TRANSFER OF SKILLS BETWEEN INCONSISTENT EDITORS

Peter G. Polson and Elizabeth Muncher

University of Colorado, Boulder

David E. Kieras

University of Michigan, Ann Arbor

Tech Report # 87-10

TRANSFER OF SKILLS BETWEEN INCONSISTENT EDITORS

Peter G. Polson and Elizabeth Muncher

University of Colorado, Boulder

David E. Kieras

University of Michigan, Ann Arbor

1 Introduction

Our normal experience with computer systems is that increasing prior familiarity results in an increasing ability to learn a new computer system; such positive transfer effects can be very large (e.g., Polson, Bovair, and Kieras, 1987; Anderson, 1987). On the other hand, most users have probably experienced situations in which previous experience seems to conflict or interfere with learning and using a new system, due to inconsistencies between the old and new systems (Karat, Boyes, Weisgerber, and Schafer, 1986).

Current application programs on most systems have the kinds of inconsistencies that could potentially cause such interference. There are a limited number of functions that recur in a surprising number of different contexts. Examples are entering and editing a line of text in a manuscript, in a spread sheet, or as an operating system command. Different sequences of user actions must be employed to accomplish the same goals in each context. However, while such inconsistencies are common, do they cause negative transfer effects? Are negative effects of inconsistent methods strong enough to overwhelm positive transfer effects? This paper is concerned with the dynamics of transfer between two versions of the same application where the user has to learn inconsistent methods to accomplish the same goals.

1.1 Transfer and Interference

It is important to define some terms traditionally used in discussion of transfer effects, and to distinguish between observed transfer effects and the theoretical mechanisms that might underlie them. These terms and distinctions originated in the classical verbal learning literature in experimental psychology, in which transfer has been most systematically explored (Postman, 1971). As a phenomenon, *transfer* refers to how much learning to do a second task is either facilitated or impaired by having previously learned a different initial task. Assessing transfer necessarily involves comparison with a control group of some sort, such as one that continues to practice the first task, which produces the maximum performance, or one that learns the second task after learning an unrelated task, which would be expected to produce the lowest performance.

Positive transfer means that the second task is done better than the unrelated control, but normally not as well as the practice control. The phenomenon known as *negative* transfer must meet a tight criterion: performance on the second task must actually be *worse* than the unrelated control. In other words, the true negative transfer situation requires that the learner be worse off for learning the related prior task than for learning an unrelated one. In the classical verbal learning literature (Postman, 1971, pp. 1061- 1070), true negative transfer occurs only in a limited range of experimental conditions, materials parameters, and degrees of learning; there are no well documented examples in the human-computer interaction literature.

Finally, *interference* is a term used to characterize negative effects of previously learned material, such as reductions in learning rate, performance, and retention. Interference is also the label given to a set of possible theoretical mechanisms that explain such negative effects. *Unlearning*, *response competition*, and other specific mechanisms have been proposed to explain interference effects (Postman, 1971; Postman and Underwood, 1973).

Transfer between two inconsistent versions of a complex task will involve a complex mixture of interference and facilitation that determines overall transfer performance (Martin, 1965). Consider a user's representations of how to use two different text editors. Both representations share common knowledge of the overall organization of the text editing task and common low level methods, e.g. cursor positioning or formulating search strings. Two editors may have consistent or inconsistent representations of the structure of their respective editing commands depending on the similarity of the two editors. Finally, two very similar editors can share a common command structure but have different keys bound to same editing function. The consistent components will facilitate acquisition of the skills involved in how to use the second editor, and the inconsistent components could produce interference effects. Singley and

Anderson (1985) observed large positive transfer effects between very different editors suggesting that facilitory effects are more powerful that any interference produced by inconsistencies.

1.2 Transfer between Inconsistent Editors

The goal of the experiment reported in this paper is find out if inconsistencies in the commands of two text editors produce significant interference effects. The relationships between the inconsistent editors investigated in this study are analogous to relationships between learning tasks that produce reliable negative transfer effects in the classical verbal learning literature. In particular, two of the editors correspond to the strongest negative transfer situation in the classical literature; this is the "reversal" situation (the A-B, A-B_r paradigm) in which the same stimulus and response items that had to be associated in the first learning task are re-paired in the second learning task.

An *interference* interpretation of the transfer between inconsistent editors assumes that the representations of methods for operating the first version of an application interfere with the encoding or retention of methods for the second version. The amount of interference between two tasks depends on the similarity of their representations, with increasing similarity leading to increasing interference (Postman, 1971, pp. 1045-1070; Martin, 1965). In addition, it is possible that both representations could coexist, and response competition (Shiffrin and Schneider, 1977) can cause reductions in transfer performance and intrusions of old responses.

A *repair* interpretation assumes that knowledge of how to use the second version of an application is acquired by "repairing," or modifying, the representation of how to use the first version. The repair interpretation makes the basic assumption that the difficulty in transferring from one editor to another is a function of the magnitude and difficulty of the repairs. The more similar the two editors, the more similar are the representations, and the fewer and simpler the repairs. However, before the representations can be modified, the learner must detect the problem, and then determine the nature of the repair. Depending on the relative difficulty of these processes, a wide range of positive and negative transfer effects could be observed.

Although this paper deals with only the empirical effects in a transfer experiment, it is important to note that the production-system analysis described in Kieras and Polson (1985) and Polson, Bovair, and Kieras (1987) is consistent with a variety of possible interference mechanisms. Interference mechanisms could appear in production systems in the form of response competition effects; a production with a similar condition to the correct production could mistakenly fire, producing incorrect behavior. The probability of such interference could depend on details of the learning situation like those described in the classical interference theory of memory, such as degree of learning and order of learning. Effects of just this sort were observed by Kieras and Bovair (1986) in the pattern of errors in recall of simple procedures for operating a control panel device. However, repair mechanisms can also operate in production systems; in fact the model of positive transfer presented in Kieras and Bovair is essentially a very simple repair system, in that it examines the old and new representations for similarities between rules, and when necessary, modifies an old rule to work in the new context.

1.3 Overview of the Experiment

The objective of the experiment reported in this paper is to explore empirically the dynamics of transfer between inconsistent versions of the same application. This experiment attempts to distinguish between interference and repair interpretations of the processes involved in transferring between inconsistent methods.

In this experiment, the first version of the application is reasonably well learned, undergoing two days of intensive practice. Three experimental groups learned and practiced three different text editors and then were transferred to a common editor. The three editors varied in similarity to the common editor. In addition, there were two control groups. The first learned and practiced unrelated tasks before learning the common editor. The second learned and practiced the common editor for all five days of the experiment.

The procedures used in the experiment simplified the repair process. Subjects were told that they are being transferred to a new editor and then were given explicit instruction on the new methods. Thus, the processes of detecting inconsistent methods and determining the new method were drastically simplified, meaning that the only substantial repair process was that of actually making the modifications in the method representations. Thus, under these conditions, increasing the similarity of the two editors should result in fewer and simpler repairs, producing increasing positive transfer, just the opposite prediction from the interference interpretation.

2 Method

2.1 The Text Editors

Four different screen editors were used in this experiment. All of the editors had some common features. The cursor was moved using arrow keys; none of the editors had a find function or other cursor positioning functions. The first step in each edit operation was to move the cursor to the beginning of the edit, followed by issuing an editing command. Edits were terminated with an ACCEPT or UNDO key. The UNDO function reversed the effects of the last editing operation and repositioned the cursor at the end of the last accepted edit. All of the manuscript editing tasks performed in this study involved the 15 types of changes. The changes were defined by three different editing operations (delete, copy, and transpose) combined with five different objects (character, word, line, phrase, and sentence).

2.1.1 Single Keystroke Editors (SK1 and SK2)

These two editors has 15 commands defined by combination of the three actions and the five objects listed above. The editing commands are control-shifted single characters. Pairings of functions to characters was nonmnemonic for both editors. The second version of this editor used the same 15 letters repaired with new functions. The target location for COPY operations is specified by positioning the cursor at the target and then hitting the ENTER key. The beginning of the second range for TRANSPOSE is specified in an identical manner.

2.1.2 Cross Product Editor (CP)

This editor has the same basic structure as the SK editors, except the commands were mnemonic. Each command was generated by combining a single letter for the editing function (D, C, or T) with a letter for the object (C, W, L, P, or S). Thus all command were generated by the cross product of the two sets of letters.

2.1.3 Block Mode Editor (BM)

The block mode editor has three commands, each activated by a labeled function key (DELETE, COPY, and TRANSPOSE). The range of any editing function is involves selecting the block of text, indicated by highlighting, to be operated on. The range can be selected using the cursor keys or a single character find function. The select range operation is terminated by ENTER. The target location for COPY is selected using the cursor keys followed by an ENTER. TRANSPOSE involves the interchange of two selected ranges. The cursor keys are used to move to the beginning of the second range where the same general select range function is used to specify the second range.

2.2 Design

There were five training groups, two controls and three experimental groups. The nonspecific transfer control group (CNTRL-SK1) learned a series of word processor utility tasks before being transferred to the final common editor. The practice control group (SK1-SK1) learned and practiced the same editor both during original learning and transfer learning. The three experimental groups learned two different editors. The re-paired group (SK2-SK1) first learned one single keystroke editor and then was transferred to a second single keystroke editor in which all of the original 15 single keystroke commands had been reassigned to different editing functions in a second editor. This group of course corresponds to the classical verbal learning negative transfer condition, A-B, A-B_r. The cross product group (CP-SK1) first learned the cross product editor. The block mode group (BM-SK1) first learned the block mode editor.

When being trained on a text editor, subjects learned the three editing functions (delete, copy, move) in one of two different training orders. Training order 1 was delete, copy, transpose; training order 2 was transpose, delete, and copy. For each subject, the same training order was used with both editors.

Twenty-five subjects were run in each of the practice control group, the re-paired group, the cross-product group, and the block mode group; 15 subjects were run on training order 1, and 10 subjects on training order 2, in each group. Twenty-three subjects were run in the nonspecific transfer control condition.

2.3 Subjects

Subjects were recruited from the Boulder community and paid for their participation. They were required to have only minimal or no computer experience and to be able to type.

2.4 Apparatus

The experimental procedures were controlled by two programs that interacted with each other. The first program implemented the screen editors described previously. There were four such programs, one for each editor. The second program was a computer-assisted instruction (CAI) system that controlled training and practice procedures including presentation of instructional material and presentation of feedback on errors. There were two VDTs. The material to be edited appeared on one, and the subject inserted text and pressed command keys on its keyboard. The second VDT presented all instructions and feedback under the control of the CAI package.

The CAI package presented the instructional materials and then turned control over to the editor. When the subject completed an edit and depressed the ACCEPT key, the editor transmitted a description of the the edit to the CAI package where it was evaluated. If the edit was correct, there was no interruption in the procedure, and the subject was permitted to go on to the next edit. If an error was made, the CAI package would provide explicit feedback.

2.5 Procedure

There were two major phases of the experiment, original learning and transfer learning. Original learning ocurred on Days 1, 2, and 3; transfer learning took place on Days 4 and 5. The three experimental groups learned different editors during original learning, but all groups learned the SK1 editor during transfer learning. Three different procedures were used in this study: (1) the training procedure for text editing, (2) the practice procedure for text editing, and the (3) the procedure use for training and practicing word processing utility tasks.

2.5.1 Training Procedure for Text Editing

The training procedure was used on Days 1 and 4. There were *three* phases in the training procedure. In the *first* phase, all groups received a general description of the task of text editing, instruction on cursor positioning, and a description of the apparatus. Presentation of the instructional material was self-paced, and subjects could review previously presented frames.

The *second* phase involved learning three editing functions (delete, copy, and transpose). Subjects were trained on the editing functions one at a time. They first received detailed instruction on how to perform a function and then were drilled on the function. The drill involved editing a two page manuscript containing ten edits using the function. After making an error, the subject received feedback that described how to correctly do the edit, and then reviewed the instructions for the editing function, and then were returned to the drill. The criterion for learning was one error-free completion of the drill.

The *third* phase involved practice of all editing functions. Subjects were given a manuscript containing 30 edits, two each of the 15 different types of edits defined by all combinations of the three editing functions and the five objects. Subjects were required to redo incorrect edits. Feedback and brief instruction was given after errors.

2.5.2 Practice Procedure for Text Editing

The practice procedure was used on Days 2, 3, and 5. At the beginning of each practice session, subjects were given a brief review of the commands for the particular editor that they were practicing. They were then required to edit two ten-page manuscripts. Each manuscript contained 90 edits in six blocks of the 15 possible edits, whose order was randomized in in each block. Subjects took a short break between editing the two manuscripts. Specific feedback was provided after any error, and subjects were required to redo incorrect edits.

2.5.3 Word processor utility tasks.

The nonspecific transfer control group (CNTRL-SK1) learned and practiced various utility tasks for two different versions of a stand-alone, menu-based, word processor. Tasks included printing a document, deleting a document, copying a diskette, changing document format parameters, and other similar utility tasks. Each task involved traversing a series of menus to a terminal menu and entering parameters necessary to complete the task. On exit from the terminal menu, the system simulated completion of the task. The control tasks were intended to be of approximately equal difficulty to the text editing tasks, and to provide subjects with generalized experience of interacting with a computer terminal, and to give subjects experience with transferring from one version of an application to another.

2.5.3 Procedure for Original Learning.

Four of the groups (SK1-SK1, SK2-SK1, CP-SK1, and BM-SK1) learned a text editor using the training procedure on Day 1 and then practiced using that editor on Days 2 and 3 using the practice procedure.

Subjects in the CNTRL-SK1 were trained first on one version of the word processor and then a second version on Days 1 and 2 respectively. They learned eight utility tasks on each Day. On Day 3, they then practiced both sets of tasks acquired on Days 1 and 2 using the corresponding version of the word processor. Subjects were trained on each task using a serial anticipation procedure to a criterion of three errorless recitations of the task. The training procedure was identical to that used by Foltz, Davies, Polson, and Kieras (1987) and closely related to the procedures used by Polson, Muncher, and Englebeck (1986).

2.5.4 Procedure for Transfer Learning.

The procedure was identical for all groups during transfer learning. On Day 4 subjects learned editor SK1 using the text editor training procedure. On Day 5 all groups practiced with SK1 using the practice procedure.

2.5.5 Training on the Block Mode(BM) Editor

There were small but important differences in the training of subjects using the BM editor during original learning. These differences were due to the nature of the BM editor and the instructions they received concerning object selection. They were not explicitly told that there were different text objects, such as characters and words, and were not taught optimal methods for selecting each object on the BM editor. Rather, they learned a single method for selecting text objects by defining a range for the operation which involved moving the cursor to reverse-video the relevant text on the screen. The training procedure did not require that they learn an optimal method of selection for each type of text object. In contrast, subjects in SK1-SK1, SK2-SK1, CP-SK1 groups were explicitly told about the five types of objects and where forced to learn the optimal methods for each edit, and so learned 15 different editing methods defined by the combination of three actions and five objects. The subjects in the BM-SK1 group apparently had a different representation of the task at the end of original learning, compared to the other groups, having learned three editing methods defined by the combination of three editing functions and a single object selection method.

3 Results

The presentation of the results is organized into two major sections. The first section describes the results from the training procedures for those groups learning text editors on Days 1 and 4. The data from the nonspecific transfer control group (CNTRL-SK1) on Days 1 through 3 will not be discussed since they are not comparable to the data from conditions learning and using text editors. The second major section will discuss performance during the practice procedure on Day 5.

3.1 Training Days 1 and 4

The training days procedure involved three phases: general instructions, instruction and drill on three editing functions, and practice on a manuscript with 30 mixed edits. The data on general instructions will not be discussed, since these were identical for all groups.

3.1.1 Training Times

Training time was defined as the sum of the times to complete instruction and drill on the three editing functions. The mean training times plotted as a function of days are shown in Figure 1. Lines connect the four groups who were trained on a text editor on both days. The mean training time for the CNTRL-SK1 group on Day 4 is presented as an isolated point. The mean training times averaged over the four groups learning editors on both days was 64.03 min. on Day 1 and 31.63 min. on Day 4. The group means on Day 1 ranged from 51.78 min. to 81.33 min. with the BM-SK1 group being the slowest by almost 20 min. The Day 4 means for the four groups ranged from 22.34 min. to 41.38 min. An analysis of variance on the data for the four groups learning text editors on both Days 1 and 4 yielded significant F ratios for groups, $F(3, 92) = 7.89$, $p<.001$, $MSe = 428.45$, days, $F(1,92) = 299.85$, $p<.001$, $MSe = 177.21$, the groups by days interaction, $F(3,92) = 9.04$, $p<.001$, and the training order by days interaction, $F(1,92) = 5.12$, $p<.05$.

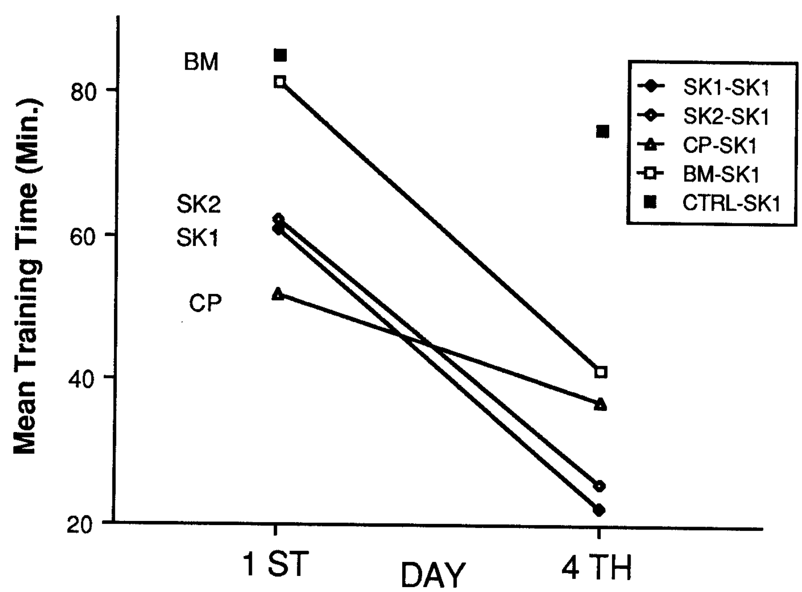

FIGURE 1. Mean training times plotted as a function of day.

Separate one-way analyses of variance were done for each of the two training days. The F-ratio for between groups on Day 1 was highly significant, $F(3, 96) = 8.72$, $p<.001$, $MSe = 442.37$. Post hoc comparisons showed that the block mode editor was learned significantly slower than the mean of the two single keystroke editors, $t(118) = 3.85$, $p<.01$. The differences between the average of the single keystroke editors and the cross product editor approached significance, $p < .10$. Thus on Day 1, the block editor was the most difficult to learn, the cross-product editor was the fastest, and the two single-keystroke editors were intermediate in difficulty.

As mentioned above, the difficulties experienced by the block mode subjects were in part due to the instructions they received and a peculiarity of the block mode editor. For example, there was no direct method for selecting a single line in the block mode editor. Subjects had to discover the correct method, placing the cursor two spaces beyond the last character on the line, by trial and error, and they often required experimenter intervention. Generally, the majority of subjects learned to use the cursor keys, an inefficient range selection method, to specify phrases, lines, and sentences. A large portion of the additional training time for the BM group occurred on the first editing function they learned. Training times for the second and third functions were much closer to the other groups. This is the pattern expected if subject were having trouble learning object selection methods, which once mastered, transfer to the other editing functions.

The differences between groups in transfer learning, as shown by training time on Day 4, were also highly significant, $F(4, 118) = 31.55$, $p < .001$, $MSe = 324.70$. Post hoc comparisons showed that the re-paired group (SK2-SK1) was not significantly different from the practice control (SK1-SK1), $t(118) = .68$, $p > .40$, that the group learning the cross product editor on Day 1 (CP-SK1) was significantly slower that the average of the practice and re-paired groups, $t(118) = 2.91$, $p < .01$, and that the group learning their first editor on Day 4 (CNTRL-SK1) was significantly slower that the average of the block mode and cross product editor groups, $t(118) = 6.43$, $p < .001$. The groups learning cross product and block mode editors on day 1 were not significantly different from each other, $t < 1.0$.

Thus, unlike what would be expected from the interference interpretation, the two single-keystroke groups had equivalent transfer learning time, even though the SK2-SK1 group corresponded to the classic maximum interference situation (Postman, 1971). The cross-product editor produced less positive transfer than the single-keystroke conditions, and the block mode condition produced even less. Finally, the group that did not learn a previous editor was by far the slowest during transfer learning. Apparently, the similarity between the original and transfer editors is associated with increased positive transfer, as predicted by the repair interpretation, rather than decreasing, as would be expected if interference was the dominating effect. It appears that in this experiment, there was only positive transfer, the amount of which was determined by the similarity of the two editors, and negative transfer was not present.

3.1.2 Original Learning Practice Times

The *practice time* was defined as the time required to complete the 30 mixed edits at the end of each training session. The mean practice times plotted as a function of days are shown in Figure 2. The lines connect the four groups who were trained on a text editor on both days. The mean practice time for the CNTRL-SK1 group is plotted as an isolated point on Day 4. The mean practice times averaged over the four groups learning editors on both days was 29.43 min. on Day 1 and 19.94 on Day 4. The group means on Day 1 ranged from 21.31 min. to 34.26 min. with the CP-SK1 group being the fastest by almost 10 min. The Day 4 means ranged from 13.51 min. to 25.07 min. with the SK1-SK1 group being faster by almost 7 min. An analysis of variance on the practice times gave a very similar pattern of results to the analysis for training times. The F-ratios for groups, $F(3, 92) = 8.64$, $MSe = 82.02$, days, $F(1,92) = 82.55$, $MSe = 49.92$, and the group by day interaction, $F(3, 92) = 12.23$ were all highly significant ($p>.001$).

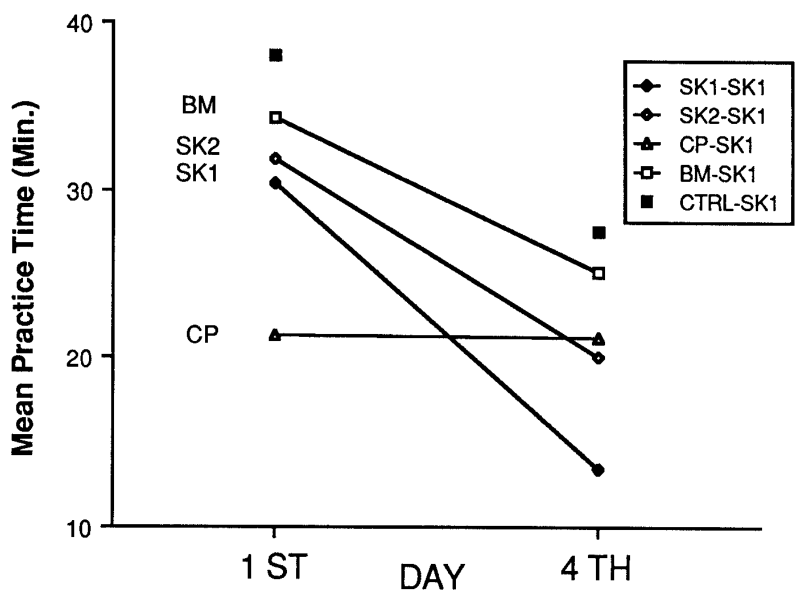

FIGURE 2. Practice time plotted as a function of days.

A one-way analysis of variance of the practice times for the four groups that learned text editors on Day 1 was highly significant, $F(3, 99) = 8.7$, $p<.001$, MSe=91.7. The Tukey B procedure was used to evaluate all possible pairwise differences between the groups (Winer, 1971). The cross product editor group (CP-SK1) was significantly faster than the remaining three groups who did not differ significantly from each other. Observe that in spite of the fact that the block mode editor was more difficult to learn, this group of subjects achieved a roughly equivalent level of performance to the two single-keystroke groups by the end of the first training day.

A one-way analysis of variance on the practice data for Day 4 was also highly significant ($F(4,118)=18.0$, $p<.01$, MSe=38.7). Using the Tukey B procedure, the practice control group (SK1-SK1) was significantly faster than the remaining four groups. The re-paired group (SK2-SK1) and the group learning the cross product editor on Day 1 did not differ significantly from each other, and the group learning the block mode editor and the nonspecific transfer control did not differ significantly from each other, and were both significantly slower than the group receiving the cross product editor.

Thus although the training times for the re-paired group and the practice control were similar (Figure 1) we see that the re-paired group does differ from the practice control in terms of performance time, being about the same as for the similar cross-product editor. There is still no evidence of negative transfer; the re-paired group is second only to the practice control in performance. The ordering of conditions is still that of improved performance with increasing similarity of the second editor to the first.

3.2 Practice Results for Day 5.

All subjects practiced on Day 5 using the SK-1 editor to edit two manuscripts that had 90 edits each.

3.2.1 Total Editing Times Figure 3 shows the mean times in minutes to complete each of the two manuscripts. One subject in the BM-SK1 group was dropped from the analysis because of excessive error rates on both manuscripts, greater than 50%. Data from two subjects in each of the SK1-SK1 and BM-SK1 groups and one subject in the CP-SK1 group were lost due to procedural errors. The mean editing times on the first manuscript ranged from 34.59 min. to 54.49 min. with an overall average of 44.35 min. The overall mean editing time for the second manuscript was 36.18 min with a range of 30.44 min. to 41.21 min. across the five groups. The fastest group, by about nine min. on the first manuscript and five min. on the second, was SK1-SK1. The slowest group was

BM-SK1 by nine min. on the first manuscript and six min. on the second. An analysis of variance showed that there was a highly significant difference between groups, $F(4, 106) = 9.91$, $p<.001$, MSe= 142.01. The difference as a function of practice between manuscripts, $F(1, 106) = 104.63$, MSe=35.20, $p < .001$ and the groups by manuscript interaction, $F(4, 106) = 3.47$, $p < .02$ were significant.

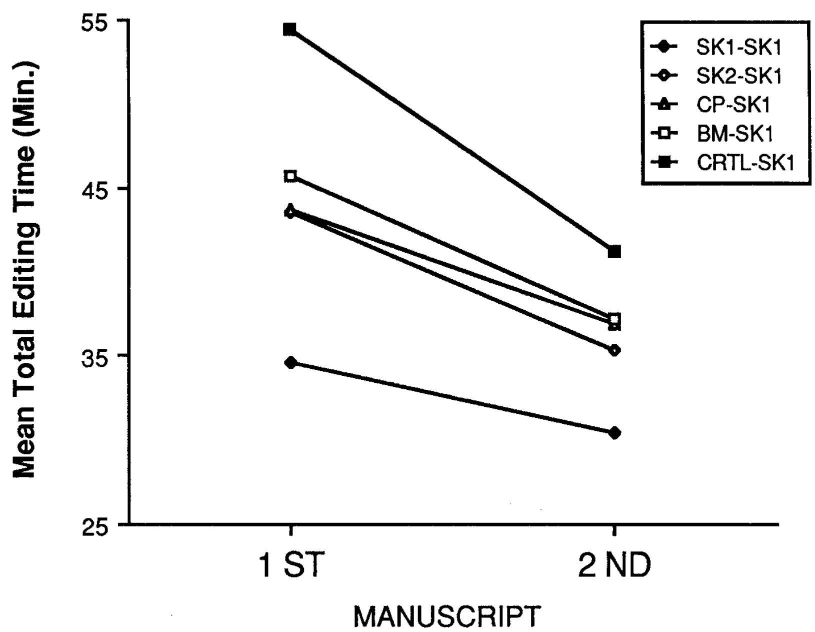

FIGURE 3. Mean Editing Times on Day 5 plotted as function of manuscript.

The one-way ANOVA's on editing times for both manuscript 1 , $F(4, 111) = 9.47$, MSe = 112.52, and manuscript 2, $F(4, 111) = 6.03$, MSe = 57.82 were highly significant, p's < .001. The patterns of post hoc test results with Tukey B procedure supports the impression given by figure 3. The SK2-SK1, CP-SK1, and BM-SK1 groups were not significantly different from each other for both manuscripts. The remaining comparisons were done using single degree of freedom contrasts. The SK1-SK1 group was significantly faster than the means of these three groups: manuscript 1, $t(111) = 3.68$, $p < .001$, and manuscript 2, $t(111) = 3.29$, $p < .005$. Tests between the mean of the SK2-SK1, CP-SK1, and the BM-SK1 groups the CNTRL-SK1 group found significant differences on both manuscript 1, $t(111) = 2.81$, $p < .01$, and manuscript 2, $t(111) = 2.20$, $p < .05$.

Clearly, on Day 5, the differences between the original learning groups have largely disappeared for those with equal practice on SK1, the second editor. The practice control has had 5 days of practice on the same editor, and so is far down on the learning curve. The nonspecific transfer control has had the same amount of practice on the second editor as the other transfer groups, but because this group did not enjoy the benefits of the large positive transfer from a first editor, they have not progressed as far on the learning curve.

3.2.2 Numbers of Errors

Figure 4 shows the mean number of errors per manuscript plotted as a function of manuscripts for the five separate groups. The overall mean errors on for the first and second manuscripts was 7.78 and 7.94 respectively. The SK1-SK1 group had the lowest error rates with approximately three fewer errors on both manuscripts. An analysis of variance showed that only the groups factor was significant, $F(4, 106) = 3.84$, $p<.01$, MSe= 49.26. The remaining F-ratios were very near or less than 1.0. Only the one-way ANOVA for manuscript 1 was significant, $F(4, 111) = 3.67$, MSe = 34.42, $p < .01$. Using the Tukey B procedure, no differences in mean errors were observed between the three groups learning different editors or the group learning its first editor on Day 4 for either manuscript. The SK1-SK1 group had significantly fewer errors than the other four groups on the first manuscript.

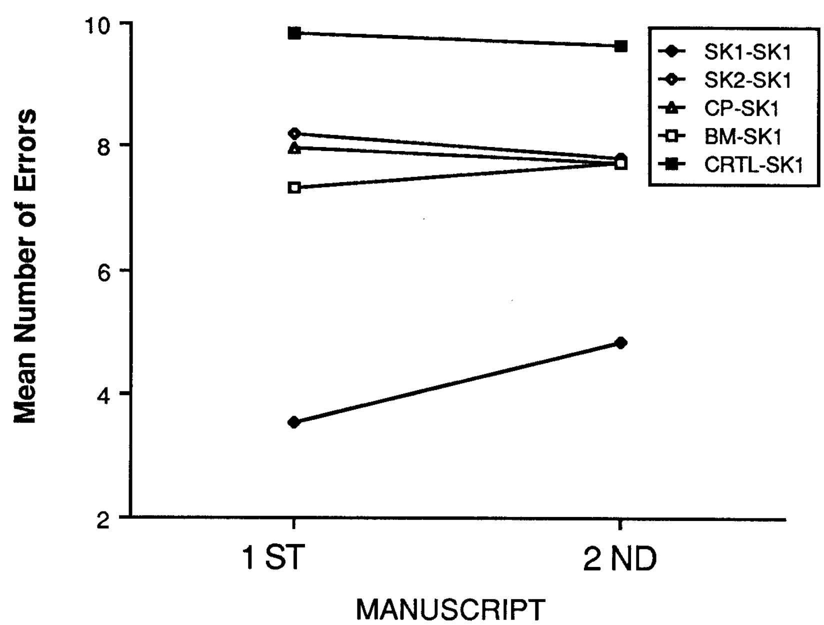

FIGURE 4. Mean number of errors made on day 5 while editing a manuscript plotted as a function of manuscript.

4 Discussion

4.1 The Power of Positive Transfer

It appears that even though these text editors differed fairly substantially in structure, the amount of positive transfer was very large, apparently large enough that any interference effects were swamped in the overall measures, and so did not produce any true negative transfer. Thus, subjects with previous text editing experience took a maximum of 20 additional minutes of training to learn a different editor when measured against a baseline of subjects in the practice control (SK1-SK1). The slowest group with previous text editing experience took about half again as long as subjects learning their first text editor. Thus, as observed in our earlier work, positive transfer for complex tasks like text editing is very powerful.

4.2 Repair verses Interference Interpretations

In dealing with the inconsistent methods, the repair position assumes that detecting inconsistencies and constructing repairs can lead to a wide range of transfer effects depending on the difficulties of various components of the repair process. This experiment was designed to simplify the repair processes in a way that permits a test of repair versus interference interpretations. Subjects were told that they were being transferred to a new editor and were given explicit instructions on the new editing methods, making it possible for them to make effective repairs quickly. Thus, the repair position predicts the least difficulty in transfer for subjects learning two similar editors and the greatest difficulty for subjects learning two dissimilar editors. The interference position makes the converse predictions.

On Day 4, the training times for learning an identical editor for the five groups were SK1-SK1 = SK2-SK1 < CP-SK1 < BM-SK1 < CNTRL-SK1. The most surprising thing was that the nonsignificant difference between the re-paired (SK2-SK1) and practice control (SK1-SK1) groups. This is the rank order predicted by repair interpretation, glossing over the lack of a difference between re-paired and practice control groups. Also note that all groups show large positive transfer effects if we consider either the BM-SK1 or CNTRL-SK1 groups as the appropriate nonspecific transfer control group. There was no evidence of interference at the level of these overall measures; even though the SK2-SK1 group executed the methods more slowly than SK1-SK1, they were no slower than the group transferred from the conceptually similar CP editor and the dissimilar BM editor. These results are strong support for the repair interpretation.

The practice time results on Day 4 and the Day 5 practice results are consistent with the repair position too. Polson and Kieras (1985) assumed that speed-up in editing performance is primarily due to composition (Anderson, 1987). The SK1-SK1 group had a compact, highly efficient representation of the skill by Day 5. The SK2-SK1, CP-SK1, and BM-SK1 had a mixture of well-practiced composed rules and novice rules. The CNTRL-SK1 group was using a novice rule set. As predicted, the editing performance is ordered SK1-SK1 < SK2-SK1 = CP-SK1 = BM-SK1 < CNTRL=SK1.

4.2 Generality of these Results

The conditions of this experiment were chosen to produce as much negative transfer as possible, based on the prior literature on human learning and memory (Postman, 1971). That no negative transfer appeared, and there was little evidence of substantial interference effects is surprising. However, Anderson (1987) reports an unpublished experiment done by Singley and Anderson (1986) comparing conditions similar to SK1-SK1 and SK2-SK1 and found no evidence of interference. Perhaps humans are able to learn new procedures with relatively little interference, an idea reminiscent of the classical "response selector" mechanism in verbal learning. If we suppose that text editing is normally done under cognitive control, rather than automatically (cf. Schneider, 1985), this relatively immunity to interference makes sense. Perhaps substantial interference effects only appear when the first or second editors have been practiced enough that the procedures are executed automatically at least some of the time. This might explain the anecdotal evidence of interference problems.

However, it could be argued that few aspects of human-computer interaction are executed often or intensively enough for interference to be a problem for more than a short time or for more than an occasional error. However, notice that these results do not address what might happen if users had to switch back and forth between editors, maintaining each at a high level of practice. This situation in computer usage is rapidly becoming more common.

Thus, we can conclude from these results that interference and negative transfer do not appear to play an important role in situations where moderate amounts of practice are involved. But, determining whether substantial interference effects appear after considerable practice under laboratory conditions, and whether they are important in realistic human- computer interaction situations, requires further research.

4.3 Implications for Software Design

It is tempting to conclude that the lack of negative transfer, and the relatively fast learning of the SK2-SK1 group, means that software designers should not be too concerned about maintaining consistency between related applications. But, notice that in spite of the lack of negative transfer, the amount of positive transfer is closely related to the consistency of the first editor with the second. Thus, for example, having learned SK2 or CP produced more positive transfer than having learned the BM editor, and considerable more than having learned a set of menu-based procedures.

Thus, these results argue strongly for the importance of maintaining consistency in the methods for related applications; the new result in this experiment is that such consistency is important not because of its role in preventing negative transfer, but because of its role in promoting positive transfer.

4.ACKNOWLEDGEMENTS

This research was supported by International Business Machines Corporation. The opinions and conclusions contained within this paper are those of the authors and not necessarily those of IBM.

5 REFERENCES

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

Anderson, J.R. (1987). Skill acquisition: Compilation of weak-method problem solutions. *Psychological Review*, 94, 192-210.

Foltz, P.W., Davies, S.E., Polson, P.G., & Kieras, D.E. (1987) Transfer between similar menu systems. A Poster presented at CHI'87 Human factors in computer systems. Toronto, Canada. April 7-9.

Karat, J., Boyes, L., Weisgerber, S., & Schafer, C. (1986). Transfer between word processing systems. In M. Mantei & P. Orbeton (Eds.), *Proceedings CHI'86 Human Factors in Computer Systems* (pp. 67-71). New York: Association for Computing Machinery.

Kieras, D.E. & Bovair, S. (1986). The acquisition of procedures from text: A production-system analysis of transfer of training. *Journal of Memory and Language*, 25, 507-524.

Kieras, D.E., & Polson, P.G. (1985). An approach to the formal analysis of user complexity. *International Journal of Man-Machine Studies*, 22, 365-394.

Martin, E. (1965). Transfer of verbal paired associates. *Psychological Review*, 72, 327-343.

Polson, P.G., Bovair, S. & Kieras, D.E.(1987). Transfer between text editors. In P. Tanner & J. Carroll (Eds.), *Proceedings CHI'87 Human factors in computer systems.* New York: Association for Computing Machinery.

Polson, P.G., & Kieras, D.E. (1985). A quantitative model of the learning and performance of text editing knowledge. *Proceedings of the CHI 1985 Conference on Human Factors in Computing.* San Francisco: ACM.

Polson, P.G., Muncher, E., & Engelbeck, G. (1986). A test of a common elements theory of transfer. In M. Mantei & P. Orbeton (Eds.), *Proceedings CHI'86 Human factors in computer systems* (pp.#78-83). New York: Association for Computing Machinery.

Postman, L. (1971) Transfer, interference, and forgetting. In L. King & L. A. Riggs (Eds.), *Experimental psychology* (pp. 1019-1132), New York: Holt, Rinehart, & Winston.

Postman, L. & Underwood, B.J. (1973). Critical issues in interference theory. *Memory and Cognition*, 1, 19-40.

Schneider, W. (1985) Training High-Performance Skills: Fallacies and Guidelines. *Human Factors*, 27, 285-300.

Shiffrin, R. & Schneider, W. (1977) Controlled and Automatic Human Information Processing II. *Psychological Review*, 84, 127-190.

Singley, K., & Anderson, J.(1985). Transfer of text-editing skills. *International Journal of Man-Machine Studies,* 22, 403-423.

Singley, K., & Anderson, J.(1986). A key-stroke analysis of learning and transfer in text-editing. Manuscript submitted for publication.

Winer, B.J. (1971). *Statistical principles in experimental design.* (2nd ed.). New York: McGraw-Hill.

Ziegler, J.E., Vossen, P.H., & Hoppe,H.U. (1986) *Assessment of learning behavior in direct manipulation dialogues:: Cognitive complexity analysis.* ESPRIT Project 385 HUFIT, Working Paper B.3.3. Fraunhofer-Institut fur Arbeitswirschaft und Oranisation, Stuttgart, West Germany.