1984

Knowledge Organization for Command Languages

Barry D. Floyd
The University of Michigan

Marilyn M. Mantei
The University of Michigan

Knowledge Organization for Command Languages

Barry D. Floyd and Marilyn M. Mantei
Graduate School of Business Administration
The University of Michigan

ABSTRACT

With the rising cost of the human resource and the increasing use of on-line information systems in the business environment a serious concern over user productivity in the use of interactive systems has arisen. This concern is well-founded.

Reports indicate that improving the productivity of computer systems users can have a major impact on costs. For example, Shneiderman (1982) reports on a Quality Inn study which indicates that for each second deducted from the average 150 seconds taken to make a room reservation, $40,000 per year is saved. Olson (1983) reports that for AT&T, each one second reduction in the average of 30 seconds spent per telephone operator transaction (e.g., requests for a telephone number), translates into a savings of $20,000,000 per year. In each of these two examples the operators must interact with on-line computer systems to perform their tasks.

Other research indicates that users of on-line information systems take a large amount of time to correct errors (Norman, 1981; Card, Moran, and Newell, 1983). Given the above costs per second of task performance errors, which typically take many seconds to correct, can be expensive. The research described in this paper addresses this error problem.

In order to understand why users make errors we must understand how they use command languages. As a first step in this understanding, we look at the internal memory structure users build of command languages. We believe that this structure has a role in error generation in the following way. Information is organized in human memory into many small groupings which psychologists call "chunks." Whenever a user is required to access a chunk of information in memory, the entire contents of the chunk are retrieved. If only a portion of the chunk is needed in the interaction, additional processing is required for the selection of the portion needed. Because the access and use of this piece of information does not typically require additional processing (or it would not have been stored as a single unit to begin with), a conflict exists between performing the usual behavior of retrieving and using the entire chunk versus that of retrieving the chunk and then performing the additional processing which selects the appropriate portion. The retrieval and use of the entire chunk typically wins this processing conflict leading to the error or outputting too much information.

Because of this phenomena, we build a case in the paper against the design of interactive scenarios that cause the initial building of a specific human memory organization that is then violated by some later, less frequently used interactions (e.g., system error messages). We claim that these less frequently used interaction scenarios will result in many errors.

*Paper is under revision for Communications of the ACM.

We do not show in this paper that these errors take place (although some of our preliminary studies support this claim (Floyd, 1984)). Our work is a first step in understanding why these errors occur. In order to demonstrate the validity of our claim, we need to show that extremely regular human memory structures are built through a user's experience with command language, since it is these structures that are prone to error when violated. To do this we took an existing computer command language for a time sharing system and predicted what subunits of the language would be grouped together in the user's own memory. We based our predictions on the way in which the syntax of the language forced portions of the language specification to be optional and other portions to be required. We reasoned that the required portions would be stored as a single chunk since they would always be used together and that each distinct optional portion would also be stored as a single chunk.

The language which we chose to study is MTS, an operating system command language in use at the University of Michigan. MTS is a user-initiated command language where the first component of the command is the command verb. The verb is followed by positional parameters and then by keyword parameters. The end of a command is indicated by pressing a return key. An example command is shown below.

```
RUN *FTN INPUT = SOURCEFILE OUTPUT = LISTING T = 4
```

We hypothesize that the users' knowledge organization for MTS commands looks as follows: The primary organizational unit is the command. This major organizational structure is divided into substructures which consist of either the command verb and its required parameters or each of the optional parameters used with the command. For keyword parameters this consists of the keywork and the keyword argument. We expect that the substructures are the chunks in the interaction which should not be interrupted.

The experiment we ran to test this proposed memory organization uses a reading/ verbatim recall task. In the experiment, we asked MTS experts to view each word of a sequence of MTS commands individually, i.e., one word per screen display. The subjects' task was to produce a verbatim recall of the entire command sequence immediately after viewing the last word in the sequence. Each subject controlled the amount of time each word in the command sequence was displayed by pressing a response key which erased the current word and displayed the next word. The viewing time differences between single words were used to provide border knowledge organization (chunk) boundaries. These predictions are based on the theory that subjects will pause to recode information presented on a recall task at those boundaries that correspond to internal organization boundaries.

To determine whether our proposed chunk boundaries are statistically significant in describing reading times, we constructed a linear regression model with the reading times as the dependent variable. The independent variables represented (1) the boundary conditions where variables are coded to indicate whether a word is located either before of after predicted chunk boundaries, and (2) word characteristics which are known to affect reading times, e.g., word length.

The regression results showed significantly longer word reading times at the memory organization boundaries that we predicted. Our hypotheses are therefore supported by the experimental data, and thus provide support for our theories on how users mentally organize command languages.

These results are expected to impact the design of command languages. For example, research literature in other domains support the claim that disruption of a person's knowledge organization adversely affects performance. This implies that an on-line system should be designed so that it does not disrupt the structure learned by the user. Any aspect of a system which requests a user to specify a subpart of a command, such as a parameter value, should take into account where the mental boundaries of the subpart are located.

In retrospect, we find that we can take a wide variety of interactive computer procedures and determine the internal memory organization for these sequences through this technique. We have also developed a method for predicting what the memory organization for an interactive language will be, given the syntax and typical usage parameters of the language. Although this method has not been completely worked out, future research can use the method in studies on the ease of learning and using interactive languages.

In summary, we have shown that users do build an internal memory organization for an interactive language. We have combined these results with other research on memory organization to make recommendations for the design of interactive sessions, namely, when and how to interrupt the user. We have also laid the groundwork for future research on how to design learnable and usable interactive languages.

REFERENCES

Card, S.K., Moran, T.P. and Newell, A. *The Psychology of Human—Computer Interaction.* Lawrence Erlbaum Associates: Hillsdale, New Jersey, 1983.

Floyd, B.D. *Dissertation Proposal: Knowledge Organization for Command Languages.* Graduate School of Business Administration, The University of Michigan, Ann Arbor, Michigan, 1984.

Norman, D.A. "Categorization of Action Slips," *Psychological Review,* Volume 88, Number 1, 1981, pp. 1-15.

Olson, J.S. Reitman. Personal communication based on data gathered in Bell Laboratories study, 1983.

Shneiderman, B. "The Future of Ineractive Systems and the Emergence of Direct Manipulation," *NYU Symposium on User Interfaces,* New York University, New York, New York, May 26-28, 1982.