Editing Tools for ZOG, a Highly Interactive Man-machine Interface

Donald L. McCracken and George G. Robertson

Carnegie-Mellon University

Abstract

The ZOG project at Carnegie-Mellon University is investigating a novel man-machine interface predicated on a large, rapid-response menu-selection network, with each node in the network being a display-screen-sized menu called a *frame*. This paper begins by introducing ZOG: its basic operation and its essential properties. Editing of frame networks (including their creation) has a major role in ZOG use due to the need for large networks. This is particularly true for an application called the ZOG Project Management Net, which is a large, shared frame network used as a communication medium by ZOG project members. Some initial data from Management Net use leads to observations on use of the ZOG editing tools. The existing ZOG editing facilities are then briefly described, followed by discussion of three separate approaches for obtaining advanced editing tools.

1. Introduction to ZOG

ZOG [Robertson, Newell and Ramakrishna, 1977; Robertson, 1979] consists of a potentially very large database of information structured as small, interconnected pieces. There are facilities for navigating among the pieces, using the display screen of a computer terminal as a window to view one piece at a time. Each piece of the database, called a *frame*, is structured by convention with a one-line title at the top of the screen, a few lines of text just below the title, then a set of numbered (or lettered) menu items of text called *selections*. It is the selections that interconnect the frames. When an item is *selected*, by typing its number (or letter) at the terminal keyboard,[1] ZOG "moves" immediately to the frame "pointed to" by the selection (i.e., the new frame appears on the display, replacing the frame that was being viewed before). This new frame has the same general format as the original. It typically contains further information on the selected item, and itself has selections leading to yet more detailed information. (At some point, there must of course be "dead-ends" where no selections point to further frames.) The result is a hierarchical structuring of the information. The whole network of frames in a ZOG system is called a *ZOGnet*.

The philosophy behind ZOG was pioneered by the PROMIS (Problem Oriented Medical Information System) group at the University Of Vermont medical school [Schultz and Davis, 1979]. The PROMIS system, designed to support the operation of a hospital, has a frame network of about 40,000 frames, which is sufficient to represent all the procedures of the hospital staff, plus a significant body of medical knowledge (e.g., drug information and lists of diagnostic procedures). In addition to the frames, there are large patient record and population study files. Rapid-response, touch-selection terminals provide a simple, uniform interface to this large, varied database, plus a command interface for taking action. The early PROMIS experience demonstrated significant benefits of such an interface, in effectiveness and ease of use. This prompted initiation of the ZOG project at CMU to study the PROMIS interface as a general communication interface -- to determine its applicability over a wide range of tasks and study its crucial parameters in order to understand and optimize it.

Figure 1 shows an exemplary ZOG frame. This frame, called **Mng1** (see upper right hand corner), is the initial frame for the ZOG Project Management Net, which is being used by members of the ZOG project as a communication medium. In this frame there are ten selections: seven numbered 1 to 7, plus three lettered M, U and B. The line at the bottom beginning with edit and ending with info contains the so-called *global pads* which provide global ZOG functions for navigating through the network, for getting help on the use of ZOG, for invoking the ZOG frame editor, and for making comments to the ZOG maintainers. These are invoked by merely typing their initial (lower case) letter, and are normally available at every frame in the ZOGnet.

```
Top of the ZOG project management net                                Mng1

This net is the repository of all information for the ZOG project concerning
status, goals, design issues, and data analysis.
Its function is to enhance the flow of communication among the ZOG people.
Select any of the options below (by typing the number) for more information.

  1. Status information: system versions, ZOG networks, ...

  2. Goals and work in progress                          M. Mail

  3. Issues (problems, suggestions, etc.)

  4. Data on ZOG use, results of analyses                U. Update log for this net

  5. People involved with ZOG

  6. Meetings of the ZOG group

  7. Administrative details                              B. Backup procedure for this net


edit  help  back next mark return  zog  display user  comment  goto find  info
```

Figure 1: ZOG frame **Mng1**

[1] Special ZOG terminals exist with touch-sensitive screens, allowing selection by simply touching the desired item. These are, however, still too expensive and unreliable for widespread use.

It is crucial that the "response-time" of ZOG (i.e., the time for the new frame display to appear once a selection is made) be very fast -- no more than one second. This is fast enough that the user does not feel limited by the system. He can make rapid side explorations without forgetting the main thread he is pursuing, and he can move rapidly down long paths he is thoroughly familiar with. Given human short-term memory limitations, this rapid traversal seems essential if the user is to integrate related information that happens not to be close together in the ZOGnet.

Current ZOG use at CMU is based on 1200 baud terminal links. This produces an average ZOG response time of about 4 seconds, which is (barely) tolerable for most ZOG use.[1] Two special ZOG terminals exist that are capable of 1/20 second response in conjunction with a touch-sensitive screen. Since there are too few for a large community of users, their main purpose is experimental investigation of the response-time parameter.

It is also crucial for ZOG that there be capacity for a large number of frames -- tens of thousands or more. If ZOG is to be effective, the user must find himself in a world where all of his questions and all the data he requires have been laid out in advance in the ZOGnet. At every frame there must be selections to be taken that deal with whatever information, elaboration, help and explanation may be required.

1.1. ZOG as an integrative medium

ZOG allows information of many types and from many sources to be integrated into a single ZOGnet. ZOG is not just a *database* system in the standard sense of that word. A ZOGnet can accomodate high-level concepts and discussion of issues in addition to low-level, highly-structured data. For example, some or all of the following types of information might be built into a ZOGnet, depending on the application:

Background information
Historical information
References to literature
Glossaries
Examples
Raw data
Tutorials
Discussions of issues
Indexes along many dimensions

All this various information can be connected wherever appropriate, with extra frame selections that "cross-reference" (lead directly to) related information of a different type.[2]

1.2. ZOG as a simple, uniform interface

It is not hard to learn how to navigate through a ZOGnet. In our experience, a single session of a half hour or less is all that is necessary to understand basic use of ZOG. Thus, ZOG's simplicity and uniformity is a significant virtue. Also, ZOG is less likely to be viewed by new users with distrust since it is a totally passive object -- a mere tool that is totally under the user's control.

1.3. ZOG as a dynamic, shared medium

The existence of ZOG on a computer system permits a whole community of users to be simultaneously using a single version of the frame network. There are technical problems when many people try to change the same part of the ZOGnet at the same time, but solutions to these problems exist. This opens up the possibility of ZOG as a communication medium for group discussion or debate. The ZOG group itself is now using such a shared ZOGnet, called the ZOG Project Management Net, for cooperative management of the project.

1.4. ZOG as an active interface to other systems

ZOG is more than just a static text-display system. Selections can have actions associated with them in addition to pointing to other frames. These actions can communicate with any arbitrary computer program, essentially as if the user were directly running this other program himself. Thus, ZOG can act as an interface to other programs, with ZOG selections being used to control them. An example of this is the *integrated programming environment* application, with ZOG replacing the standard command language on a computer system.

2. The Role of Editing in ZOG

The problem with ZOG as a viable communication interface is, of course, the human effort required to produce the large frame networks. Estimates of the effort invested in the PROMIS network exceed 100 man years, or approximately 5 frames per man-hour. Our initial experience with ZOG also yields this rough figure of 5 frames per hour as an overall measure of creation rate. This is based on scanty evidence at this point,[1] yet it is remarkable how the measure has held close to 5 over several different task domains and several different builders. Editing overheads constitute a significant portion of this creation time (exactly how much we can't yet say), though not nearly all of it. The human cognitive task of organizing new material to fit into the network structure is very difficult, and consumes large amounts of time that no conceivable editing aid could circumvent. Because of this, we would be quite happy to achieve a mere doubling of the frame creation rate from 5 to 10 frames per hour through improvements to ZOG's editing facilities.

The exact role of the editing facilities in ZOG depends on the application. Some ZOG applications will be relatively static once their ZOGnets are completed, so that for every hour spent in building the ZOGnet, there might eventually be many hours spent using it, with no substantial further changes. The medical knowledge component of PROMIS is a good example of this. At the other end of the static-dynamic spectrum is a network such as the ZOG Project Management Net, whose daily use involves heavy editing by many users. This latter case makes the largest demands on ZOG's editing facilities. For this reason, plus the fact that we have begun to routinely collect statistics on

[1] Future plans call for access to a dedicated ZOG machine from all departmental terminals with a 4800 baud display rate, producing an average response time of about one second.

[2] A class of ZOG applications called issue analysis make heavy use of ZOG's integrative capabilities. See [Mantei and McCracken, 1979] for details.

[1] We had as of February 1979 around 5,000 frames divided among a dozen or so applications. Reasonably accurate statistics on frame creation rate were kept for fewer than 1,000 of these.

use of the Management Net, we will focus on this particular application of ZOG in much of what follows.[1]

3. The ZOG Project Management Net

The user community of the ZOG Management Net is a small group (about eight) of professionals (faculty, staff, and graduate students) directly involved with the ZOG project. The purpose of the management net is to provide group communication on all aspects of ZOG: system changes, goals of the project, discussion of issues, proposals for changes or new features, reports of system problems, and information on meetings. A typical session with the Management Net lasts about a half hour. The user normally reviews in a systematic manner all changes to the net since his last session,[2] commenting on each as he sees fit with his own editing changes. In addition, he might well add several new items, to be commented on by later users. Most editing changes are simply adding some text to an existing frame, though occasionally whole new frames are created. Averaged over the long term, the appropriate measure of growth for the Net is the number of new frames rather than the number of editing operations.

Public use (by the ZOG group) of the Management Net did not start from scratch. The Net was primed with an overall organizational structure, plus a couple of hundred frames of information (mostly on known problems and issues), for a total of about 275 frames. After the first month of public use the number of frames had grown to 400.

3.1. The Structure of the Management Net

The frames of the Management Net are partitioned into ten *subnets* corresponding to the following functional categories: root structure, mail, status, goals, issues, data analysis, people, meetings, administrative details, and update log. Much of this structure is reflected by the selections on the root frame shown in Figure 1. Such subnet groupings are supported by the basic ZOG system. They provide a way of selectively operating on subsets of frames (i.e., all frames in a given subnet), plus an important mnemonic aid (as the alphabetic part of a frame's identifying name) for helping a user to know what context he is in while roaming around a ZOGnet.

Recall the earlier description of a frame in connection with Figure 1: a *title* on the top line, then some lines of *text* (called *frame-text* when necessary to distinguish it from *text* in the general sense), then some *selections*, and *global pads* across the bottom line. There are really two kinds of selections: the numbered selections in Figure 1 are *options*, while the lettered selections are *local pads*. The distinction is largely a convention observed by builders of ZOGnets. *Options* normally provide straightforward elaboration of the information conveyed by the frame title and text, while *local pads* have a more specific function such as cross-referencing, leading directly back to frames higher in the structure, or perhaps providing a utility function. A static analysis of the Management Net provides the following picture of the average frame:

title: 35 characters
frame-text: 190 characters (3 lines)
3 options: 60 characters each
2 local pads: 15 characters each

The relatively small average size of frame-text (only 3 lines), plus the fact that the options seem to contain as much information as the frame-text, are reflections of ZOG philosophy. The amount of information on each frame is purposely kept fairly small, and the structuring of information as option lists is carried through even to the terminal nodes of the network.

3.2. Editing in the Management Net

As of this writing the ZOG Management Net has only been opened up to public use for about a month. The data presented here are based on the initial two weeks of public use. During this period six people yielded a total of 56 editing sessions, with a total editing time of 23 hours. There were 84 new frames created. At the end of the period there were a total of 360 frames. With only this brief experience, we have already learned some interesting lessons about ZOG's editing tools:

1. Slow creation rate

The users of the Management Net seem to find its structure transparent, and say that it is "easy" to add things. Yet the average creation rate for the first two weeks was only 3.6 frames per hour. This is somewhat slower than the 5 frames per hour we have obtained for other ZOGnets. Such a comparison may not be appropriate since the goal of Management Net use is not to create new frames as efficiently as possible, but rather to communicate with other people. Also, the structure of the Management Net tends toward fuller frames than in other applications. For example, a ZOGnet of 330 frames on cognitive psychology averages 360 characters per frame, compared to about 460 for the Management Net. In future, we may attempt to normalize for frame size by measuring growth rate as number of new text characters per minute. The initial period of Management Net use produced 30 characters per minute.[1] (It is interesting to note that this rate is roughly an order of magnitude slower than average typing rates of the Management Net users. This gives an overall perspective to the claim that the hard part of ZOG editing is deciding what text to add and where to add it, not actually entering the text.)

2. ZOG exploration is important even when editing

One might guess that in editing a ZOGnet most of the time would be spent using the editing tools. However, a considerable amount of time goes into deciding how to structure the material being entered: where it fits in the existing ZOGnet, how to format it, and how to link it via cross-references to related material. This activity requires significant exploration of the net in preparation for actually making the desired changes. During the initial two-week period, 55% of the editing time was spent in moving around the net looking at frames rather than making changes. For every frame created, 29 frame accesses were made (with a

[1] Another important consideration for focussing on the ZOG Management application is its relevance to Office Automation (the subject of the conference session which contains this paper). A ZOG system similar to the Management application could be construed as a communication medium for an office environment.

[2] The Find facility mentioned in the next section provides the user a convenient way to find all changes since a certain date and time.

[1] The Management Net application involves very little deletion of material -- old material is simply marked as such and left. For applications that do involve substantial deletion of material, the measure of new text per minute might have to be modified.

19 sec average stay at each frame). For every invocation of the frame editor (to edit an existing frame or create a new frame), about 6 frame accesses were made. One important conclusion of this is that fast reponse is important for editing as well as for pure accessing; i.e., a slow terminal is not satisfactory for editing of a ZOGnet.

4. Current ZOG Editing Tools

4.1. ZED: the ZOG frame editor

Most ZOGnet building is currently done with an elaborate screen editor called *ZED* (for Zog EDitor) which is used both for creating new frames and modifying existing frames. ZED is a two-dimensional generalization of the *alter mode* in the SOS text editor [Weiher and Savitzky, 1974]. During an edit, ZED maintains the display so that the builder always sees the frame exactly as a user would see it. The display is structured as a set of *items* corresponding to the internal structure of the frame (title, text, options, and local pads). There are two sets of commands, one for moving around and manipulating items, and the other for manipulating the text of a single item. ZED also has commands for manipulating the various non-displayed parts of a frame, e.g., next-frame links and actions. Our experience indicates that ZED is an effective tool for creating and manipulating individual frames, although users not familiar with computer systems often have difficulty learning how to use it.

4.2. Creation by selection

When using ZOG, selecting a menu item that does not yet lead to another frame can automatically invoke ZED to create the missing frame. This allows a natural top-down style of building ZOGnets.

4.3. Find and Substitute

The Find facility in ZOG has several different uses, both for the ZOGnet builder and user. Find will search the ZOGnet, or a subpart of it, for all frames: (1) with a particular character string in the displayed text, (2) modified by a particular person, (3) modified since a particular date and time, or (4) any combination of the first three. It is used by the ZOGnet builder to locate a frame that is to be the destination of a new link, or to see some text in a frame different from the one being edited, to help with wording or organization. For shared ZOGnets such as the ZOG Project Management Net, Find is often used to locate all changes made to the ZOGnet since the last time the user accessed it.

The Substitute facility makes text substitutions throughout whole regions of a ZOGnet. As with the Find facility, Substitute allows the builder to specify some set of frames by text string, date and time of last modification, and name of last modifier. The builder also specifies a new string to substitute for the original search string. During the substitution, the builder may request that each substitution be displayed for approval before it is finalized.

5. Advanced Editing Tools

The basic editing tools described above are for interactive construction and modification of single frames. They are good at what they do, but we find a serious need for higher-level tools that can manipulate collections of frames. Below we discuss three current approaches to higher-level editing tools for ZOG.

5.1. ZOGnet Schemas

One advanced ZOGnet editing tool being developed attempts to exploit the structural regularities that are likely to be found in any large ZOGnet. It allows the net builder to create generalized pieces of ZOGnet called *schemas* which may then be used many times during the building process by copying them and instantiating their variable parts to suit the particular purpose.[1]

Schemas tend to be most effective for nets with a relatively high uniformity of structure. Some ZOG applications do lend themselves to such rigid structure, but many interesting applications seemingly do not. We suspect a case might be made that forcing uniformity of structure onto a ZOGnet will increase its effectiveness, especially for new users trying to learn the structure, but this still must be proven. It is one of several important questions we hope to obtain experimental evidence for in the near future.[2] If the evidence points with favor to high degrees of structure, then we expect schemas will be an important net-building tool.

Schemas apply at the time of creation, and do not seem to work for retroactive editing. This is serious simply because a builder is usually unable to specify beforehand what the structure of his net should be. The notion of appropriate structure grows incrementally from the experience of building the net itself. Thus, we are looking for ways that schemas can be used to "coerce" structure upon frames created before the final form of the schemas was determined.

5.2. The External Database Approach

Another advanced set of tools is being developed for a Library Database application of ZOG. The BROWSE system [Fox and Palay, 1979], based on ZOG, is designed to allow a user to *browse* through an organized database listing all the holdings of the CMU Computer Science Library. Each entry contains regular bibliographic information including abstract, location, alternate classification, and pointers to related entries. The entries are maintained in a database separate from ZOG and in a mostly ZOG-independent form, with special tools for creating, editing and searching. In addition to the content information, the database contains definitions of many entry types (e.g., book, technical report, journal article) and corresponding frame formats for ZOG. (These type-definitions are functionally similar to the schemas mentioned above). A compilation process produces a ZOGnet from this separate database, obtaining both the content and the formatting information from the database.

This approach has advantages in ease of global format changes, and in the ability to make construction of rich cross-linkages in the ZOGnet part of the automatic compilation process. But is also requires a highly structured domain, and many interesting ZOG applications do not seem to have the requisite structure.

---

[1] This work is being done by Kamesh Ramakrishna.

[2] The ZOG project is strongly concerned with psychological study of the man-machine interface. [Newell, 1977] contains some initial notes on the approach, while [Mantei, 1979a; 1979b], [Mantei and Cattell, 1979] and [C.K. Robertson and Newell, 1979] contain some initial results of psychological studies of ZOG.

5.3. Global ZOGnet Editing Tools

There do not yet exist adequate global editing tools for ZOG; i.e., tools that manipulate many frames at once. The Substitute facility described above operates globally, but is limited in the nature of changes it can make. What we need is a facility for tasks such as adding a particular new option to certain frames (i.e., those that satisfy some condition on their existing structure). We would also like in some cases to operate on groups of frames as a unit rather than being limited to editing operations on single frames.

There are two approaches to obtaining global editing tools, both of which are being pursued. The first is gradual accretion of specialized facilities -- when a strong need surfaces, a facility to meet that need is designed and added. The Find and Substitute facilities are cases where specialized facilities evolved to meet specific needs.

The second approach is to define a language for specifying editing operations, and then express every immediate editing need in terms of the language. This is a more flexible and open-ended approach, but may be inefficient or ill-adapted for many particular editing tasks. One possibility for such a language is a production system, which is a form of programming system using condition-action rules operating on a global working memory [Waterman and Hayes-Roth, 1978]. For ZOG we would like to express the conditions and actions themselves as frames, to avoid the need for a new specification language of some sort. In fact, we are pursuing the idea of using the schemas discussed earlier, with their variables, to specify match conditions on frames that are candidates for editing.

We are not completely optimistic about the success of a condition-action language for global editing. For one thing, the global editing operations we think we need may be so diverse as to belie the use of a single mechanism to specify them. Also, many predicted global editing needs involve quite complex rules that are probably beyond the scope of any feasible automatic editing tool. We will undoubtedly have to leave the user in the loop to oversee the application of the editing rules and to supply some hand adjustments. The global editing task, since it must ultimately deal with the meaning of language, is a challenging problem for the field of artificial intelligence.

6. Conclusion

ZOG appears to have potential as a universal man-machine interface, effective in a wide variety of applications for both naive users and experts. It replaces the notion of an intelligent communication agent with a passive tool whose intelligence has been pre-compiled into its large frame network. The problem with such an interface is fundamentally an editing problem: How do we create this large, intelligently-structured database? Our existing ZOG editing tools work well at the level of individual frames, and their use is illustrated in the ZOG Management application. Yet there is a critical need for more sophisticated tools that deal with larger units of structure. Three current approaches to such advanced tools may yield moderate gains in editing efficiency. Unfortunately, the ideal for ZOG editing seems to be beyond our grasp. The artificial intelligence problem of how to build an intelligent machine for communicating with humans has not been avoided with ZOG; it has rather been transmuted into the problem of building an intelligent machine to aid the creation of ZOG networks.

7. Acknowledgements

This work was supported by the Office of Naval Research under contract number N00014-76-0874, and partially by the Defense Advanced Research Projects Agency (DOD), ARPA Order 3597, monitored by the Air Force Avionics Laboratory under contract number F33615-78-C-1151.

We would also like to acknowledge the extensive support by other members of the ZOG group: Allen Newell, Kamesh Ramakrishna, Marilyn Mantei, Andrew Reiner, Andrew Palay, Mark Fox, Lynne Reder, Lee Gregg, and Charlene Robertson.

8. References

Fox, M. and Palay, A. (1979), "The BROWSE System: at the Interface of Computer and Information Science", submitted to the 1979 *Annual Conference of the American Society of Information Science.*

Mantei, M. (1979a), "Modeling User Behavior in Computer-Presented Learning Tasks", to be presented at the *American Education Research Association Annual* Meeting, April 8-12, San Francisco.

Mantei, M. (1979b), "Disorientation Behavior in Man-Computer Interfaces", Ph.D. thesis, Annenberg School of Communications, University of Southern California, Los Angeles.

Mantei, M. and Cattell, R. (1979), "A Study of Entity-Based Database Interfaces", Technical Report, Computer Science Department, Carnegie-Mellon University, Pittsburgh.

Mantei, M. and McCracken, D. (1979), "Issue Analysis with ZOG, a Highly Interactive Man-Machine Interface", to be presented at the *First International Symposium on Policy Analysis and Information Systems*, Duke University, June 28-30, 1979.

Newell, A. (1977), "Notes for a Model of Human Performance in ZOG", Technical Report, Computer Science Department, Carnegie-Mellon University, Pittsburgh.

Robertson, C.K. and Newell, A. (1979), "Iterative Predictive Evaluation: A Method for User Studies", in preparation.

Robertson, G. (1979), "Some Design Considerations for the ZOG Man-Computer Interface", *Proceedings of Third NATO Advanced Study Institute in Information Science*, in press.

Robertson, G., Newell, A., Ramakrishna, K. (1977), "ZOG: A Man-Machine Communication Philosophy", Technical Report, Computer Science Department, Carnegie-Mellon University, Pittsburgh.

Schultz, J. and Davis, L. (1979), "The Technology of PROMIS", to appear in *Computer*, IEEE Computer Society.

Waterman, D. and Hayes-Roth, R. (eds) (1978), *Pattern-Directed Inference Systems*, Academic Press.

Weiher, W. and Savitzky, S. (1974), "Son of Stopgap (SOS)", Stanford Artificial Intelligence Laboratory Operating Note 50.3 (1971), Carnegie-Mellon University edition, Pittsburgh.