Strategic Approach to Computer Literacy

Suresh K. Bhavnani
HCI Institute
Carnegie Mellon University
Pittsburgh, PA 15213 USA
Tel: +1-412-268-5663
E-mail: suresh@andrew.cmu.edu

ABSTRACT
Despite experience, many users do not progress from a basic use of computer applications to a more efficient use. To address this problem, we designed a strategic approach to training which focused on teaching efficient strategies in addition to commands. A controlled experiment which compared this approach to traditional command-based instruction revealed that some strategies indeed require explicit training before they are learned. However, others are automatically acquired just by learning commands, and yet others may require more practice than we anticipated. These results have direct implications to instructional design. Because the strategic approach took the same time to teach as the traditional approach but did not harm learning of commands, it offers a promising alternative to command-based instruction.

Keywords
Strategy, strategic knowledge, efficiency, training.

INTRODUCTION

Several real-world and longitudinal studies in the use of complex computer systems such as UNIX, word processors, and spreadsheets, have shown that despite many years of experience, many users with basic command knowledge do not progress to an efficient use of applications. [1, 3, 4, 5]. Recent research suggests that strategic knowledge holds the key to efficient use. Strategic knowledge provides users alternate ways to perform a task and how to choose between them [2]. Such knowledge can reduce task time and errors, in addition to making content easy to modify [2].

Because strategic knowledge is difficult to learn through current interfaces [1], we hypothesized that this knowledge must be explicitly provided to users through training. This hypothesis was tested in a small semester-long graduate level class devoted to teaching computer-aided drawing [2]. The results suggested that strategic training could improve students' abilities to use efficient strategies. This paper presents preliminary results from a larger experiment which examined if the same strategic approach could be used to teach efficient strategies to freshmen in a limited time.

THE STRATEGIC APPROACH TO INSTRUCTION

Carnegie Mellon University requires all freshmen to pass a seven-week course called the Computer Skills Workshop (CSW). The goal of CSW is to ensure that all freshmen have basic skills to use computer applications. The course focuses on teaching basic commands to perform simple networking tasks using applications such as UNIX and e-mail, in addition to simple word processor and spreadsheet tasks using MSWord and MSExcel.

In contrast to the above approach, we designed and implemented an experimental version of the course using a *strategic approach* to training. This approach taught the same set of commands as the regular course, but in addition taught nine general strategies which were shown to be useful across computer applications. For example, students were taught that most computer applications had the power of assisting users to perform repetitious tasks. An efficient way to exploit that power was through the general strategy *operate on groups of objects*.

The general strategies were taught in two steps. *Learning to See* taught students to recognize opportunities to use efficient strategies by studying the nature of the task. For example, the *operate on groups of objects* strategy in UNIX, was taught by first showing two ways to move many files sharing the same extension: (1) move one file at a time and (2) move multiple files with the wild card operator. The first method was shown to be repetitious, time-consuming, and error prone compared to the second method. The second method was then shown to be an instance of the general strategy *operate on groups of objects*. In the *Learning to Do* step, students executed the strategy on their own for a similar task. Later in the course, the same strategy was taught in MSWord and MSExcel with different commands to emphasize its general nature.

EXPERIMENTAL METHOD

Eight CSW sections, each taught by student instructors and each containing approximately 21 students, were chosen for the study. Four sections received instruction ordinarily provided by CSW and formed the control group. The other four sections (balanced by student major) received instruction through the strategic approach and formed the experimental group. After the course ended, students in both groups were offered $25 to perform tasks in a post-test and were informed that their participation would not affect their grades. This yielded 42 control, and 48 experimental students.

These students were asked to perform tasks in UNIX, MSWord, and MSExcel, and were required to provide handwritten descriptions of how they completed the tasks. The tasks provided 12 opportunities (2 in UNIX, 4 in MSWord, and 6 in MSExcel) to use 9 efficient strategies. The tasks were designed to take a maximum of an hour and a half. Interactions were recorded through a screen capture tool and command recorders. MSWord and MSExcel documents containing completed tasks were also collected.

RESULTS AND DISCUSSION

Both groups took the regular CSW exams which tested mainly command knowledge. An analysis of their mean scores revealed no statistical difference between the two groups (96.07 control, 95.54 experimental). This suggests that even though the strategic approach took the same amount of time as the regular instruction, it did not harm the learning of basic commands.

Figure 1 shows our preliminary analysis of nine strategy opportunities (the remaining three will require lengthy video transcription). The numbers represent the percentage of students in each group who used the strategy and completed the task. As shown by the gray cells, the experimental group did significantly better than the control group in four strategies ($p<0.05$ for each of the four strategies). This result confirms that some strategies indeed require more than command instruction. For example, each of the control instructors explicitly taught how to use the split window command. However, only 10% of the control students used it in the post-test task requiring comparisons

Strategies assessed in post-test	**E**	**C**
UNIX		
1. Operate on groups of objects (e.g. mv with *)	56	17
2. Check group before operating (e.g. ls with *)	6	0
MSWord		
3. Make organizations known to the computer (e.g. tables)	92	95
4. Exploit dependencies to generate variations (e.g. styles)	46	0
MSExcel		
5. View parts of spread-out information to fit simultaneously on the screen (e.g. split window)	52	10
6. View relevant information, do not view irrelevant information (e.g. zoom)	29	10
7. Make dependencies known to the computer (e.g. make formulas dependent on cells)	98	93
8. Exploit dependencies to generate variations (e.g. modify values by modifying dependent cells)	93	83
9. Generate new representations from existing ones (e.g. charts)	98	95

Figure 1. The percentage of students in the experimental (E) and control (C) groups who used the general strategies and completed the tasks. The gray cells show statistically significant differences (based on chi-square tests).

of distant cells in a large spreadsheet. In contrast, the experimental instructors taught the split window command in the context of Strategy 5 which emphasized that when spread-out information needs to be compared or modified, they should be viewed simultaneously on the screen to avoid scrolling. This resulted in 52% of the experimental students using the strategy in the post-test. Since this strategy was taught to the experimental students only in MSWord but tested in MSExcel, the result demonstrates that strategies could transfer across applications.

In four other strategies (shown in white), most users in both groups used the strategies with no significant difference between the groups. This result shows that some strategies can be automatically acquired just by learning commands. Finally, Strategy 2 (shown in black) was mostly unused by both groups. This suggests that some strategies may require much more practice than we anticipated. Overall, the results demonstrate that strategies can be taught effectively without apparent penalties in training time, command knowledge, and task completion.

FUTURE RESEARCH

The differences in strategy use have motivated us to explore what makes strategies hard to learn and use. We are also analyzing the relationship of variables such as major, prior experience, gender, task time, errors, and class attendance, to strategy instruction and learning. The goal is to develop a framework to design computer literacy courses which not only teach commands, but also teach efficient strategies that are generalizable across computer applications.

ACKNOWLEDGMENTS

This research was supported by the National Science Foundation, Award# EIA-9812607. The views and conclusions should not be interpreted as representing the official policies, either expressed or implied, of NSF or the U. S. Government. The author thanks T. Daigneault, S. Carpenter, B. John, J. Larkin, S. Mathan, F. Peck, F. Reif, R. Thomas, G. Vallabha, and the students in the study.

REFERENCES

1. Bhavnani, S.K., and John, B.E. From Sufficient to Efficient Usage: An Analysis of Strategic Knowledge. *Proceedings of CHI'97* (1997), 91-98.
2. Bhavnani, S.K., John, B.E., and Flemming, U. The Strategic Use of CAD: An Empirically Inspired, Theory-Based Course. *Proceedings of CHI'99* (1999), 42-49.
3. Doane, S.M., Pellegrino, J.W., and Klatzky, R.L. Expertise in a Computer Operating System: Conceptualization and Performance. *Human-Computer Interaction* 5 (1990), 267-304.
4. Nilsen, E., Jong H., Olson J., Biolsi, I., and Mutter, S. The Growth of Software Skill: A Longitudinal Look at Learning and Performance. *Proceedings of INTERCHI'93.* (1993), 149-156.
5. Rosson, M. Patterns of Experience in Text Editing. *Proceedings of CHI '83* (1983), 171-175.