AD A123363

CMU-CS-82-122

Experimental Evaluation of Tools for Teaching the ZOG Frame Editor

18 May 1982

C. Kamila Robertson
Robert Akscyn
Computer Science Department
Carnegie-Mellon University
Pittsburgh, PA, USA

Accession For
NTIS GRA&I
DTIC TAB
Unannounced
Justification
form 50 per
By
Distribution/
Availability Codes
Dist | Avail and/or Special
A

DTIC COPY INSPECTED 2

This research was sponsored by the Defense Advanced Research Projects Agency (DOD), ARPA Order No. 3597, monitored by the Air Force Avionics Laboratory under Contract F33615-78-C-1551. It was also partially supported by the Office of Naval Research under Contract N00014-76-0874, and by a grant from the Palo Alto Research Center of Xerox Corporation. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency, the Office of Naval Research, Xerox Corporation, or the U. S. Government.

Table of Contents

1. INTRODUCTION 1
2. THE ZOG SYSTEM 2
3. EXPERIMENTAL DESIGN 3
4. METHODOLOGY 4
5. PROCEDURE 4
    5.1. Users 4
    5.2. The Task 4
    5.3. Data Collection 5
    5.4. Treatment of Data 5
    5.5. Expectations for User Behavior 5
6. RESULTS 6
7. DISCUSSION 8
8. CONCLUSIONS 9
9. ACKNOWLEDGEMENTS 10

List of Figures

Figure 2-1: A Self-describing ZOG Frame 2
Figure 3-1: Design of the ZED-Learning Experiment 3
Figure 6-1: Average Learning Curves 7
Figure 6-2: Comparision with other editors 9

List of Tables

Table 6-1: Learning Scores (Min ± CV) 6

Experimental Evaluation of Tools for Teaching the ZOG Frame Editor

C. Kamila Robertson and Robert Akscyn

18 May 1982

Abstract

This paper focuses on ZOG, a rapid-response, menu-selection, software system intended as a general-purpose interface to a computer. ZOG databases are networks of screen-sized displays called frames. ZOG's frame and net editor (ZED) combines coventional text-editing facilities with facilities specialized to the network character of the database. One of the design goals for ZOG is that ZOG be relatively self-contained in terms of instruction on the use of ZOG and ZED. This paper compares two ZOG-based tools for teaching naive users to edit with ZED: an on-line (net) users' manual and an off-line users' manual (derived from the on-line manual). They are compared first with each other, then with eight editors evaluated by Roberts and Moran [5]. The results indicate that (1) off-line and on-line manual users take about the same time to complete a standard instruction sequence, but (2) off-line users use ZED more effectively at the end of the sequence. Finally, (3) ZED learning falls in the middle of the range of Roberts and Moran's editors in terms of minutes required on average to learn to do a new editing task.

1. INTRODUCTION

In the past few years there has been a growing interest in evaluating human-computer interfaces, including interfaces to computer text editors. Several studies [1], [2] model users' interaction with an editor in terms of keystrokes and time required to acquire the next unit of text modification. Roberts [4] and Roberts and Moran [5] applied this model to compare time to learn a basic core of editor commands for eight editors--TECO, WYLBUR, NLS, WANG, BRAVOX, BRAVO, GYPSY, and EMACS.

ZOG, an interactive system developed at Carnegie-Mellon University [6], has a growing user community with growing needs. The ZOG project needs to find practical ways of responding rapidly to users' difficulties and improving the system generally. We also want to find methods of evaluating a system undergoing frequent design changes. In particular, we wish to evaluate ZOG's editor ZED, which combines facilities like those of other editors with facilities specialized to the hierarchical character of ZOG's databases. In a previous paper [3], we studied time for experts to complete a standard set of editing tasks using ZED. Roberts' editor evaluation scheme [4] offered the possiblity of relatively quick comparison of ZED with other editors.

In this study, we look at the behavior of beginners learning ZED, measured by time to learn a basic set of editing commands. This measure will be used to evaluate several tools for teaching ZED. We will continue to use Roberts' experimental scheme as a framework for comparison. Below, we first present a brief description of ZOG. Following this, we describe our experiment with beginners. Then we discuss the differences among the teaching tools. Finally, we discuss the results of comparing ZED with Roberts and Moran's editors.

2. THE ZOG SYSTEM

ZOG is a general purpose, rapid-response, menu-selection interface to a computer system. ZOG's databases are strongly hierarchical, multiply linked *nets* of displays called *frames*, each the size of a conventional standard (24 x 80) terminal display screen. Each frame (see Figure 2-1) consists of a set of *items*: a title, a few lines of text, a set of numbered (or lettered) menu items called *options* and *local pads*, and a line of ZOG commands called *global pads* at the bottom of the screen. Global pads include *back* (back up one frame) and *edit* (edit the current frame). An option, local pad, or global pad is selected by a single character, usually the first in its description.

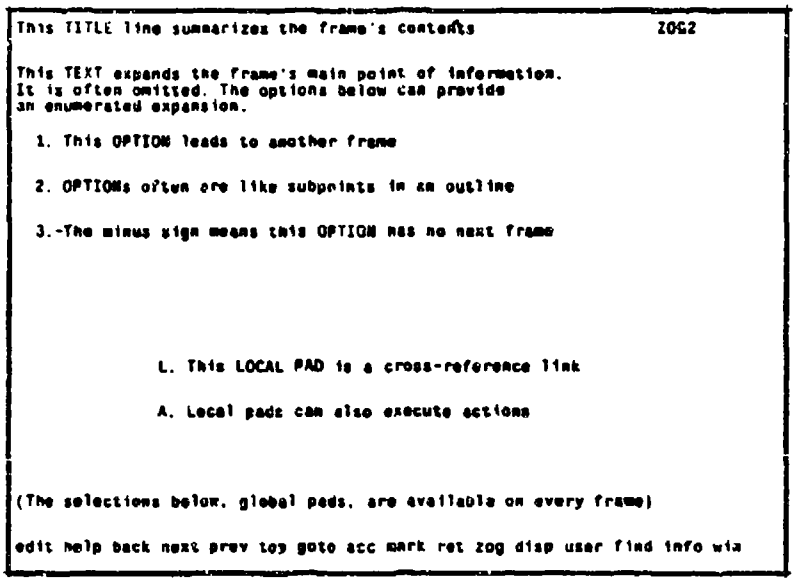

Figure 2-1: A Self-describing ZOG Frame

An option or local pad can point to a program and (or) another frame. Local pads usually point down subsidiary paths in the net. When the user makes a selection, the system executes the program or displays the appropriate next-frame. This structure allows rapid traversal of large amounts of information, with the system guiding the user in natural language. If the user selects an option or local pad with no next frame, ZOG will, at the user's option, create a new frame linked to that selection. ZOG then places the user at the new frame, in the editor (ZED). Thus a user creating a ZOG net moves freely between ZOG selection mode and ZED.

ZED is a display editor with commands for editing the textual content of the frame, rearranging the positions of items on the frame, and editing the non-displayed information such as next-frame links. Most ZED commands are single characters. After the user has selected the global pad *edit*, all keyboard input is interpreted as ZED commands rather than ZOG selections. Within ZED there are several modes: *command* mode, in which characters are interpreted as commands and command arguments, *insert* mode, in which characters are inserted into the text at the current cursor location,

position-item, and ZED *help*. The *exit* command returns the user to ZOG selection mode.

3. EXPERIMENTAL DESIGN

The question posed in this experiment is: for ZOG-naive users, how long will it take to learn a basic core of ZED commands with different teaching tools? Specifically, we consider: (1) human teacher, (2) on-line tutorial, (3) on-line manual, or (4) off-line manual. All three of the schemes without a human teacher exist in our environment and are under consideration as possible standard teaching techniques.

The teaching tool is the independent variable. Each teaching tool provides the same content. The tools differ chiefly in the way the user accesses them (by searching on-line or by page turning, for example), and in who controls the access (the user or a teacher). Average time to learn a new task (see Section 4 below) is the dependent variable.

The goals of the study as a whole are (1) to compare the teaching tools for speed of learning and ease of use and (2) to compare learning scores calculated by Roberts' method, for ZED and other editors. Calibration is provided by running one additional condition with a human teacher teaching the EMACS editor (replicating one of Roberts and Moran's conditions). We can then compare teacher-EMACS with human teacher-ZED, and teacher-ZED with other ZED teaching tools.

The complete design is shown in Figure 3-1. In this design, Part II corresponds to Roberts' method (described below). Part I, administered immediately before Part II, is an orientation to ZOG which does not involve editing. Part III is a test administered without any teaching tools, about a week after Parts I and II, to check for long term learning. Part III tasks were created to have the same structure as Roberts' exercises and quizzes. The user does the tasks of Part III and then must correct mistakes and omissions, so that a total time-to-completion can be recorded. This paper addresses the last two conditions in the table: comparison between on-line and off-line manuals. The results of the other conditions will be reported elsewhere.

PART I	Tutorial style introduction to ZOG net searching				
PART II	Emacs with teacher	ZED with teacher	Tutorial	On-line manual	Off-line manual
cycle1	oral lesson Quiz1	oral lesson Quiz1	tutorial section Quiz1	net search Quiz1	document search Quiz1
cycle2	lesson Quiz2	lesson Quiz2	lesson Quiz2	lesson Quiz2	lesson Quiz2
...					
cycle5	lesson Quiz5	lesson Quiz5	lesson Quiz5	lesson Quiz5	lesson Quiz5
PART III	Editing test:memo,autobiography,science fiction selections done to completion (all correct)				

Figure 3-1: Design of the ZED-Learning Experiment

4. METHODOLOGY

Roberts and Moran were interested in variations over editors, for a fixed teaching tool. In contrast, we are interested in variations in learning a single editor, due to teaching tool. However, Roberts' method has proved highly applicable to our goals. She developed a set of experiments including a test of time to learn a set of commonly used *core* commands, a score card for functionality, a test of expert performance time, and a score card for error and disaster potential. For this paper, we used her learning paradigm, which follows a set syllabus. The syllabus introduces a set of basic editing tasks with a sequence of exercises, each followed by a quiz. Exercises are optional; the user is to do as much and as many as he feels he needs to learn to use the editor. Quizzes are mandatory. A short summary of commands is available throughout.

Roberts developed a set of about 40 basic tasks. *Tasks* are defined functionally. Each consists of finding out what the next editing change is, locating the change in the on-line document, modifying the text, and verifying the change. Tasks include operations such as inserting, deleting, moving, splitting and merging. The text thus modified can be a character, word, line, sentence, or paragraph. The teaching sequence is composed of a set of five alternating exercises and quizzes. Each of these is composed of a set of editing tasks. The tasks are indicated by corrections, marked in red, which the user is to make.

During quizzes the user is to ask questions and use the summary only if absolutely necessary. Quizzes are scored cumulatively. The user receives one point for each task which was done correctly on a quiz (by whatever method), and done correctly on subsequent quizzes if there was opportunity. The principal data collected are: (1) total task time, and (2) time per task learned. Roberts provides a fixed set of quizzes and exercises to teach and test these tasks. The user is assigned a learning score, in minutes per task learned. Learning a command is defined as using it correctly at least once and thereafter using it correctly as opportunity arises during assigned tasks. The average of all users' scores is the score for the editor.

5. PROCEDURE

5.1. Users

Users were four beginners per condition. A beginner is defined as a college student or equivalent who has had at least one session on a terminal, but no more than one computing course or the equivalent. In this experiment, we found that most of the students who applied to be our users had some (less than one year) experience with EMACS, a display oriented editor in extensive use at Carnegie-Mellon. EMACS has a set of commands which is very different from ZED. Thus our users had had some editing experience, but with a set of commands which would not transfer directly to ZED use.

5.2. The Task

Roberts' documents were mapped onto ZOG frames, with approximately 10 to 12 lines of text per frame. Frames in the exercises and quizzes were linked linearly (that is, with a minimum of hierarchical structure) to minimize ZOG searching. The core of editing tasks in ZED was defined so that editing was done within a fixed net structure. Tasks included moving text between frames using the move/copy facility, but not changing the basic net structure. Most ZED editing in fact occurs within frames, and the editors with which ZED was being compared contain nothing comparable to net building. This task is realistic for ZOG use and is similar to the ongoing training situation of people learning to use ZOG/ZED at present.

Roberts' syllabus had to be adapted to work with all of our teaching tools. For the conditions using manuals, the chapter on editing plus the entire table of contents were available. The user could look up something specific or just read the manual. (The complete off-line manual is *The ZOG User's Guide* [7]). The off-line manual was presented in a three-ring notebook. The on-line manual was contained in a ZOG net and accessed by a local pad "M. Manual") from every introduction, exercise, and quiz frame. The on-line manual consisted of the same text as the off-line manual, one concept/command usually corresponding to one frame. The user searched the manual net and then used the *return* global pad to return directly to the frame from which he started, outside the manual. The human teacher followed Roberts' syllabus as closely as possible. However, in all conditions search by content was learned early, although in Roberts' syllabus this comes at the end. ZED editing depends heavily on the user's ability to search by content.

On-line versus off-line conditions were counterbalanced over morning and evening experiment sessions. A copy of the document net was created for each user to modify. One user at a time sat at a PERQ (personal computer) display simulating a Concept terminal, with a 9600 baud hardwired line to a DEC Vax 11/780 computer. ZOG was already invoked, and the appropriate teaching tool was ready. One user was given a single teaching tool.

The rule for questions and use of the summary was as in Roberts and Moran's method. In addition, during quizzes, the user was to limit his use of the teaching tool (e.g., the manual) to occasions when he was unable to continue otherwise. ZED help frames could be used at any time.

5.3. Data Collection

Each user was videotaped. A copy of the screen display the user was reading was superimposed on the television picture, along with a millisecond timestamp. Videotape data were accurate to one thirtieth of a second (the frequency of the video frames). During the session, ZOG unobtrusively recorded the user's path through the net and the selections and editing commands at each frame, each timestamped, on a log file. These data were pooled to identify errors and to partition time among reading the manual, reading the summary of commands, reading ZED help, using ZED commands, and taking breaks. Ultimately, the teaching tools are to be characterized and compared for the relative type and amount of use of the teaching tool, and for type and number of user errors. Quiz scores were obtained by comparing hardcopy of the edited frames with the quiz documents.

5.4. Treatment of Data

Since the material presented in Part I is essentially orientation to the system and does not bear directly on editing, Part I data did not enter into the comparison of editing results. For Part II, total editing time was determined. Editing time includes time learning editing commands and time making the corrections to the documents, and excludes breaks and major system delays that were unrelated to editing. Part II total editing time was divided by the cumulative quiz score, to obtain a learning score. For Part III, the total time to completion was observed. Significant nonediting delays were removed from these figures.

5.5. Expectations for User Behavior

We anticipated that the manual users would search for the commands they needed as the need arose, but that the on-line manual users might take longer. They must leave the visual context of the frame to be edited to use the manual; the off-line manual users could maintain the context on the screen while searching. Tutorial users and human teacher users had less choice in the way they "accessed"

information. Tutorial users should take longest of all because of the many directed tasks they must perform along the way through the instruction. Human teacher users have the most direct access to information (by asking questions) and so might be the fastest overall.

6. RESULTS

The EMACS condition indicates whether our experiment results were comparable with Roberts and Moran's. A major difference in learning score between our EMACS users' average learning score and that of their EMACS users would tell us to be cautious. Our first two users averaged 5.9 minutes per task: theirs averaged 6.6. (Both sets of EMACS users are represented in Table 6-1 and Figure 6-2, which will be discussed below.) We will confirm this result with further EMACS users of our own. However, these averages are close enough to indicate that our experiment is generally comparable with theirs and that we can place our on- and off-line users in Roberts' and Moran's continuum of editors. We will proceed first with comparison between on- and off-line, and then with comparison of ZED conditions with the other editors.

All of our learning score results of the on-line and off-line manual users are shown in Table 6-1. The left column contains scores for off-line and on-line teaching tools for ZED; the right column contains learning scores for Roberts and Moran's editors.

ZED Learning Scores		Roberts and Moran's Learning Scores	
		TECO	19.5 ± .29
		WYLBUR	8.2 ± .24
		NLS	7.7 ± .26
On-line	7.1 ± .20		
		BRAVO	7.3 ± .14
Off-line	6.7 ± .25		
		EMACS	6.6 ± .22
		WANG	6.2 ± .45
Our EMACS	5.9 ± .19		
		BRAVOX	5.4 ± .08
		GYPSY	4.3 ± .26

Table 6-1: Learning Scores (Min ± CV)

Learning curves for our users and for Roberts and Moran's users are represented in Figure 6-1, in the same format as Roberts' [4] Figure 4.1. Our Figure 6-1 also contains plots of Roberts' data for her worst editor in terms of time to learn, TECO/second teacher, and for her best editor, WANG. The sloping segments of each curve represent time spent in instruction and exercises. The horizontal segments represent quiz time. This format represents the user's knowledge as increasing during non-quiz time and remaining constant during quizzes, but realistically, some learning does occur during quizzes.

The average for off-line learning was 6.7 minutes per task, and for on-line, 7.1 minutes per task. Overall learning time was 149 minutes for off-line, and 158 for on-line manual. A t-test indicates that on-line and off-line manual users do not differ significantly in minutes per task (Roberts' learning score). The graph confirms this.

A significant difference between on-line and off-line users did result from the retained learning test in Part III of our experiment. To do Part III, off-line users averaged 1722.82 seconds, and on-line users, 2585.72 seconds. A t-test shows that off-line guide users' time to completion was significantly longer

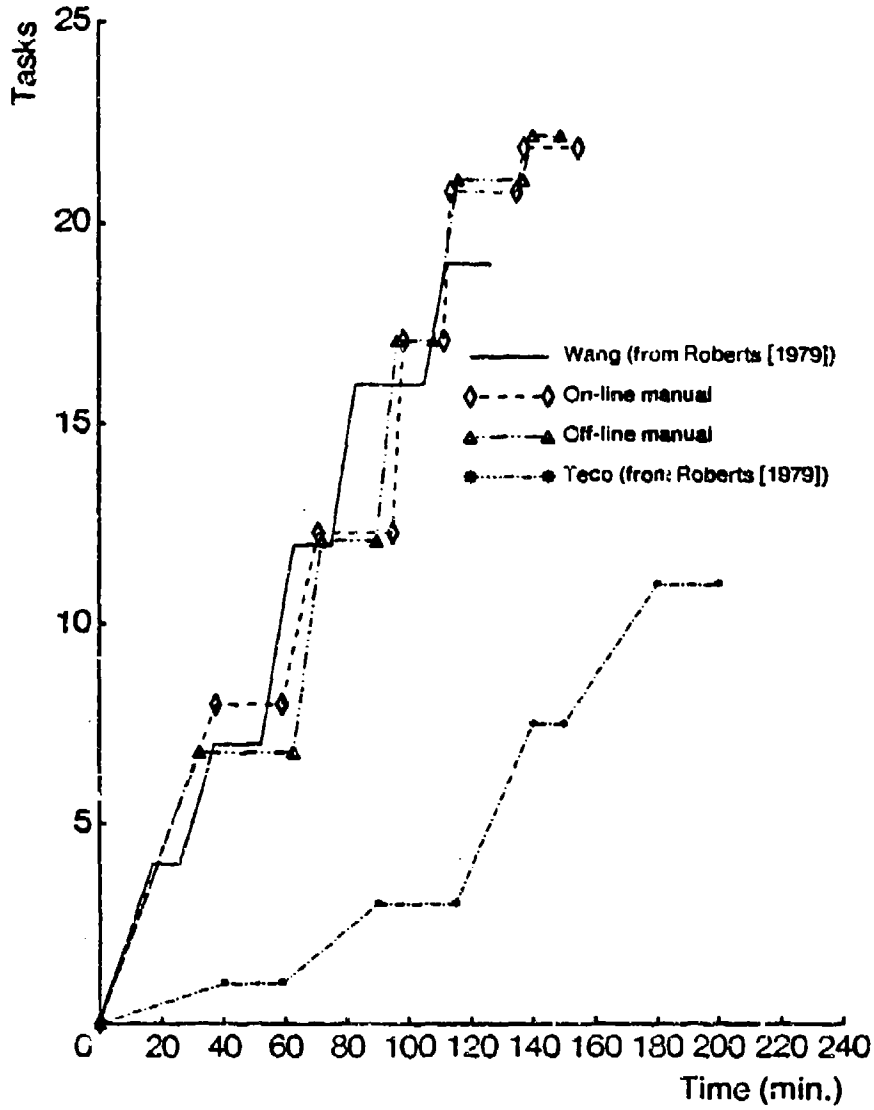

Figure 6-1: Average Learning Curves

than on-line users' ($\alpha = .05$). Total time to completion is composed of initial time (up until the user first said he was finished), and correction time if any (time to correct mistakes and omissions discovered by the experimenter). The off-line average was 1568.41 seconds; on-line was 2131.76. Here, a t-test showed that off-line guide users had significantly less initial time ($\alpha = .025$). Correction time differed in the same direction but was significant only at the $\alpha = .25$ level, since there was a large variance in correction time.

It is useful to express the effectiveness of the teaching tool in terms of the time per task completed. For the retained learning test, off-line users took an average of 78.3 seconds per task; on-line users averaged 117.5 seconds (which, like the totals above, is significant at $\alpha = .05$). These can be

compared with Quiz 5, the point at which the users had gone through the entire instruction sequence, before the one-week wait for Part III. In Quiz 5, the off-line average is 86.7 seconds, and the on-line average, 202.6. This difference again is significant at $\alpha = .05$. These users are of course novices. For comparison, expert ZED users take about 30 seconds per task [3]. Similarly, Roberts and Moran's expert users took about 49 seconds per task for TECO (the longest), 37 for EMACS, and 19 for GYPSY (the shortest).

Comparing Quiz 5 with Part III minutes per task, t-tests show that: (1) for off-line, this difference is not significant; (2) for on-line, it approaches significance, at $\alpha = .15$. Nevertheless, given that the expectation is a decline in performance after a week, this trend toward improvement is noteworthy.

Figure 6-2 shows our off- and on-line users' learning scores along with those of Roberts and Moran's Figure 4 in [5]. Our two EMACS users have been represented in the same column with Roberts and Moran's EMACS users, to show the degree to which our two EMACS conditions had similar results. Off-line and On-line ZED users appear in the middle of the range of Roberts and Moran's editors.

Roberts [4] gives data for individual users for four of the editors, so we can compare them with our users statistically. (Only graphical data was available in [5] for the other four.) Both on-line and off-line ZED groups had significantly better learning scores than Roberts' faster set of TECO users, who learned from Roberts' second teacher ($\alpha = .005$ for both). T-tests comparing the two ZED groups with Roberts' other editors (WYLBUR, NLS, and WANG) do not show significant results. Roberts' tests indicate that all her TECO users had significantly higher (worse) learning scores than users of her three other editors, and there were no significant differences among the three editors. Our results place off-line and on-line ZED users in the faster of her two overall groups.

7. DISCUSSION

The videotapes show that contrary to our expectation, both on-line and off-line manual users searched the manual almost entirely at the beginning of the session. Use of both manuals fell off rapidly as the user gained experience editing. Off-line manual users apparently did not derive any benefit from the ability to keep their editing context on the screen since most of their "book learning" occurred before most of the editing tasks were attempted. Since both groups used the manuals in this fashion, it is not surprising that there is little difference in time to learn between the two conditions. We do not view this as a failure in design in showing differences among the teaching tools. Rather, we view this result as showing that potential differences among teaching tools are not necessarily used by real users in a realistic situation.

The difference we observed between on- and off-line guide users in the Part III test of retained learning may indicate that the on-line users did not spend as much time, or as much attention, in reading their instructions. Studies of the relative use of guide (total time reading the guide, and relative use of guide versus summary versus ZED help) are underway.

With respect to confusion resulting from use of other editor, our users reported that some experience with interactive editing helped with learning ZED but that knowing EMACS tended to be a source of confusion in learning ZED commands at first. Even so, the users were able to make use of the manuals to find whatever information they needed. The users asked almost no questions of the experimenter though they had the option to do so.

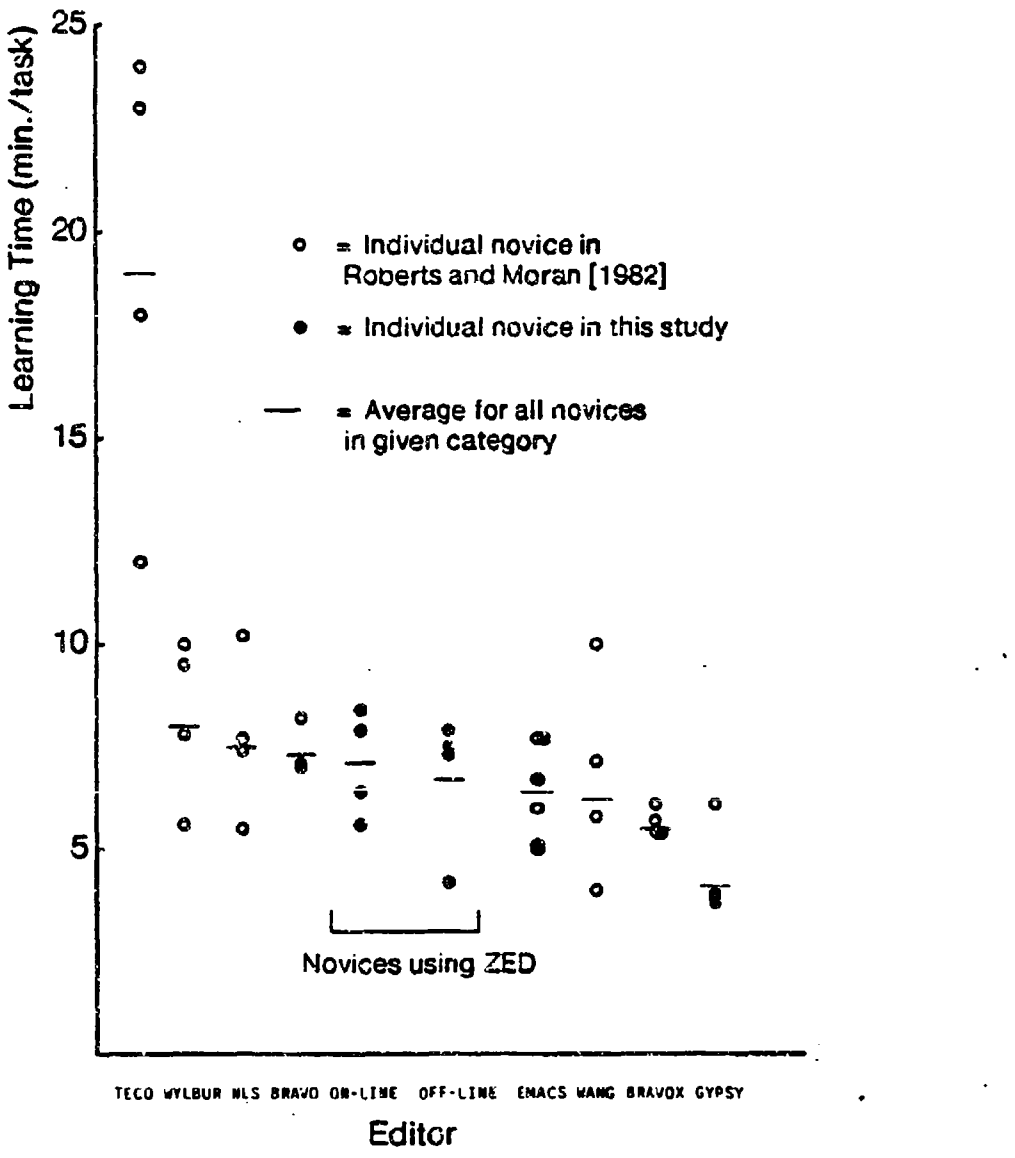

Figure 6-2: Comparision with other editors

8. CONCLUSIONS

First, Roberts' methodology has proved highly effective in evaluating ZOG, as it did in our study of expert performance [3]. It provides a way of approaching the attributes of the system and also a means of comparing with other, dissimilar, editors. Second, we have found that the on-line and off-line manuals, though different in appearance, are not necessarily used differently. Our expectations about the benefit of preserving editing context on the screen were not upheld because of users' style of accessing the information. However, the off-line guide did result in more effective performance, as indicated by the retained learning test in Part III.. This difference is indicated as early as the point when the user takes Quiz 5.

Both teaching tools were effective overall, however, since performance was maintained from Quiz 5 to Part III for both tools. In general, the ease of learning ZED falls midway along Roberts and Moran's continuum of editors.

In the future, we plan to study the way users partition their time reading the screen and the various documents available to them, and their searching versus modification behavior in the editor. Also, we anticipate having a version of ZOG which allows the user to view two frames on one display screen. This would permit searching an on-line manual in one window while editing a frame in the other. It will be interesting to see whether in this situation the user takes advantage of the ability to preserve context. We plan to evaluate this system as soon as it is available. These results will be added to those of our tutorial and teacher conditions. The methodology used here will allow us to locate each with respect to other teaching tools and other editors.

9. ACKNOWLEDGEMENTS

The authors wish to thank Allen Newell for his ideas and his help in preparing this report, Sandra Esch for running experiment sessions and for data analysis, and Elise Yoder for her work on both versions of the ZED manual and for general assistance in the experiment. We also thank Patty Stengel for help in preparing the figures and the final version of the paper, and Sally McDowell for preparing data files.

References

[1] Card, S., T. P. Moran, A. Newell.
The Psychology of Human-Computer Interaction.
Lawrence Erlbaum Associates, Hillsdale, N.J., in press.

[2] Card, S., T. Moran, and A. Newell.
The keystroke-level model for user performance time with interactive systems.
Communications of the ACM 7:396-410, 1980.

[3] Robertson, C. Kamila, D. L. McCracken, and A. Newell.
Experimental Evaluaton of the ZOG Frame Editor.
In *Proceedings of the Annual Conference of the Canadian Man-Computer Comunications Society*. 1981.
Waterloo, Ontario, Canada, June 10-12, 1981.

[4] Roberts, Teresa L.
Evaluation of Computer Text Editors.
Technical Report SSL-79-9, Xerox Palo Alto Research Center, November, 1979.

[5] Roberts, Teresa L., and T. P. Moran.
Evaluation of Text Editors.
In *Human Factors in Computer Systems*. 1982.
Gaithersburg, MD, March 15-17, 1982.

[6] Robertson, G., D. McCracken, and A. Newall.
The ZOG approach to man-machine communication.
International Journal of Man-Machine Studies 14:461-488, 1981.

[7] Yoder, Elise, and R. Akscyn.
ZOG Users Guide.
Technical Report, Computer Science Dept., Carnegie-Mellon University, 1982.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CMU CS-82 122	2. GOVT ACCESSION NO. AD-A123363	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) EXPERIMENTAL EVALUATION of TOOLS for TEACHING the ZOG FRAME EDITOR		5. TYPE OF REPORT & PERIOD COVERED Interim 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) C Ramila Robertson Robert Akscyn		8. CONTRACT OR GRANT NUMBER(s) N00014-76-0874
9. PERFORMING ORGANIZATION NAME AND ADDRESS Carnegie-Mellon University Computer Science Department Pittsburgh, PA. 15213		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Arlington VA 22217		12. REPORT DATE May 18 1982 13. NUMBER OF PAGES 16 Pages
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Approved for public release; distribution unlimited

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Abstract

This paper focuses on ZOG, a rapid-response, menu-selection, software system intended as a general-purpose interface to a computer. ZOG databases are networks of screen-sized displays called frames. ZOG's frame and net editor (ZED) combines coventional text-editing facilities with facilities specialized to the network character of the database. One of the design goals for ZOG is that ZOG be relatively self-contained in terms of instruction on the use of ZOG and ZED. This paper

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

compares two ZOG-based tools for teaching naive users to edit with ZED: an on-line (net) users' manual and an off-line users' manual (derived from the on-line manual). They are compared first with each other, then with eight editors evaluated by Roberts and Moran [5]. The results indicate that (1) off-line and on-line manual users take about the same time to complete a standard instruction sequence, but (2) off-line users use ZED more effectively at the end of the sequence. Finally, (3) ZED learning falls in the middle of the range of Roberts and Moran's editors in terms of minutes required on average to learn to do a new editing task.