	Deletes the current line
		r	Replaces the current line (↑z exits the insert mode)
		s/a/b	Substitutes the first occurence of 'a' with 'b' on the current line and prints the line
	EMACS	↑d	Deletes the character marked by the cursor
		]d	Deletes the word marked by the cursor
		[delete]	Deletes the character to the left of the cursor
		↑k	Deletes from the current cursor position to the end of the line
		a	Inserts the character 'a' at the current current cursor position (EMACS is in insert mode by default)

* In the table, ↑ denotes a control character and] an escape character.

Of course, the character of the commands differed markedly between the line editors and the screen editor. The majority of EMACS commands pertained to moving the cursor and involved special terminal keys. In all editors, subjects were spared from learning the procedures for reading and writing files, and were instead fed files automatically by an experimental program. See Table 1 for a listing of the core set of commands for each of the editors.

Subjects edited sections of a book on cognitive psychology that resided on a local computer system. The book was sectioned into 18-line files, and each file was randomly mutilated by a text-mutilation program. The program performed six of 12 possible mutilations on each file. The 12 mutilations were defined by crossing the editing operations insert, delete and replace by the data objects character, word, string and line. It took two files to cover all 12 mutilations; the same mutilations occurred in every other file. Each file constituted a single trial.

The subjects' task was to correct the errors introduced by the mutilation program. They worked from a marked-up copy of the files placed in a loose-leaf binder (see Fig. 1 for a sample page). The binder sat flat on a table beside the terminal. Each page corresponded to a single file.

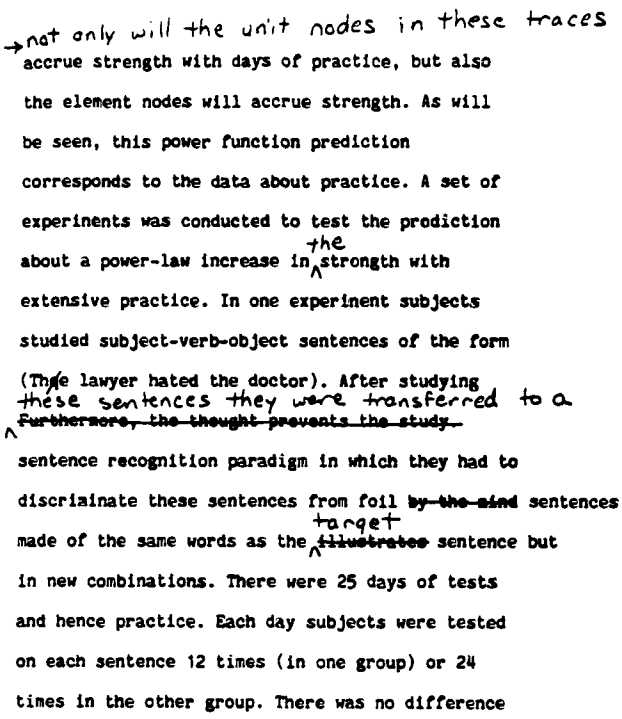

FIG. 1. Sample page of corrections.

DESIGN

The study used a 2 × 2 between-subjects design with two control groups. The first factor was the number of line editors the subjects learned to use (one vs two), and the second was the initial line editor used (ED vs EDT). The two control groups did not learn to

use line editors; their first exposure to text editing was EMACS. One of the control groups spent the entire experiment editing with EMACS. (This was the control that would reveal whether transfer to EMACS from the line editors was positive or negative.) The other control group practised typing at the terminal prior to editing with EMACS. This group typed for the amount of time the experimental groups spent using the line editors. (This was the control that would reveal the perceptual-motor component of transfer.)

PROCEDURE

Several days prior to the experiment, subjects were pretested on the Building Memory Test and assigned to conditions based on this score and also on typing speed. On the same day, subjects received a brief introduction to the computer and the computer terminal. No explicit instruction on text editing was given.

The experiment itself consisted of six consecutive days of text editing. Each day consisted of a three-hour session interrupted by two ten-minute breaks after the first and second hours. Subjects were run in pairs in a quiet experimental room. Each subject worked independently on her own Zenith H19 terminal. Throughout the experiment, the two subjects were the sole users of a DEC VAX-750 computer.

On the first day of the experiment, all subjects except those in the typing control group were given a brief introduction to the set of comnands they would be using that day. This introduction consisted of a short description of each command, followed by a demonstration on the terminal. This introduction lasted approximately 30 minutes. The subjects then began editing at the terminals. An experimenter was present in the room at all times to answer any questions or help with particularly difficult problems. Experimenters were told not to intervene unless a subject asked for help. A single tutor was designated for each editor, so that all subjects' experiences with a single editor would be similar. As experimenter was totally confounded with editor in the experiment, the results should not be regarded as a totally valid comparison of the editors.

The subjects spent the first two days practising with their first editor. On the third day, those subjects in the two-line-editor group switched to their second editor (either ED or EDT), whereas the other subjects remained on their first. However, all the subjects received a second introduction to the set of commands they would be using. (This constituted a review for those subjects who did not switch.) In this way, the amount of formal instruction received by subjects was constant across groups. On the fifth day of the experiment, all experimental subjects and the typing control group transferred to EMACS. After receiving formal instruction on the commands, these subjects spent the last two days practicing EMACS. (The EMACS control group spent all six days learning to use EMACS. They received formal instruction on the first, third, and fifth days.)

Those subjects in the typing control group spent the first four days typing the manuscript that the experimental groups were editing. In addition to incorporating all the corrections marked on the manuscript, subjects had to correct typing mistakes as they were made. This rule was enforced by a program that checked the stream of keystrokes against a target file and deactivated the keyboard once a difference was detected. Subjects could only reactivate the keyboard by pressing the delete key, which erased the mistake. This practice resulted in a level of frustration similar to that

experienced by subjects in the experimental groups. Thus, the typing control group had experience reading the manuscript, interpreting the edits, and interacting through the terminal.

Keystroke data accurate to within one second were collected for all subjects. In addition, the edited versions of the mutilated files were saved to allow for error-checking. Finally, a log of the interaction between the subject and the editor was kept to facilitate interpretation of the keystroke data.

Results

The results section is organized into three parts. The first part presents learning data for the three editors. The second analyses transfer between the line editors, and the third analyses transfer from the line editors to the screen editor.

LEARNING FOR THE THREE EDITORS

Figure 2 shows practice curves for the three editors used in the experiment. These curves were derived from the two experimental groups that spent four days on a single line editor and the control group that spent six days on EMACS (only the first four days are presented here). The results are expressed as a measure that approximates the number of seconds spent per correct editing operation. The measure was calculated by first adjusting the total time on a trial (T) by incrementing the time by one sixth for every error (E) committed:

$$T_{adj} = T + \left(\frac{T}{6}\right)E$$

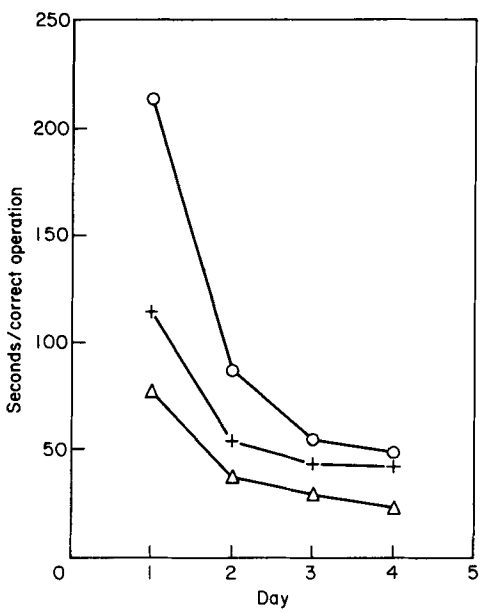

FIG. 2. Learning curves plotting seconds per correct operation for three editors. + = ed; ○ = edt; △ = emacs.

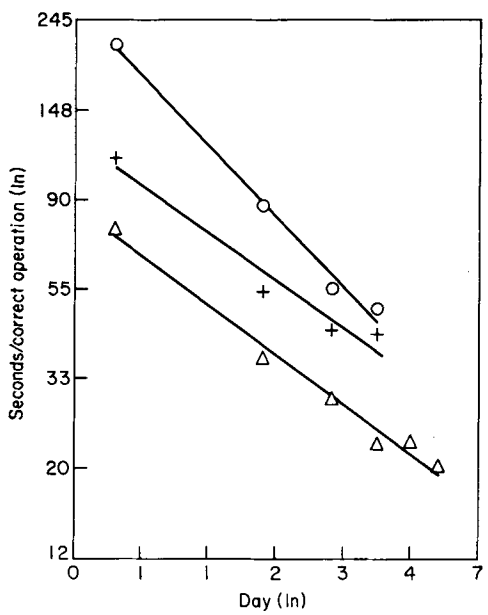

FIG. 3. A log-log transformation of the learning data in Figure 2. + = ed; ○ = edt; △ = emacs.

This adjusted total was then divided by six to arrive at time per correct editing operation. Errors were defined as mismatches in character sequences between the subject's edited file and the target file. Errors were scored in an all-or-none fashion, with a maximum of six errors per trial.

The learning curves show that, on all four days, EDT is the slowest editor and EMACS is the fastest. Also, EMACS appears to be levelling off at a much lower asymptote than the two line editors (an advantage of approximately 20 seconds per edit, or 2 minutes per trial). A two-way repeated measures analysis of variance yielded main effects for editor ($F = 8{\cdot}6$; $df = 2,9$; $P < 0{\cdot}01$), day ($F = 23{\cdot}4$; $df = 3,27$; $P < 0{\cdot}01$), and also an interaction ($F = 3{\cdot}0$; $df = 6,27$; $P < 0{\cdot}03$). The interaction implies that the editors are being learned at different rates. Subsequent Newman-Keuls Multiple Range tests revealed that, across days, EDT was significantly slower than both ED ($F = 3{\cdot}68$, $df = 9$, $P < 0{\cdot}05$) and EMACS ($F = 5{\cdot}8$, $df = 9$, $P < 0{\cdot}05$). The ED-EMACS difference was non-significant. Also, EDT was being learned significantly faster than EMACS ($F = 3{\cdot}84$; $df = 3,18$; $P < 0{\cdot}05$). Other comparisons of learning rates were non-significant.

To corroborate this analysis, we fitted the learning curves to power functions (see Fig. 3). Table 2 presents the parameters for the equations. The power law of practice has the general form

$$T = AP^{-b}$$

where T is time, P is number of trials (or days), A is the time on trial one, and b is the slope of the function on a log-log plot (usually between zero and one). We can see from our equations that,although EDT is slowest on day one (it has the largest intercept on a log-log plot), it is also learned the fastest (it has the steepest slope). However, EMACS still emerges as the superior editor.

The reasons for the advantage of a screen editor such as EMACS over line editors such as ED and EDT have been discussed elsewhere (Roberts, 1979; Gomez *et al.*,

TABLE 2
Power functions for three editors

Editor	Equation
(1) ed practice curve	$T = 4{\cdot}3P^{-0{\cdot}53}$
transfer curve	$T = 3{\cdot}7P^{-0{\cdot}20}$
transfer curve displaced 2·0 days	$T = 4{\cdot}4(P+2{\cdot}0)^{-0{\cdot}63}$
(2) edt practice curve	$T = 4{\cdot}8P^{-0{\cdot}79}$
transfer curve	$T = 3{\cdot}9P^{-0{\cdot}34}$
transfer curve displaced 1·8 days	$T = 5{\cdot}0(P+1{\cdot}8)^{-1{\cdot}0}$
(3) emacs practice curve	$T = 3{\cdot}9P^{-0{\cdot}55}$
transfer curve	$T = 3{\cdot}4P^{-0{\cdot}41}$
transfer curve displaced 0·9 days	$T = 4{\cdot}0(P+0{\cdot}9)^{-0{\cdot}32}$
transfer curve displaced 0·3 days	$T = 3{\cdot}6(P+0{\cdot}3)^{-0{\cdot}55}$

1983). A popular view is that screen editors offload spatial memory by providing a static display of text and a method of addressing characters by cursor position. More generally, differences between editors have been ascribed to differences in the number of keystrokes required to perform edits (Card *et al.*, 1983). In our experiment, the difference between ED and EDT is most likely due to different procedures for locating lines. Specifically, lines are addressed by line number in ED and by content in EDT. In fact, both line addressing methods are available in both editors, although we chose to teach only one in each. So, to locate line 12 in ED, one merely types *12p*; whereas in EDT, one types *t 'string'*, where *string* is some sequence of characters unique to line 12. In most cases, the latter method involves not only more keystrokes but also more mental preparation time, especially in novices. This may explain the observed differences in both initial and asymptotic performance in the two line editors.

In a complex skill such as text editing, the exact shape of the learning curves is most likely determined by a variety of interacting factors. Two of these factors might be fewer episodes involving error recovery and the acquisition of more efficient editing strategies. Both of these factors would not only combine to reduce total time, but would also reduce the total number of keystrokes. Figure 4 plots the average number of keystrokes per trial for the three editors on each day. A two-way analysis of variance yielded main effects for editor ($F = 8{\cdot}7$; $df = 2,9$; $P < 0{\cdot}01$) and day ($F = 4{\cdot}9$; $df = 3,27$; $P < 0{\cdot}01$). It is interesting to note that the pattern of keystroke data mirrors almost exactly the pattern of timing data presented in Fig. 2. Such a correspondence suggests that, in learning as well as expert performance (Card *et al.*, 1983), the number of keystrokes correlates highly with total time.

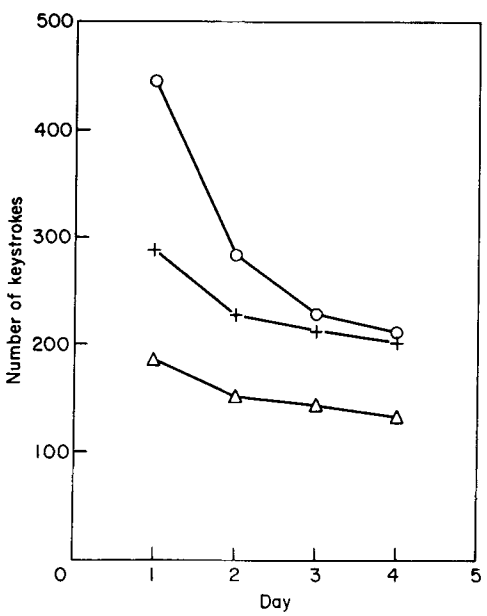

FIG. 4. Learning curves plotting number of keystrokes per trial for three editors. + = ed; ○ = edt; △ = emacs.

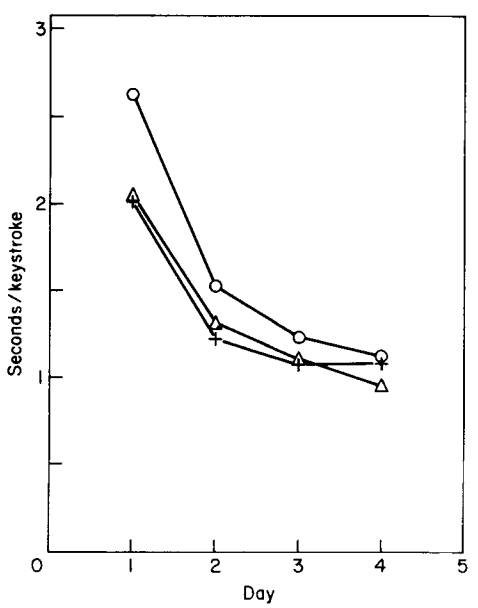

FIG. 5. Learning curves plotting seconds per keystroke for three editors. + = ed; ○ = edt; △ = emacs.

Can the decrease in the number of keystrokes account for all the learning that is taking place? If the decrease in the number of keystrokes is solely responsible for the shape of the learning curves, then seconds per keystroke should be constant across days. Figure 5 shows that seconds per keystroke decreased markedly across days, suggesting that subjects were either becoming faster typists or spending less time thinking. Given the magnitude of the effect and the fact that our subjects were already skilled typists, the latter seems more plausible. An analysis of variance for this data revealed that seconds per keystroke did not differ significantly among the editors. Thus, it appears that the difference between editors is entirely a function of the different number of keystrokes required to successfully execute commands. On the other hand, speed-up is due to both a decrease in number of keystrokes and time per keystroke. Neves & Anderson (1981) have shown a similar pattern of results in a geometry theorem-proving task.

TRANSFER BETWEEN LINE EDITORS

Transfer from EDT to ED

Figure 6 shows the massive amount of transfer from EDT to ED, in logarithmic form. The curves compare data for subjects who spent four days learning to use ED (ed practice curve) with those who spent two days learning to use ED after first learning to use EDT (transfer from edt). The curves are aligned so that both groups' performance on the first day of ED are compared. As can be seen, exposure to EDT prior to ED resulted in a substantial improvement in performance on the first day, a saving of

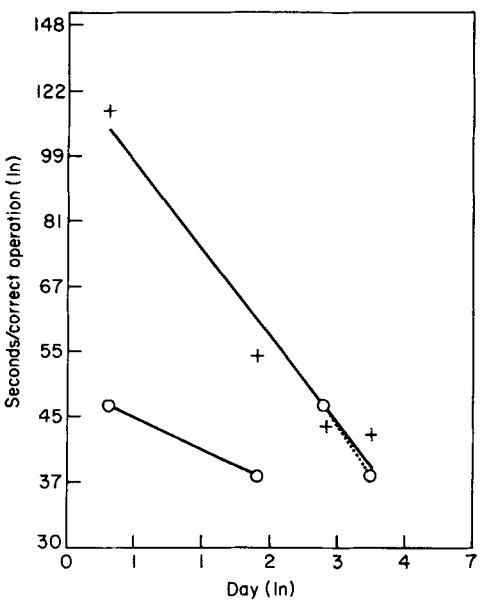

FIG. 6. A log-log plot showing transfer from EDT to ED in terms of seconds per correct operation. The transfer curve is plotted to facilitate comparison of performance on the first day of ED. + = ed practice curve; O—O = transfer from edt; O- - -O translated 2·0 days.

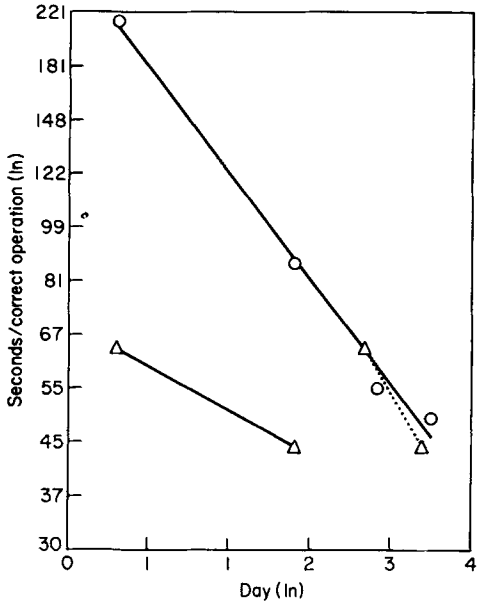

FIG. 7. A log-log plot showing transfer from ED to EDT in terms of seconds per correct operation. The transfer curve is plotted to facilitate comparion of performance on the first day of EDT. O = edt practice curve; Δ—Δ = transfer from ed; Δ- - -Δ = translated 1·8 days.

approximately 70 seconds per edit (7 minutes per trial). Indeed, it appears that the transfer subjects' performance on the first and second days is nearly equivalent to the practice curve points for the third and fourth days.

Rosenbloom & Newell (in preparation) point out that prior experience (or transfer) can be modelled using the standard power function for the task, adjusted for prior trials. The idea is that any prior experience the subject has had translates into some measurable displacement of the power function. Rosenbloom & Newell expand the power law of practice into the following form:

$$T = A(P+E)^{-b}$$

where E is the number of prior trials. (A negative value of E would represent negative transfer.) Figure 6 shows an attempt to model our transfer data in terms of the Rosenbloom & Newell proposal. All curves have undergone log-log transformations, and the value of E for the translated curve is 2·0. This means that, on day three, those subjects who spent the first two days learning to use EDT performed just as well on ED as those who spent the first two days learning to use ED. Table 2 gives the parameters for the two equations. We can see that, with the adjustment for prior trials, the transfer function is a fairly good approximation of the standard power function for ED.

Transfer from ED to EDT

Figure 7 shows the transfer from ED to EDT. The savings on the first day are even greater than in the previous case (150 vs 70 seconds per edit). However, the Rosenbloom & Newell analysis for this data shows that, although the adjusted transfer curve is a fairly good fit to the standard power function for EDT (see Table 2), the number of days of prior trials is only 1·8. This is slightly less than the estimate of 2·0 that we obtained for the transfer from EDT to ED.

Number of keystrokes and keying rate

To characterize further the positive transfer between the line editors, we compared the number of keystrokes per trial for the various practice and transfer conditions. These comparisons were based on data from the first and second days of editing with a particular line editor. Thus, the data are from the third and fourth days of the experiment for the transfer subject. Although the transfer groups saved an average of 38 and 118 keystrokes on ED and EDT respectively across days, neither result yielded a significant main effect.

Although the difference in total keystrokes was not significant, analyses of variance using seconds per keystroke as the dependent measure showed that transfer subjects were keying at a higher rate than the practice subjects in both ED ($F = 10{\cdot}3$; $df = 1,6$; $P < 0{\cdot}05$) and EDT ($F = 13{\cdot}8$; $df = 1,6$; $P < 0{\cdot}01$). On average, transfer subjects were spending less than half the time (1·1 vs 2·3 seconds) per keystroke on day one than practice subjects.

TRANSFER FROM THE LINE EDITORS TO EMACS

Transfer curves

Figure 8 shows the transfer from the line editors to EMACS. Curves are plotted for control subjects who saw nothing but EMACS (emacs practice curve), control subjects who typed at the terminal for four days prior to using EMACS (transfer from typing),

experimental subjects who learned to use one line editor, and experimental subjects who learned to use two. Different orderings of the editors were collapsed in the two-editor condition. The most noticeable result is that, on the whole, those subjects who had four days of prior line editing experience showed substantial transfer on the first day of using EMACS (a savings of approximately 35 seconds per edit). One should recall, however, that this is on average less than half the amount of transfer observed between the line editors.

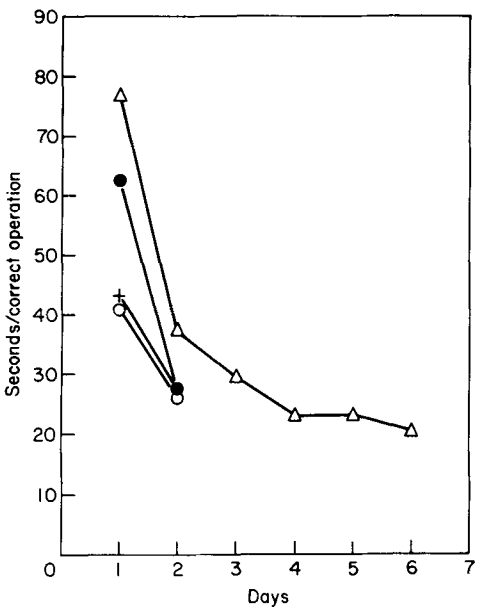

FIG. 8. Transfer to EMACS in terms of seconds per correct operation. Transfer curves are plotted to facilitate comparisons of performance on the first day of EMACS △ = emacs practice curve; ● = transfer from typing; + = transfer from one line editor; ○ = transfer from two line editors.

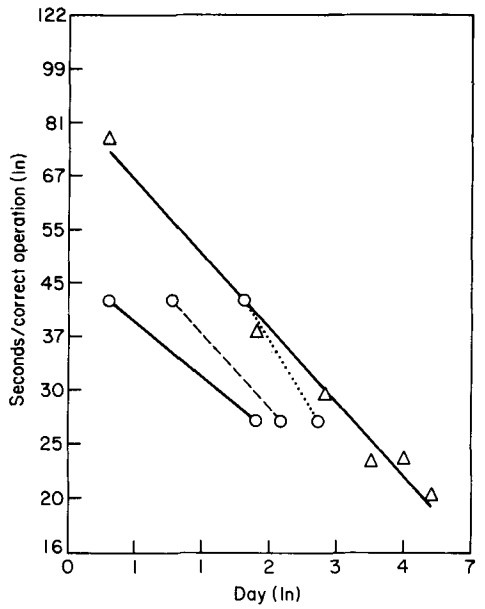

FIG. 9. A log-log transformation of the transfer data in Figure 8. △ = emacs practice curve; ○ = transfer from line editors; ○- - -○ = translated 0·9 days; ○— —○ = translated 0·3 days.

A two-way analysis of variance yielded a main effect for experience prior to transfer ($F = 5{\cdot}8$; $df = 3,20$; $P < 0{\cdot}01$). As expected, subsequent Newman-Keuls tests revealed significant differences between the group that had no prior experience (emacs practice curve) and the groups that had prior line editing experience (one line editor, $F = 5{\cdot}25$, $df = 20$, $P < 0{\cdot}01$; two line editors, $F = 5{\cdot}74$, $df = 20$, $P < 0{\cdot}01$). No other differences were significant, including the difference between the typing and EMACS control groups.

Examining the data in finer detail, one sees that the prediction concerning the advantage of two line editors over one is supported to only a very modest degree. Those who learned to use two line editors had an advantage of about 2 seconds per edit (12 seconds per screen) on both the first and second days. However, as mentioned above, this difference was not significant.

Figure 9 shows a rather unsuccessful attempt to apply the Rosenbloom & Newell analysis to this transfer data (the one-line-editor and two-line-editor curves have been

combined for this analysis). When the transfer curve is displaced 0·9 days, its first point lies on the practice function, but its slope is too steep. When the curve is displaced 0·3 days, the slopes (rates of learning) are equivalent but the first point does not lie on the practice curve (see Table 2 for the regression equations). As was observed to a lesser degree in the case of transfer from ED to EDT, it seems that a single index, namely the number of prior trials, is not sufficient to describe the pattern of results.

Number of keystrokes and keying rate

Figure 10 shows that, compared with the control groups who had had no prior line editing experience, the experimental groups used substantially fewer keystrokes per trial on the first day of EMACS (a difference of approximately 30 keystrokes). A two-way repeated measures analysis of variance produced a significant interaction between prior experience and days ($F = 3{\cdot}83$; $df = 3,20$; $P < 0{\cdot}05$). Newman-Keuls tests yielded significant differences for all four comparisons between the control groups and the experimental groups on day one (all $F > 3{\cdot}4$, $df = 20$, $P < 0{\cdot}05$). There were no significant differences between groups on the second day of transfer.

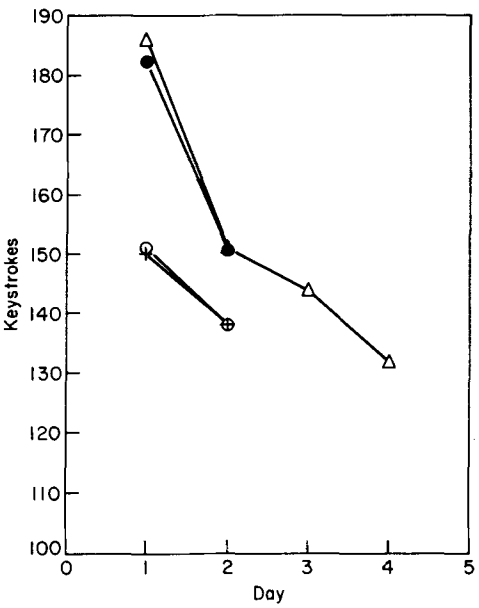

FIG. 10. Transfer to EMACS in terms of number of keystrokes per trial. △ = emacs practice curve; ● = transfer from typing; + = transfer from one line editor; ○ = transfer from two line editors.

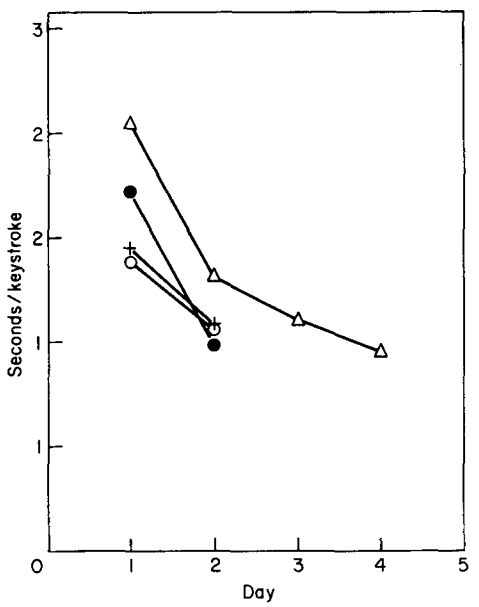

FIG. 11. Transfer to EMACS in terms of seconds per keystroke. △ = emacs practice curve; ● = transfer from typing; + = transfer from one line editor; ○ = transfer from two line editors.

In addition to fewer total keystrokes, the experimental subjects were keying at a higher rate than the control subjects (1·2 vs 1·5 keystrokes per second). Figure 11 presents the pattern of results. An analysis of variance of the keying rates yielded a significant interaction between experience prior to EMACS and day of transfer ($F = 4{\cdot}83$; $df = 3,20$; $P < 0{\cdot}01$). Subsequent Newman-Keuls tests revealed that, on the first

day of transfer, the two experimental groups were indeed keying significantly faster than the EMACS control group (both $F > 4{\cdot}7$, $df = 20$, $P < 0{\cdot}05$). As in the earlier keystroke anslysis, there were no significant differences on the second day of transfer.

Error data

Although errors are accounted for implicitly in our dependent timing measure, it is sometimes useful to look at error rates in pure form, apart from any timing data. Figure 12 shows the error rates for the four groups of subjects (controls, one prior editor,

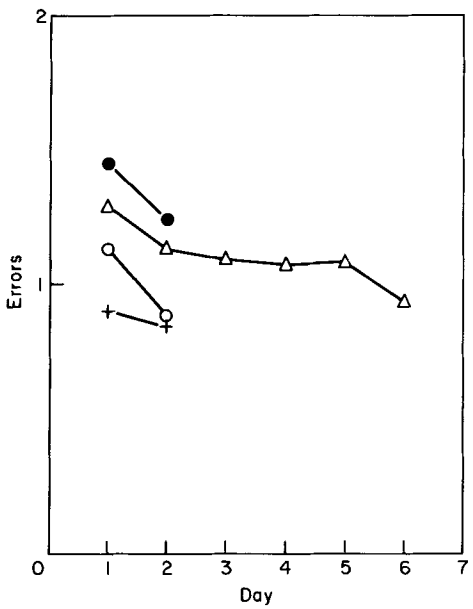

FIG. 12. Transfer to EMACS in terms of residual errors. △ = emacs control errors; + = one-editor errors; ○ = two-editor errors; ● = typing control errors.

two prior editors) on EMACS. Although an analysis of variance yielded no significant main effect for prior experience,the pattern of results is still rather intriguing. We see first that, overall, those subjects who had had four days of prior experience on line editors made fewer errors than both groups of control subjects. In fact, the error rate on the second day of EMACS for the experimental subjects approaches the sixth-day rate of the practice curve. (0·94 vs 0·92). However, what is most striking is the relatively high error rate on the first day of transfer for those subjects who had learned to use two line editors. Indeed, if times had not been adjusted for errors in the earlier analysis, the advantage of the two-editor group over the one-editor group would have been fifty percent larger (18 seconds per trial vs 12 seconds per trial). It is somewhat impressive that the two-editor group maintained a slight advantage in the face of a higher error rate.

Of additional interest is the high error rate for the typing control group, especially given the group's high keying rate (see Fig. 11). Perhaps the typing manipulation served to shift these subjects along the speed-accuracy function for text editing in EMACS.

Discussion

We have just begun to analyse the massive amount of data (approximately 500,000 keystrokes) collected in this experiment. At the moment, our results are expressed in terms of rather global measures. However, the basic outlines of the phenomenon of transfer are beginning to emerge. In this experiment, we observed:

(1) near total transfer among the two line editors;
(2) moderate amount of transfer from the line editors to the screen editor;
(3) slight transfer from typing to the screen editor.

Transfer manifested itself in a number of ways. These included:

(1) a reduction in total time;
(2) a reduction in total keystrokes;
(3) a reduction in residual errors;
(4) an increase in keying rate.

It is our view that the transfer of a complex skill such as text editing is not a unitary phenomenon, capable of being characterized as merely positive or negative. Complex skills are composed of many subcomponents that interact in complex ways and contribute individual and collective terms to the transfer equation. Although at present, we have no data on the differential transfer of specific commands or operators, the shapes of our transfer functions attest to the heterogeneity of transfer. It should be remembered that Rosenbloom & Newell's simple notion of prior trials was unable to account for our transfer data. This measure seemed particularly attractive at first, because it purported to describe what was most likely a complex phenomenon with a single index. However, although the concept of prior trials was adequate for representing transfer from EDT to ED, it could not represent transfer from ED to EDT or from the line editors to the screen editor. In the latter case, subjects who had spent four days learning to use line editors had the initial performance of control subjects who had spent approximately one day on EMACS but were learning like subjects who had spent less than half a day on EMACS. In short, the transfer subjects did not fit anywhere on the EMACS power function.

We propose a different analysis, that allows for the differential practice of a general component and a specific component when learning a skill. These general and specific components are not defined in absolute terms; they are defined relative to some transfer task. This analysis is based on the recent theoretical work on skill acquisition by Anderson (1982, 1983). A brief summary of that work follows.

OVERVIEW OF THE ACT THEORY OF SKILL ACQUISITION

Anderson's ACT theory of skill acquisition lies within a broader class of theories that use production systems to model human cognition (Newell & Simon, 1972; Thibadeau, Just, & Carpenter, 1982; Card *et al.*, 1983). In its most basic formulation, a production system consists of a set of condition-action rules called productions, and a working memory. A simple production is of the form:

IF the goal is to delete a character
  AND the editor is EMACS
  AND the character is marked by the cursor,
THEN hold down the control key and type 'd'.

A production such as this fires when the conditions of the IF clause of the production match the contents of the working memory. When more than one production matches on a particular cycle of the system, conflict resolution principles apply to select a single production for application. This enforces a strict seriality on the flow of control.

Anderson uses a self-modifying production system to model skill acquisition. The theory breaks down acquisition into two major stages: a *declarative stage*, where a declarative representation of the skill is interpreted by general productions and a *procedural stage*, where the skill is directly embodied in domain-specific productions. The transition from the declarative to the procedural stage is achieved by the process of knowledge compilation. Knowledge compilation consists of two separate mechanisms: the *composition* mechanism collapses sequences of general productions into single, highly specific productions, and the *proceduralization* mechanism deposits domain knowledge from long-term memory directly into productions. Taken separately, these compilation mechanisms can account for many of the phenomena associated with practice: elimination of piecemeal application of operators, dropout of verbal rehearsal, fewer working memory errors, and power-law speed-up (Anderson, 1982).

Once the transition from the declarative to the procedural stage is complete, additional learning mechanisms tune the compiled productions. *Generalization* makes productions more general by substituting variables for constants in production conditions or by deleting conditions altogether. *Discrimination* makes productions more specific by adding extra conditions or condition-action pairs. The addition of a condition-action pair amounts to adding both an IF and a THEN clause to a production. This makes both the test and the action more specific. One can see that generalization and discrimination are, in fact, inverse processes; both are used to control the range of application of productions. Along with a third tuning mechanism, *strengthening*, generalization and discrimination explain the continued improvement in performance following knowledge compilation.

Modelling transfer in ACT

To apply the ACT theory to the study of transfer, one first realizes that single productions are the units of cognitive skill, the "elements" that Thorndyke was searching for. A first approximation to an understanding of transfer involves comparing two sets of productions for different tasks. To the extent that the production sets overlap, transfer will be positive from one task to the other. This formulation is in fact a modern version of the Thorndyke and Woodworth Theory of Identical Elements. It is also similar to the recent proposal by Moran.

A TWO-COMPONENT MODEL OF TRANSFER

The calculation of transfer based on production set overlap naturally leads to a two-component model. The first component, the general component, is the intersection of two production sets, and the second component, the specific component, is the remainder of a particular set. The larger the set overlap, the larger the general component and the greater the amount of transfer. At high levels of transfer (e.g. from EDT to ED), the general component totally overshadows the specific component and transfer can be expressed fairly well in terms of a single parameter (prior trials). However, at middling levels of transfer (e.g. from the line editors to the screen editor), the specific component is too large to be ignored and transfer must be expressed in terms of two parameters, the relative contributions of the general and specific components.

TABLE 3
Power functions using the two-component model

Editor	Equation	$A_s/(A_s+A_g)$
ED	$T=9{\cdot}5+86G^{-0{\cdot}94}$	0·00
EDT	$T=5{\cdot}8+86G^{-0{\cdot}94}+29S^{-0{\cdot}94}$	0·25
EMACS	$T=15{\cdot}1+28G^{-0{\cdot}94}+12S^{-0{\cdot}94}$	0·30

Table 3 shows the results of an attempt to model our transfer data in terms of this two-component model. The extended form of the power law for this analysis is:

$$T=X_e+A_gG^{-b}+A_sS^{-b}$$

where X_e is asymptotic performance, G is practice on the general component, S is practice on the specific component, and A_g and A_s are the relative contributions of each component to the skill (A_g and A_s should sum to the value of A in the earlier formulation of the power law). Using this equation to predict performance on the first day of ED following two days of EDT, one would select the two-component function for ED and calculate the time using a value of 3 for G (third day of text editing) and 1 for S (first day of ED).

One should observe from the table that the specific component in ED is zero and is therefore proportionately smaller than the specific component in EDT (0 vs 25%). A value of zero in ED is probably a spurious result that merely implies that the true value is quite small. On the other hand, the relatively large specific component in EDT can probably be traced to the novel procedure for locating lines using strings, as described earlier. In all, the difference between the two specific components is substantial and in the right direction given the observed asymmetry in transfer between the two line editors. The set of productions shared by both EDT and ED (the general component) represents a larger proportion of the total set of productions shared by both EDT and ED (the general component) represents a larger proportion of the total set of productions in ED. In fact, this analysis implies that the ED production set is nearly a subset of the EDT production set. Therefore, transfer from EDT to ED is larger than transfer from ED to EDT.

Additional features of the data support this analysis. The fact that EDT was worse than ED on day one but improved more quickly (had a steeper slope), implies that the production set for EDT was larger than the set for ED. Mathematically, it follows that the shared productions constituted a smaller proportion of EDT than ED, since EDT was larger overall. In this way, the data lends some support to the simple model. It is interesting to note that the model could make predictions of asymmetric transfer based on the relative difficulties of two "overlapping" skills.

Although the specific component is not small in the EDT equation, it is even larger in the EMACS equation. As can be seen, the specific component amounts to 30% of the total production set for EMACS. This is what accounts for the steep slope of the EMACS transfer function, and also the inability of Rosenbloom & Newell's one-component model to fit the data.

SOURCES OF TRANSFER

What we have done so far has been a purely mathematical analysis of transfer, based on the behaviour of a few global measures. Ultimately, the primary purpose of such an analysis is to shed light on this fundamental question: What knowledge is transferred? Although a more informed answer to this question will be possible following a detailed analysis of the keystroke data, the question is so pressing that it demands at least a cursory treatment now.

High-level goal structure of text editing

One possible source of transfer is a knowledge of the basic goal structure of text editing. According to the GOMS formulation of Card *et al.* (1980*a*, 1983), text editing consists of a series of largely independent unit tasks, each of which are accomplished through the satisfaction of three subgoals: acquire the unit task from the manuscript, move to the line requiring modification, and modify the text. All editors share this high-level goal structure; some may share even larger chunks of the goal tree (e.g. the line editors used in this study). Presumably, this goal structure must be learned and could therefore be a source of positive transfer to a new editor. Indeed, Robertson (1984) has shown that text-editing novices differ from experts in their formulation of goals and plans. He argues that, to model novice behavior, the GOMS model must make certain allowances for deviations from the standard goal tree.

Conceptual mappings

In all likelihood, a major source of the massive positive transfer between the line editors was a common functionality. Although the surface features of the commands in the two editors were largely different, their underlying conceptual structures were nearly identical. This means that, in addition to the high-level nodes mentioned above, many of the intermediate levels of the goal tree were shared in the two editors. For example, to insert a line in ED, one moves to the line above the line to be inserted and types "a" for append. In EDT, one moves on the line below the line to be inserted and types "i" for insert. To exit the insert mode in ED, one types a period by itself on a line immediately followed by a carriage return. To exit from the insert mode in EDT, one presses "z" while holding down the control key. Although these methods are rather different, they have the same logical structure. This is the kind of similarity Moran (1983) intends to uncover using his ETIT technique.

This research was supported by the Personnel and Training Research Programs, Psychological Services Division, Office of Naval Research, under Contract No. N00014-84-K-0064. We thank Jeff Shrager for his help in programming the experiment and Jean McKendree and Peter Pirolli for their help in analysing the data.

References

ANDERSON, J. R. (1980). *Cognitive Psychology and its Implications.* San Francisco: W. H. Freeman.

ANDERSON, J. R. (1982). Acquisition of cognitive skill. *Psychological Review*, **4**, 369–406.

ANDERSON, J. R. (1983). *The Architecture of Cognition.* Cambridge: Harvard University Press.

BOTT, R. A. (1979). A study of complex learning: theory and methodologies. La Jolla, California: UCSD Center for Human Information Processing *Report* 7901.

BRUCE, R. W. (1933). Conditions of transfer of training. *Journal of Experimental Psychology*, **16**, 343–361.

CARBONELL, J. (1983). Learning by analogy: formulating and generalizing plans from past experience. In Michalski, R. S., Carbonell J., & Mitchell T., Eds, *Machine Learning: An Artificial Intelligence Approach.* Palo Alto, California: Tioga.

CARD, S. K., MORAN, T. P. & NEWELL, A. (1976). The manuscript editing task: a routine cognitive skill. Palo Alto, California: Xerox Palo Alto Research Center, *Technical Report SSL-76-8.*

CARD, S. K., MORAN, T. P. & NEWELL, A. (1980*a*). Computer text-editing: an information-processing analysis of a routine cognitive skill. *Cognitive Psychology,* **12**, 32–74.

CARD, S. K., MORAN, T. P. & NEWELL, A. (1980*b*). The keystroke-level model for user performance time with interactive systems. *Communications of the ACM,* **23**, 396–410.

CARD, S. K., MORAN, T. P. & NEWELL, A. (1983). *The Psychology of Human-Computer Interaction.* Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Digital Equipment Corporation (1982). Introduction to the EDT editor. *Manual AA-J824B-TE.*

DOUGLAS, S. A. (1983). Learning to text edit: semantics in procedural skill acquistion. *Ph.D. Thesis,* Stanford University.

DOUGLAS, S. A. & MORAN, T. P. (1983). Learning operator semantics by analogy. In *Proceedings of the National Conference on Artificial Intelligence.* Washington, D.C.

EGAN, D. E. & GOMEZ, L. M. (1982). Characteristics of people who can learn to use computer text editors: hints for future text editor design and training. In *Proceedings of the ASIS Meeting,* **19**, 75–79.

EKSTROM, R. B., FRENCH, J. W. & HARMAN, H. H. (1976). *Manual for Kit of Factor-Referenced Cognitive Tests.* Princeton: Educational Testing Service.

ELLIS, H. C. (1965). *The Transfer of Learning.* New York: MacMillan.

GICK, M. L. & HOLYOAK, K. J. (1983). Schema induction and analogical transfer. *Cognitive Psychology,* **15**, 1–38.

GOMEZ, L. M., EGAN, D. E., WHEELER, E., SHARMA, D., & GRUCHACZ, A. (1983). How interface design determines who has difficulty learning to use a text editor. In *Proceedings CHI'83 Human Factor in Computing Systems.* Boston.

GOSLING, J. (1981). *Unix EMACS user manual.* Pittsburgh: CMU Computer Science Department.

HALASZ, F. & MORAN, T. P. (1982). Analogy considered harmful. In *Proceedings of the Conference on Human Factors in Computer Systems.* Gaithersburg, Maryland.

HAYES, J. R. & SIMON, H. A. (1977). Psychological differences among problem isomorphs. In Castellan, N. J., Pisoni D. B. & Potts, G. R. eds, *Cognitive Theory,* Hillsdale, New Jersey: Lawrence Erlbaum Associates, **2**, 21–41.

KERNIGHAN, B. W. (1981). A tutorial introduction to the UNIX text editor. Murray Hill, N.J.: Bell Laboratories *Memor.*

MACK, R., LEWIS, C. H., & CARROLL, J. (1983). Learning to use word processors: problems and prospects. *ACM Transactions on Office Information Systems,* **3**, 254–271.

MORAN, T. P. (1983). Getting into a system: external-internal task mapping analysis. In *Proceedings CHI'83 Human Factors in Computing Systems.* Boston.

NAKATANI, L. H. (1983). Soft machines: a philosophy of user-computer interface design. In *Proceedings CHI'83 Human Factors in Computing Systems.* Boston.

NEVES, D. & ANDERSON, J. R. (1981). Knowledge compilation: mechanisms for the automatization of cognitive skills. In Anderson J. R., Ed., *Cognitive Skills and Their Acquisition.* Hillsdale, New Jersey: Lawrence Erlbaum Associates.

NEWELL, A. & SIMON, H. A. (1972). *Human Problem Solving.* Englewood Cliffs, New Jersey: Prentice-Hall.

NORMAN, D. A. (1983). Design principles for human-computer interfaces. In *CHI 83 Conference Proceedings.* Boston.

ROBERTS, T. L. (1979). *Evaluation of computer text editors.* Palo Alto: Xerox PARC *Report SSL-79-9*

ROBERTSON, S. P. (1984). Goal, plan, and outcome tracking in computer text-editing performance. New Haven, Connecticut: Yale University Cognitive Science Technical Report #25.

ROSENBLOOM, P. S. & NEWELL, A. (in preparation). Learning by chunking: a production-system model of practice. To be published in Klahr D., Langley P., & Neches R., Eds, *Self-Modifying Production System Models of Learning and Development.*

RUMELHART, D. E. & NORMAN, D. A. (1981). Analogical processes in learning. In Anderson J. R., Ed., *Cognitive Skills and Their Acquisition.* Hillsdale, New Jersey: Lawrence Erlbaum Associates.

THIBADEAU, R., JUST, M. A. & CARPENTER, P. A. (1982). A model of the time course and content of reading. *Cognitive Science*, **6**, 157-203.

THORNDYKE, E. L. & WOODWORTH, R. S. (1901). The influence of improvement in one mental function upon the efficiency of other functions. *Psychological Review*, **8**, 247-261.

WINSTON, P. H. (1979). Learning and reasoning by analogy. *Communications of the ACM*, **23**, 689-703.

YUM, K. W. (1931). An experimental test of the law of assimilation. *Journal of Experimental Psychology*, **14**, 68-82.