PROCEEDINGS of the HUMAN FACTORS SOCIETY—29th ANNUAL MEETING—1985

USING COGNITIVE MODELS TO CREATE MENUS

Kathleen M. Snyder, Alan J. Happ, Lawrence Malcus,
Kenneth R. Paap, and James R. Lewis

International Business Machines
Boca Raton, Florida
Santa Teresa, California

ABSTRACT

This experiment was conducted to generate menus for a command language based on the cognitive structures of expert and naive users. Expert and novice menus were generated from networks derived by applying a network analysis algorithm (Pathfinder) to similarity ratings of command functions. The results of a second experiment to test the effectiveness of the menus as training (transition) devices for using the functions in a command-driven style will be reported.

INTRODUCTION

Research in cognitive psychology has shown that individuals differ in their ability to process information depending on their expertise. Studies have demonstrated differences in cognitive structure in domains such as chess, music, physics and computer programming (Chase & Simon, 1973; Sloboda, 1976; Chi, Feltovich & Glaser, 1981; Adelson, 1981; McKeithen, Reitman, Rueter & Hirtle, 1981). One explanation for these differences is that experts are more efficient at chunking meaningful information into memory. Instead of attending to individual pieces of information, the expert processes groups of information which results in more efficient memory encoding and recall performance. These findings have implications for the design of software. They suggest that software should be based on the cognitive structures of experts since these provide the most efficient means of organizing material. However, this inference is valid only to the extent that the structures of the expert are accessible to the novice.

Novice users are at a particular disadvantage when learning a system since they are unfamiliar with both the tasks that can be accomplished and the command names for invoking the tasks. Menu driven interfaces are one way of reducing memory demands because they only require the user to recognize rather than recall the correct option. A common observation is that novices prefer a menu interface to a command structure as their introduction to a new system. These preferences change, however, as the novice user gradually makes the transition to an expert (Gilfoil, 1982). Command interfaces are preferred by experts since functions are more readily accessible. These findings can be be viewed as supporting the idea that a menu is useful as long as it aids in the development of an appropriate model of the system by the user. Since the user will shift to a more efficient dialogue style when the menu structure is no longer needed, one goal of a menu interface should be to facilitate the transition from novice to expert.

The most commonly used method for designing menus has been to construct them based on the recommendations of an expert design team. The rationale for having these experts design the menus is that they are the most efficient users and should provide the best organizational scheme for teaching naive users. A recent study, however, has shown that the design of software should be validated by a population of end users, who are not necessarily experts. Lee, Whalen, McEwen, & Latremouille (1984) had experts design a top index page for an electronic shopping and information retrieval system. A second group of experts ranked the menus for ease of use. Virtually no correlation (r = .08) was found between the expert ranks. When the menus were ranked by a group of representative users, however, there was high agreement as to which menus would be easier to use (r = .49). Performance measures were also highly correlated with the end users' predictions (r = .72).

Another method for designing menus based on the cognitive structures of users has employed techniques used in cognitive psychology which tap into the organization of information in memory. Researchers employing this method have used the concept of semantic distance to determine the proximity (or associative strength) of specific concepts in memory. Most of these studies have used similarity ratings or a sorting task to generate semantic distance matrices. The data are then subjected to algorithms which generate multidimensional spatial representations of the distance between concepts. McDonald, Stone, Liebelt & Karat (1982) used this method to design menus for a text editor based on the cognitive structure of individuals experienced with text editing. Although novices made fewer errors on this menu than a randomly organized one, the effect was weak. A possible explanation for this finding may be that the model of the experienced user was not directly usable by the novice. Little research has focused on the use of differing cognitive models to construct menu structures which provide the smoothest transition from a novice to an experienced level. A model facilitating this transition should result in shorter time to access functions in a menu dialogue and quicker recall of commands in a command dialogue.

One way of designing menus to reduce access time would be to increase the probability that the user will find the desired option. Roske-Hofstrand and Paap (1985) demonstrated this principle by designing a network of information screens for a large control-display unit for aircraft pilots. The Pathfinder algorithm (Schvaneveldt, Durso & Dearholt, 1985) was used to empirically derive the cognitive networks of pilots for this complex control-display unit. The Pathfinder algorithm generated a network of weighted links that connected pairs of highly related concepts. As a validation test, three menu structures were generated from the network and compared to each other and a fourth menu constructed by a design team. The three network-based menus differed in their amount of redundancy. Redundancy refers to the number of times a concept appeared under more than one major heading. The design team's menu was a strict hierarchical menu structure. When access times were compared, pilots in the highly-redundant condition outperformed the other conditions. These results suggest that a menu structure which provides many meaningful links between concepts should be easier to learn than a menu providing fewer links or no redundancy.

The goal of the present study was to generate menus for a computer operating system based on the cognitive structures of expert and naive users. Expert and novice menus were generated from networks derived by applying the Pathfinder algorithm (Schvaneveldt et. al., 1985) to similarity ratings of operating system commands. A second experiment will be conducted to test the effectiveness of the menus as training (transition) devices for using the operating system in a command-driven style.

METHOD

Subjects

Nineteen subjects were recruited from within IBM and through a local employment agency. Nine subjects were unfamiliar with computer operating systems and formed our Naive group. Ten subjects were highly experienced with the operating system and formed our Expert group.

Materials

The stimulus set for the sorting phase comprised forty-three 4 X 6 cards. Each card consisted of a command name, a brief definition of the command, and an example of how to use the command. Similarity data were collected on IBM XT and AT personal computers.

Procedure

Before beginning the sorting task, subjects read a brief introduction to provide background information on the operating system. Subjects were then instructed to read each of the cards, spread them out on the table, and "sort them into piles of related commands". Subjects were free to sort using any size or number of piles. After finishing the sort, subjects labelled the piles. If a subject was unable to name a pile, that pile was labelled with a letter. Subjects were then asked if they wished to break the piles into smaller piles. Subjects who chose to change their piles, were asked to label the new piles.

Upon finishing the sorting phase, each subject's groupings were entered into the computer. A grouping consisted of the label of a pile and the commands which were placed in that pile. The similarity ratings were then collected using the computer to present stimuli. A pair of group labels appeared on the CRT screen. Subjects rated the similarity of the two groups on a nine point scale, 1 being very similar, 9 very dissimilar. The cards remained in view and the subject could refer to them while performing the rating task. Except for comparing a group to itself, all possible comparisons were performed. Thus, subjects provided two ratings for each pair of groups (Group A to Group B and Group B to Group A).

A distance matrix was constructed from each subject's similarity ratings for the commands. The subjects' ratings were transformed to an 11 point scale (0 - 10) in order to include values for the similarity of a command to itself and the commands within a pile to each other. The similarity of a command to itself was assigned a value of zero. All commands placed in the same group were given a rating of one. The remaining pairs of commands received the transform of the value the subject had provided when comparing the two groups to which each pair of commands had been assigned. The transformation was accomplished by adding 1 to each of the group comparisons. The two ratings for each pair of commands were then averaged to give the values used in the network analysis.

RESULTS AND DISCUSSION

The Pathfinder algorithm (Schvaneveldt et al., 1985) was used to generate networks reflecting the cognitive structures of novice and expert operating system users. A network consists of a set of nodes connected by weighted links (representing the strength of the association between pairs of nodes). Next, the set of dominating nodes was determined. This is the smallest set of nodes through which all nodes in the network can be directly reached. Any node in the network can be accessed by traversing one link beginning with a dominating node. In other words, if the dominating nodes were placed at the top of a two level menu the remaining items in the command set would be found nested under and linked directly to those dominating nodes.

The novice network contained 71 links and 11 dominating nodes. The expert network contained 75 links and 10 dominating nodes. Only 28 links and 1 dominating node were common to the two networks. These findings suggest the possibility that in making the transition from novice to experts a core of 28 links is maintained while 43 are broken and replaced by 47 new ones. However, additional data from user groups intermediate between the extremes of novice and expert are required to trace this development of the expert cognitive structure from the novice.

Using the procedures developed by Roske-Hofstrand and Paap (1985), the networks together with the set of most dominating nodes were used to design the organization of two distinctly different menu structures. These menus will be compared to each other and to a command interface in another experiment, the results of which will be discussed at the conference presentation.

REFERENCES

Adelson, B. (1981). Problem solving and the development of abstract categories in programming languages. Memory and Cognition, 9, 422-433.

Chase, W. G. & Simon, H. A. (1973). Perception in chess. Cognitive Psychology, 5, 55-81.

Chi, M. T. H., Feltovich, P. J. & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. Cognitive Science, 5, 121-152.

Gilfoil, D. (1982). Warming up to computers: A study of cognitive and affective interaction over time. In Proceedings of the Human Factors in Computer Systems (pp. 245-250). Gaithersburg, MD.

Lee, E., Whalen, T., McEwen, & Latremouille, S. (1984). Optimizing the design of menu pages for information retrieval. Ergonomics, 27(10), 1051-1069.

McDonald, J. E., Stone, J. D., Liebelt, L. S., & Karat, J. (1982). Evaluating a method for structuring the user-system interface. In Proceedings of the Human Factors Society 26th Annual Meeting (pp. 551-555). Seattle, WA: Human Factors Society.

McKeithen, K. B., Reitman, J. S., Rueter, H. H., & Hirtle, S. C. Knowledge organization and skill differences in computer programmers. Cognitive Psychology, 13, 307-325.

Roske-Hofstrand, R. J. & Paap, K. R. Cognitive networks as a guide to menu organization: An applica tion in the automated cockpit. Manuscript submitted for publication.

Schvaneveldt, R. W., Durso, F. T., & Dearholt, D. W. (1985). Pathfinder: Scaling with network structures. Manuscript submitted for publication.

Sloboda, J. A. (1976). Visual perception of musical notation: Registering pitch symbols in memory. Quarterly Journal of Psychology, 28, 1-16.