# Learning Disabled Students' Difficulties in Learning to Use a Word Processor: Implications for Design

Charles A. MacArthur, Special Education
Ben Shneiderman, Computer Science
University of Maryland

## Abstract

Learning disabled students can derive great benefits from using word processors. The ability to produce a neat, printed copy can increase motivation and encourage writing for a wider audience. The editing power makes revision possible without tedious re-copying, thus freeing students and teachers to approach writing as a process involving repeated drafts. Specific problems with handwriting and spelling can also be circumvented. However, learning to use the editing capabilities often presents problems for students, especially those with learning difficulties. Word processors must be designed that are simple, easy to learn, and yet powerful. This study makes software design recommendations based on a study of learning disabled students learning to use word processing.

Two groups of four LD students (4th-6th grade) were given twelve hours of word processing instruction using two word processors. Detailed records of progess and errors were made during learning and a final assessment task. Specific design problems are reported and recommendations are made for tasks such as cursor movement, insertion/deletion, use of nulls, blanks, and formatting characters, and overall organization.

## Introduction

The use of word processing in schools promises to substantially change the learning and teaching of composition. Word processing has two key features that may affect the way students learn to write. First, the ability to produce a neat, printed copy affects students' perceptions of the quality of their writing and can increase motivation. It can also encourage writing for a wider and more varied audience. Second, the editing power makes frequent revision possible without tedious re-copying. This capability makes it feasible to teach writing as a process involving repeated drafts.

Although experienced writers revise their work frequently for content, organization, and audience impact as well as spelling and syntax, beginning writers revise infrequently and mostly for mechanical errors (Nold, 1981). A word processor frees students to concentrate on content first and revise for organization, style, and mechanics later. For learning disabled students, a word processor with a spelling checker offers the additional benefit of circumventing handwriting and spelling difficulties that make writing a frustrating experience.

Research on word processing in schools is limited, but it does provide evidence for some benefits of using word processors for writing instruction. In an exploratory study with 8th grade students, Kane (1983) found that students composed more text using a word processor than with pencil and paper. They also revised their writing more, both to change the organization and to modify individual words, phrases, and sentences. Dauite et al. (1983) reported that junior high students wrote more and made more frequent revisions using a word processor.

While word processing provides new possibilities for writing instruction, it also presents some new problems. Students must develop some minimal proficiency at typing and must learn to use the basic editing commands. Further research and experience are needed to see how great these problems are. Daiute et al. (1983) and Kane (1983) both found that most students were able to use the word processing commands after some practice and that weak typing skills did not block students from successful writing experiences although both were a source of some frustration. In a naturalistic study of computer use in special education by MacArthur (report in preparation), both regular and special education students from 5th through 8th grade were observed working with the Bank Street Writer. Most of the students had considerable difficulty using the editing features even for simple corrections. Although lack of typing skills did not seem to dampen student enthusiasm, students did type very slowly; typing would likely become a problem if word processing were used more frequently.

Some research is available on adults learning to use word processors (Carroll & Mack, 1982; Roberts, 1979; Shneiderman, et al., 1983), but few design-oriented studies have focused on children.

This study focused on the process of learning to use a word processor and the difficulties that learning disabled students have in acquiring and remembering the basic operations. It was an exploratory study with a small number of students designed primarily to generate hypotheses for future controlled experiments on word processor design. Although the research was conducted with learning disabled students, we conjecture that the results are relevant to average students and, perhaps, to adults, and that the study of learning disabled students emphasized weaknesses in software design. Studies are planned to extend the research to average students and larger groups of handicapped learners. A second purpose, not

emphasized in this paper, was initial exploration of the impact on students' writing and attitudes toward writing. The following questions guided the research:

* What difficulties do LD students have in learning to use a word processor?

* What features of a word processor make it easier or harder to learn?

* What features of instruction on word processing are important?

* How do students use the editing features to revise their work?

* What are students' affective responses to using a word processor?

## Procedures

Two groups of four learning disabled students were selected from the children attending a summer remedial reading clinic associated with the University of Maryland. Students selected were between the ages of 9 years, 6 months and 12 years, 2 months (completing 4th through 6th grades) and were of average intelligence. Available recent scores on the WISC-R, Peabody Picture Vocabulary Test, or the Detroit Tests of Learning Aptitude were used as measures of IQ. Except for 2 students with English as a second language who did not have valid scores, all scores were within one standard deviation of the mean. Reading ability, assessed with standardized tests and reported as grade levels, ranged from 2.2 to 5.0 which was from 1.9 to 3.2 years behind the average achievement for their school peers.

Each group met for 12 1-hour sessions, 4 times a week for 3 weeks. Each student worked at an Apple IIe computer; a printer was shared. One group used the Milliken Word Processor and the other used Cut and Paste. Both word processing packages are designed for ease of learning and simplicity of use. The Milliken system is targeted for school use while Cut and Paste is aimed at the home computer market.

All instruction was presented by one of the authors (CM). Instruction began with a brief overview of computer operation and the use of computers for word processing. In the first week, students were taught procedures for text entry, cursor movement, insertion and deletion of characters and words, cataloging, saving, and loading files, and simple printing. In the second week, insertion and deletion of spaces and returns for formatting and block deletion and block movement were introduced. Varying print formats were demonstrated but never thoroughly taught. All exercises and assignments were based on meaningful writing tasks. Students wrote an autobiographical paragraph and a list of names and phone numbers. Each student then wrote an article for a school newspaper which was printed and distributed as a culminating activity.

Three types of data were collected. First, the instructor kept detailed notes on planned and actual instruction. Second, a research assistant observed each student for part of each session and made detailed narrative notes. Both authors also kept notes on their observations of individual students. These notes focused on student errors, questions, and successful use of word processing features. The writing tasks that students worked on and student comments and affective responses were also noted. The authors and the research assistant met at least weekly to discuss the observations and to form additional specific questions to guide further observation and instruction.

The final data source was an assessment task on text editing developed and administered individually to all students at the end of the 12 sessions. Students were given a printed copy of a document with a number of corrections and additions marked in pencil. Changes included spelling, punctuation, capitalization, indentation, spacing of lines, and movement of one line. The document was a knock-knock joke similar in form to one used in instruction. Students loaded the file, made the changes, and saved and printed the corrected version. Detailed observation notes were taken.

## Results

Notes from the final editing task and the classroom observations were analyzed for errors, questions about operations, and successful use of word processing functions. Errors (and questions indicating confusion) were grouped into the functional categories and counted and organized by student and day. Table 1 charts the performance of students on the final editing task.

Typing conventions. The students made two types of errors apparently due to their lack of experience with typewriters. First, they made errors in spacing between words and sentences, either inserting extra spaces or leaving no space at all. These errors were apparent to the students and, after a few days experience, infrequent errors of this type were spontaneously corrected. A more persistent type of error was incorrect use of the shift key. Several students used the caps lock key to type a single capital letter. One student had difficulty learning to use the shift key because she tried to type shift and the letter simultaneously.

Cursor movement. Students had little difficulty moving the cursor. Neither word processor used an edit mode, so the arrow keys were always available for cursor movement. Inefficient cursor movement was quite common; students frequently used the left and right arrows to move through several lines of text rather than using the up and down arrows. All the students knew how to use the up and down arrows, but given the short length of student documents, the inefficiency was not costly and the behavior persisted.

Another inefficient use of the cursor points to some confusion about screen format. In moving from the beginning of one line to the end of the previous line, or vice versa, students often

moved the cursor up and over rather than back one space. This happened often when they were using the cursor with the repeat key and overshot the end of the line. In this case they seemed unaware that they could simply go back one character.

Insertion and deletion. Two persistent errors with insertion and deletion are especially interesting because of their design implications. First, when inserting a letter, word, or line in the middle their text, students frequently inserted spaces or blank lines first to make room for the inserted text. After inserting the text, they would go back and delete the extra spaces. When inserting an entire paragraph or a title or heading that should end with a return, this procedure makes some sense. One inserts a return to open a line, moves the cursor back one line, and types the title or paragraph; it is not necessary to delete the return. All of the students consistently followed this procedure, and we did not discourage them. However, when inserting a single character or word or a line within a paragraph, this method is clearly inefficient. Despite exercises on insertion, students continued to open spaces before inserting words and some students continued to do so before inserting single letters. Why? Initially, they had a strong intuitive sense that space was needed to insert new material; they expressed this belief verbally and left more or less space depending on the length of the insertion. It appears that this intuition was resistant to change; students lacked the confidence that the rest of the words would, in fact, move over to make room for their addition. Another possibility is that the movement of text as each character is inserted and the lack of space separating the word being typed from other words is visually distracting and confusing. This interpretation would make sense for students with poor typing skills who needed to look back and forth between the keyboard and screen, especially for remedial reading students with perceptual deficits. However, we have no direct evidence for this interpretation. Regardless of cause, the results suggest that alternatives to this insert method should be explored. One alternative that might be easier to learn because it is more in line with intuition is a strikeover system with an explicit insert command that opens up a blank line.

The second persistent error was in positioning the cursor for deletion, and to a lesser extent for insertion. The form of the cursor appears to make a difference in this type of error. The Milliken Word Processor uses a cursor shaped like a large capital letter "I" positioned between two letters. Cut and Paste uses a blinking rectangle positioned on a letter. In both cases, characters are deleted to the left of the cursor. Students using Cut and Paste exhibited a strong tendency to position the cursor ON the letter to delete rather than to the right of it. On the final assessment, 2 of 4 students using Cut and Paste and none of those using Milliken made this type of error.

One might expect the form of the cursor to affect errors in positioning the cursor for insertion as well. However, we rarely noted any errors in positioning the cursor to add letters or words (or the spaces to make room for letters or words). Students using both word processors did have difficulty locating the cursor to insert blank lines or lines of text they were moving. To insert a blank line students were taught to place the cursor at the beginning of a line and type return. Students were confused about whether the blank line would appear above or below the line where the cursor was located. This confusion is related to the more general confusion about returns, spaces, and screen format which is discussed in the next section.

Another error, or inefficient procedure, was observed that may be related to the form of the cursor. Two students in the Cut and Paste group used an unusual technique for replacing a single letter. The most efficient procedure, and the one taught, is to delete and then insert. These students inserted the correct letter first and then deleted the incorrect one. Possibly, the students were attempting to type over the error; the position of the cursor on the letter might encourage such an error.

One other inefficient procedure exemplifies the tendency to use the simplest procedure available rather than the most efficient. Students quickly learned to use the backspace delete key for immediate correction of typing errors. If they discovered an error two or three words back in the text, they tended to delete all the way back to it and retype those words rather than using the cursor to move back and fix the error.

Invisible characters and screen format. Another group of errors is related to confusion about the way text is formatted on the screen versus the way it is represented in the computer storage. Neither word processor was a "what you see is what you get" system (Shneiderman, 1983). Both systems represent text internally as a string of characters including returns and spaces. As text is displayed on the screen, it wraps around at the end of each line and begins a new line after a return. A similar re-formatting occurs during printing. This explanation is too abstract for elementary aged children to understand, and in fact is difficult for many adults. Children see a screen display with characters and empty space. Confusion arises because there are 3 types of "empty space." First, real spaces exist where they are typed. Second, returns are invisible (though in some word processors they are marked in some way). Finally, there are empty spaces where no character exists. Such "nulls" occur after returns, at the end of lines where words have wrapped around, and at the bottom of the file.

We observed three types of errors that reflect confusion about invisible characters and screen format. First, students initially tried to move the cursor into areas on the screen where no characters existed and were confused about why they couldn't do so. This problem receded as they learned to use spaces to indent and center words. Second, they typed a return at the end of a line rather than using the wraparound

feature or they used spaces to get to the next line instead of wraparound. Third, students used spaces rather than return to get to the next line after a paragraph. They also used an entire line of spaces rather than a single return to create a blank line. All these errors have one common feature. They look right on the computer screen, but they cause problems in re-formatting when text is inserted or deleted or the document is printed.

A closely related problem was difficulty in splitting and merging lines. Students did not understand that inserting a return would split a line or that deleting a return would merge two lines or paragraphs. This process was not explicitly taught, but the problem emerged when students accidently inserted or deleted a return and needed to recover from their error.

Although showing an explicit 'return' character on the screen is distracting, we conjecture that it is an appropriate design compromise.

Overall organization. The organization of Milliken and Cut and Paste differed considerably. Both are menu-driven systems. Milliken uses a desktop analogy. Users entering the system, see a graphic representation of a desktop with a four-choice menu: writing tools, file cabinet, typewriter, and help. Each branch leads to another menu. In order to save a document, the user, starting at the text area, must press 'Escape' twice to get back to the desk and select 'file cabinet'. We anticipated that students would have difficulty learning to find their way around the system and knowing when to go to each area. We also expected students to have problems leaving the text area and seeing their text disappear in order to go save it. However, students had little difficulty using the menus. The model of writing area, file cabinet, and typewriter seemed easy for them to grasp. Moving among the areas was relatively easy because they followed a simple rule -- press 'Escape' to get the desk and then choose one of the three areas. Some initial confusion was caused by the loading procedure; after loading a file, the user has to return to the desk and go to the writing area to see the text. Students expected to see the text right away. In addition, the file folder analogy led at least one student to believe that several 'papers' on one topic could be stored in the same file; he erased his first story by saving another one in the same file folder.

Cut and Paste uses highlighted menus at the bottom of the screen. On entering the system, the user sees the catalog of files on the disk. The Escape key is used to get to the menu, the arrow keys to highlight the desired option, and return to execute it. The menu changes as one moves from the catalog to the writing area and the print formats area. Students had far more difficulty moving around this system than the Milliken. Part of the difficulty was that since Cut and Paste was not designed for children, the vocabulary was unfamiliar, e.g., 'document' and 'catalog.' They also had difficulty manipulating the menus with escape, arrows, and return. The overall organization of the system was confusing as well, particularly in comparison to the clear division of the Milliken into writing, filing, and printing functions.

Saving and loading. Other than problems with the overall organization of these two particular word processors, the major problems with filing resulted from confusion between text in the computer and on the disk. Most problems occurred when updating a file created earlier, rather than in saving a new document. Students had trouble understanding that saving a new version under the same name would erase the old version. When told this, their first response was always to use a new name so they wouldn't lose anything. As mentioned above, one student using Milliken thought he could save more than one document in the same file folder. Another student, after making corrections to a document and being told to save the corrections, loaded the file rather than saving it. In general, students needed reminders to save their work and close supervision when doing so to avoid potential frustration. On the final editing task, all but one student loaded and saved the file correctly without help.

Moving and deleting blocks. Learning the procedures for moving and deleting blocks of text was relatively difficult for these students. Only three of the eight students were able to move a sentence on the final editing task, partly due to relatively brief instruction on the topic. Two general problems were encountered: remembering and following the steps of the procedure and positioning the cursor correctly for insertion.

Milliken uses a fully prompted approach to lead the user through each step; however, the students were reluctant to read the directions and forged ahead making mistakes. Cut and Paste uses a three step procedure of marking text, cutting to a buffer, and pasting. When the text was cut to the buffer, students thought they had lost it. They also had trouble remembering the steps.

Beyond the particular procedures of these two word processors, one general problem with moving text was evident; it is related to the problems mentioned above about opening a blank line before inserting text and positioning the cursor to insert a whole line. Once text had been marked to move, students did not know where to position the cursor to insert it. The fully prompted procedure used in Milliken did not allow them to open a blank line. Cut and Paste's two step procedure of cutting to a buffer and then pasting did allow students to open a line, which they did.

## Design Issues

A great many choices must be made in designing a word processor, and a lively debate can be found about each of those choices. This study provides preliminary evidence for some design decisions and suggests directions for further research.

Cursor form and position. At the level of text

entry and simple insertion and deletion, the designs of recent word processors, including the two used in this study, have begun to converge on a strategy for modeless editing. In this design, the system is always in insert mode, the arrow keys are used for cursor movement, and the delete key is used to delete the character to the left. Given this arrangement, the study suggests that a cursor located between letters is less confusing for deletion than one located on top of a letter. However, placement on a letter might be preferable if deletion were done directly to the letter or if the system used an overstrike mode.

Insertion. The standard insert operation, in which characters are inserted at the cursor position and remaining text is moved to the right, was avoided by the students in this study. They initially expected that this would type over their work. They persisted in opening space prior to inserting words and lines, despite contrary instruction. They may have lacked confidence in the operation because it ran counter to their intuition or they may have found the movement of text too confusing and distracting visually. An alternative worthy of investigation with reading handicapped children, at least, is a system that is normally in strikeover mode with an insert mode that opens up space for insertion.

Returns, spaces, and nulls. The most persistent difficulties of students stemmed from confusion about invisible characters and screen formatting. Students used whatever keys worked to make the text look right on the screen without being aware of the implications for future printing or insertion and deletion of text. They used returns or spaces to get to the next line rather than allowing the wraparound feature to work. They used spaces instead of returns at the end of paragraphs. They used spaces to create entire blank lines. They tried to move the cursor into 'null' areas.

A simple solution worth experimentation in word processors for handicapped children is to make returns and spaces visible in some way. Students could then see the difference between spaces, returns, and 'nulls' and see the effect of inserting and deleting returns more clearly.

Use of a 'what you see is what you get' system would solve part of the problem by eliminating differences between the screen and the printout. However, this would not deal with problems caused by extra spaces when material is inserted or deleted. However, rules could be developed for handling spaces at the right margin that would resolve this problem.

Another solution is to eliminate nulls altogether by designing a system in which the screen is filled with spaces and the user overstrikes the spaces with characters.

Overall organization. Simple, rapid, and reversible movement through the menu structure is important. The structure should be simple enough so that users can form or be shown a mental map. A more visual representation of saving and loading or elimination of the distinction between text on the computer and on the disk might aid novice users.

## Conclusion

The emphasis of this paper has been on students' errors in learning to use a word processor and the implications for design. However, it would not be proper to conclude without mentioning the impact on students' attitudes toward writing. We worked with eight learning disabled students with a history of failure at school tasks, especially writing. With one or two exceptions they wrote eagerly and continuously, though slowly. Informal review of their writing with paper and pencil and meetings with their teachers indicated that they wrote considerably longer pieces on the word processor. Students still did not spontaneously revise their writing just because the capability was there; however, they were willing to revise with adult guidance. Their excitement, especially about printing their work, was evident; they were obviously pleased with the prospect of printing a story with no errors. The newspaper they took home at the end was a solid achievement.

We believe that simplicity in software design can be obtained without sacrifice of power. The detailed account of errors has led us to many conjectures about design changes which we hope to validate in controlled experimentation. We are also interested in forming an explicit cognitive model of users which will have predictive and explanatory power.

## References


Carroll, J.M. & Mack, R.L. Learning to use a word processor: By doing, by thinking, and by knowing. IBM Thomas J. Watson Research Center, Yorktown Heights, N.Y., 1982.

Daiutte, C., O'Brien, P., Shield, A., Liff, S., Wright, P., Mazur, S., & Jawitz, W. The computer in the writing class: Problems and potentials. Paper presented at National Educational Computing Conference, Baltimore, Md., June 1983.

Kane, J.H. Computers for composing (Technical Report No. 21). New York: Bank Street College of Education, 1983.

Nold, E.W. Revising. In C.H. Frederiksen, N.S. and J.F. Dominic (Eds.), Writing: The nature, development, and teaching of written communication, (Vol. 2). Hillsdale, N.J.: Lawrence Erlbaum Associates, 1982.

Roberts, T. L. Evaluation of computer text editors. Palo Alto, California: Xerox, Palo Alto Research Center, 1980.

Shneiderman, B. Direct manipulation: A step beyond programming languages. IEEE Computer, August 1983 16 (8), 57-69.

Shneiderman, B., Hill, R., Jacob, R., & Mah, W. An empirical comparison of two Plato text editors. Journal of Computer Based Instruction. Summer 1983 10 (1-2), 43-50.



This research was supported by a grant from the U.S. Department of Education, Special Education Programs on the use of microcomputers with mildly handicapped students.