The Strategic Use of CAD: An Empirically Inspired, Theory-Based Course

Suresh K. Bhavnani
HCI Institute
Carnegie Mellon University
Pittsburgh, PA 15213 USA
Tel: +1-412-363-8308
E-mail: suresh@andrew.cmu.edu

Bonnie E. John
HCI Institute
Carnegie Mellon University
Pittsburgh, PA 15213 USA
Tel: +1-412-268-7182
E-mail: Bonnie.John@cs.cmu.edu

Ulrich Flemming
School of Architecture
Carnegie Mellon University
Pittsburgh, PA 15213 USA
Tel: +412-268-2368
E-mail: ujf@andrew.cmu.edu

ABSTRACT

The inefficient use of complex computer systems has been widely reported. These studies show the persistence of inefficient methods despite many years of experience and formal training. To counteract this phenomenon, we present the design of a new course, called the Strategic Use of CAD. The course aims at teaching students efficient strategies to use a computer-aided drafting system through a two-pronged approach. *Learning to See* teaches students to recognize opportunities to use efficient strategies by studying the nature of the task, and *Learning to Do* teaches students to implement the strategies. Results from a pilot experiment show that this approach had a positive effect on the strategic behavior of students who did not exhibit knowledge of efficient strategies before the class, and had no effect on the strategic behavior of those who did. Strategic training can thus assist users in recognizing opportunities to use efficient strategies. We present the ramifications of these results on the design of training and future experiments.

Keywords

CAD, strategy, training, GOMS, learning, efficiency

INTRODUCTION

Several longitudinal and real-world studies have reported widespread inefficient use of complex computer systems such as computer-aided drafting (CAD), spreadsheets, word processors, and operating systems [4, 7, 8, 11]. Given the tremendous leverage that computers can provide if used efficiently, this situation seriously limits the overall productivity of human-computer interaction.

The causes and persistence of inefficient methods to perform various tasks have been explored for many decades. Early work identified the *Einstellung* effect, which demonstrated not only the powerful influence of prior experience on novel tasks, but also how it prevented users from exploring other methods [10]. In more recent work on knowledge transfer within the ACT* production framework [14], Singley and Anderson note that "productions [a representation of knowledge] which produce clearly inappropriate actions contribute to poor initial performance on a transfer task but are quickly weeded out. Productions which produce actions which are merely nonoptimal, however, are more difficult to detect and persist for longer periods" (p. 137).

These experimental results have been validated in our studies of real-world CAD usage. In data collected during an ethnographic study [2], we found that despite formal training in CAD and several years of experience, many users missed opportunities to use efficient methods in routine CAD tasks. Although the users had mastered basic commands and could complete their tasks, they used inefficient strategies that resulted in an increase of low-level inputs and errors, both of which led to an increase in execution time [1, 4, 5]. Furthermore, many of these inefficient methods appeared to be directly transferred from manual drafting methods that still worked in the new medium.

The above empirical findings point to the difficulty that users have in moving from a sufficient use to a more efficient use of computer applications. We have argued that neither good interface design nor experience guarantee that this transition will occur [4]. Instead, we have come to believe that *strategic knowledge* holds the key to efficient usage and that this knowledge must be explicitly taught. This contrasts with traditional training, which focuses on the use of commands to complete simple tasks. We hypothesized that if users were taught to use efficient strategies in the context of tasks, they would be able to recognize opportunities to use them also in new tasks. To test this hypothesis, we developed and implemented a new approach to training called the Strategic Use of CAD. This paper presents initial findings that demonstrate the effects of this approach on the ability of users to recognize opportunities for using efficient CAD strategies.

We begin by describing the nature of strategic knowledge in the context of complex computer tools. We then explain the role it played in the design of the Strategic Use of CAD course offered in the School of Architecture at Carnegie

CHI '99 Pittsburgh PA USA
Copyright ACM 1999 0-201-48559-1/99/05...$5.00

Mellon University (CMU). The course content and structure were based on a two-pronged approach: *Learning to See* taught students how to recognize opportunities to use efficient strategies; *Learning to Do* taught students how to implement these strategies to complete a task. We present the results of a pilot experiment to explore the efficacy of this approach. We conclude by discussing the ramifications of these results for the design of training and future experiments.

THE NATURE OF STRATEGIC KNOWLEDGE

Complex computer applications such as CAD systems typically offer more than one way to perform a given task. Consider the task of drawing the three shapes shown in Figure 1. One way of doing this is to draw the outline of *all* the outer shapes, then to outline the inner shapes, and finally to pattern *all* the inner-shapes individually (Figure 1A). An alternate way to do the same task, shown in Figure 1B, is to draw all the elements of the first shape (Detail), to group these elements using an appropriate command like FENCE (Aggregate), and then to make multiple copies of the fence contents to create the other shapes (Manipulate).

Both of these methods allow a user to complete the task. We have called such *non-obligatory* and *goal-directed* methods strategies [4, 13]. The Sequence-by-Operation and Detail-Aggregate-Manipulate (DAM) methods illustrated in Figure 1 are prime examples of CAD strategies. When we refer to *strategic knowledge,* we refer to knowledge of such alternate methods and of how to choose between them.

To arrive at a precise understanding of the knowledge required to use various strategies, we modeled them using GOMS [6]. The DAM strategy, for example, was represented by a combination of a selection rule and a method as shown below:

```
IF   CURRENT-TASK IS DRAW MULTIPLE OBJECTS
     AND COMPOSITION IS REPLICATION
THEN ACCOMPLISH GOAL: DETAIL-AGGREGATE- MANIPULATE.

METHOD FOR GOAL: DETAIL-AGGREGATE-MANIPULATE.
    STEP 1. ACCOMPLISH GOAL: DETAIL.
    STEP 2. ACCOMPLISH GOAL: AGGREGATE.
    STEP 3. ACCOMPLISH GOAL: MANIPULATE.
    STEP 4. RETURN WITH GOAL ACCOMPLISHED.
```

The selection rule connects the nature of the task (replication) to a strategy label (Detail-Aggregate-Manipulate); the method decomposes the label into subgoals (Detail, Aggregate, Manipulate). This representation shows that the knowledge required to use the DAM strategy is fairly abstract in that it does not include knowledge of explicit CAD commands. In fact, we have shown that strategies such as DAM are useful in other applications such as word processors and spreadsheets [4]. We therefore call them *abstract strategies.* Abstract strategies such as DAM are used in combination with *concrete strategies,* which capture the knowledge of how to select and execute appropriate commands in a particular application in order to actually execute a task. Concrete strategies are important because CAD systems provide in many cases more than one command to achieve a result.

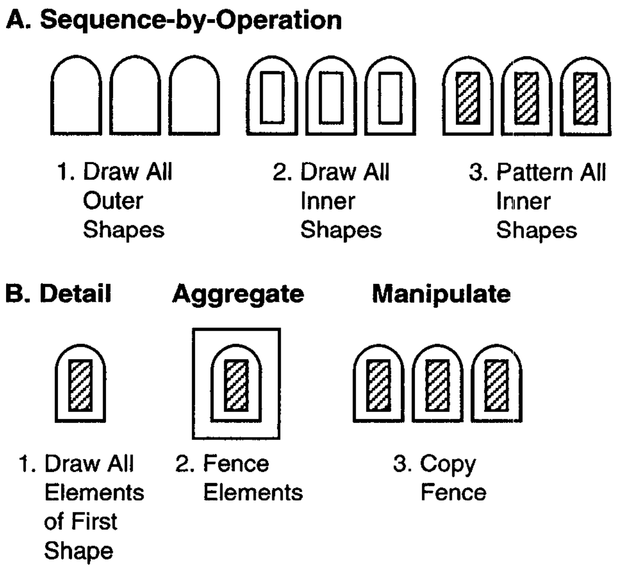


Figure 1. A comparison of the steps required for the Sequence-by-Operation strategy and the Detail-Aggregate-Manipulate (DAM) strategy.

The GOMS analysis of a real-world task using the DAM strategy estimated a reduction of almost 40% in execution time when compared to the Sequence-by-Operation strategy [1, 4]. The strategy furthermore reduced low-level mouse interactions, which tend to be error-prone. Such performance variables are important to users [1, 4]; the DAM strategy is therefore more efficient than the Sequence-by-Operation strategy for the analyzed task.

The main reason why the DAM strategy is more efficient than the Sequence-by-Operation strategy is that it exploits the iterative power of the computer through aggregation commands (such as FENCE). By aggregating the first shape and then applying the copy operation to the aggregate, the *computer* performs the iterative actions of copying each element. In contrast, the Sequence-by-Operation strategy requires the *user* to perform the iterative task of drawing and patterning multiple copies of the same shape, which leads to time-consuming and error-prone actions. Such analyses of strategic knowledge helped us understand the sources and consequences of different strategies; they led us in the end to the formulation of a general theory of efficient strategies (see [1, 5] for an explication of this theory).

Using the theory of efficient strategies, we identified other strategies that exploit the power of iteration provided by computers. Figure 2A shows the Aggregate-Drop-Modify-Strategy (ADM), which is useful for the modification of multiple elements with exceptions. Figure 2B shows how

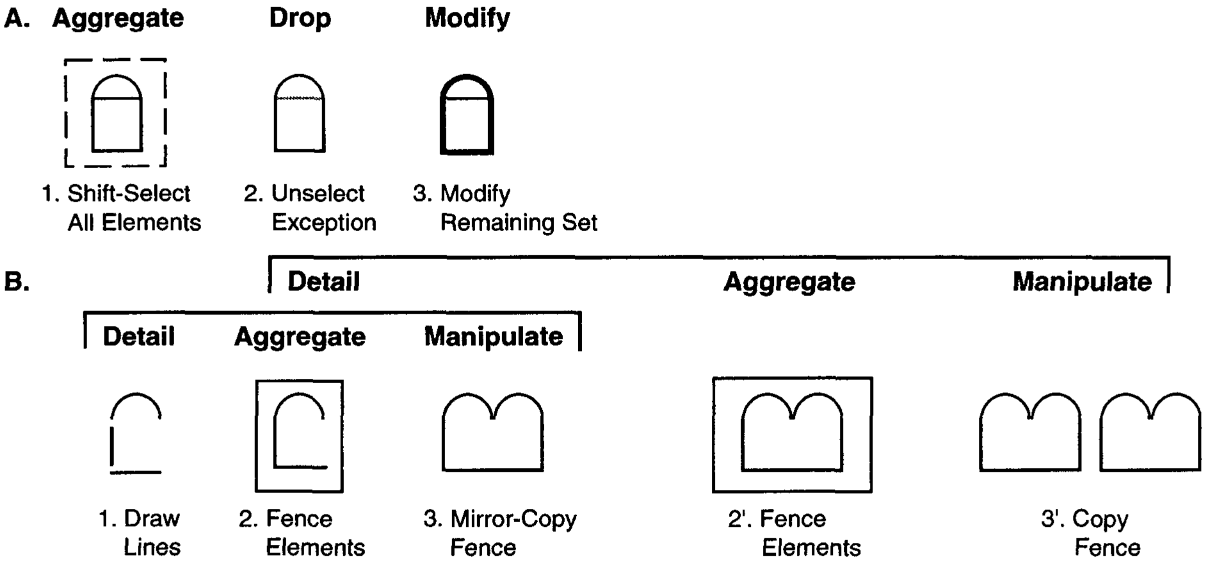


Figure 2. The Aggregate-Drop-Modify strategy (ADM) is another example of an aggregation strategy that delegates iteration to the computer. Aggregation strategies such as DAM can be used recursively to create complex drawings (B).

the DAM strategy described earlier can be used recursively to create complex drawings that have more than one level of symmetry or replication. Because the DAM and ADM strategies exploit iteration through aggregation tools, we have called them *aggregation strategies* (see [1, 4, 5] for a description of other aggregation strategies). The theory furthermore guided us in the identification of other computer powers such as precision and of strategies to exploit them [1, 5].

Several experimental and real-world studies have shown that strategic knowledge is not easily acquired by computer users in command-based training or through many years of experience [4, 7, 8, 11]. Furthermore, we have argued that even well-designed interfaces cannot express this knowledge unambiguously [4]. We therefore hypothesized that the most effective way to make users efficient is to teach them strategic knowledge explicitly. The GOMS representation and the theory of efficient strategies led directly to the design of the Strategic Use of CAD course.

THE STRATEGIC USE OF CAD COURSE

The goal of the Strategic Use of CAD course was to teach architectural graduate students how to use MicroStation™, a complex CAD package. While the students were taught how to use a wide range of commands to complete complex drawings, the course focused on how to use efficient abstract strategies that are useful in any CAD system.

Course Content

The course concentrated on teaching six classes of strategies. These strategies exploited the powers of calculation, precision, iteration, propagation, visualization and generation provided by CAD systems in various degrees. For example, an abstract precision strategy is to set up the grid units of a drawing and appropriate snap locks *before* attempting to draw any object. A concrete precision strategy specifies how to choose between various snap locks and how to use them in combination with other commands (see [1, 5] for a discussion of strategies to exploit the other powers). Figure 3 shows how the presentation of abstract and concrete strategic knowledge was interleaved throughout the course, which moved from 2-D and 3-D drawing to rendering and finally macro development.

Individual classes were held in a computer cluster. They were structured around the demonstration of abstract and concrete strategies using MicroStation™. Demonstrations were supported by lectures focussing on the underlying concepts or background knowledge, such as the mathematics of b-spline curves. Students were allowed to choose their own midterm and final projects (upon approval of the instructor) to demonstrate their understanding and mastery of efficient strategies.

Pedagogical Approach

The GOMS representation of strategies discussed earlier made salient two important knowledge components necessary to use a strategy. The selection rule suggested that a student must learn to recognize when a task offers an opportunity to use a particular strategy. We call the process of acquiring this knowledge *Learning to See*. The method component suggested that a student must learn how to implement the strategy by decomposing the task into temporally-ordered subgoals. We call the process of acquiring this knowledge *Learning to Do*.

We realized early on that it was not possible to gauge whether a student actually used an efficient strategy just by inspecting a completed drawing. For example, by looking

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Lectures	Overview Motivation	Cognitive modeling Theoretical frame-work	Action sequences Euclidean transform.					Iteration (summary)	Orthogr. views		Curves Surfaces			Generative systems	
Abstract Strategies	Pre-test DAM (as illustration)	*Calculation* *Precision*	*Iteration* Recursion	LAMM		*Propagation* Reuse Data stratification		Other aggregation strategies (AM,ADM, AMM,etc.)	*Visualization* Multiple views					*Generation* Programming	Post-test Discussion
Concrete Strategies	Getting started	Grid set-up Snap modes Basic constructions	Aggregation Manipulation	Modification	Advanced constr./modification	Cells Levels	Patterns Plotting	Advanced aggregation	3D display Rendering	3D constructions Depth control	Advanced 3D constructions	Surface modification Output from 3D models	Misc. commands	Macros	

Figure 3. A time-line for the semester-long Strategic Use of CAD course showing the interleaving of abstract and concrete strategic training. Powers of CAD are italicized, and aggregation strategies are in upper case.

only at step 3 in Figure 1A, it is not possible for an instructor to know which strategy was used to draw the three arched figures. While designing exercises to help students in the process of Learning to See and Learning to Do, we therefore asked students to describe in words and figures the steps they plan to use in completing their homework exercises, mid-term and final projects. We called these descriptions *action sequences.* The steps shown in Figures 1 and 2 are examples of very simple action sequences.

Previous research has shown that users have substantial difficulty in recognizing and using efficient abstract strategies [4, 7, 11]. We therefore required students to describe the abstract strategies they intended to use explicitly and completely in their action sequences. Students were not required to specify the concrete strategies (command selection and execution) they planned to use. Figure 4 shows the various types of knowledge taught by the Learning to See and Learning to Do components. The gray cells represent the knowledge captured in action sequences.

	Abstract Strategies	Concrete Strategies
Learning to See	Knowledge to see the structure in a task and select a strategy	Knowledge to select appropriate commands
Learning to Do	Knowledge to decompose a task based on a strategy	Knowledge to execute and combine commands

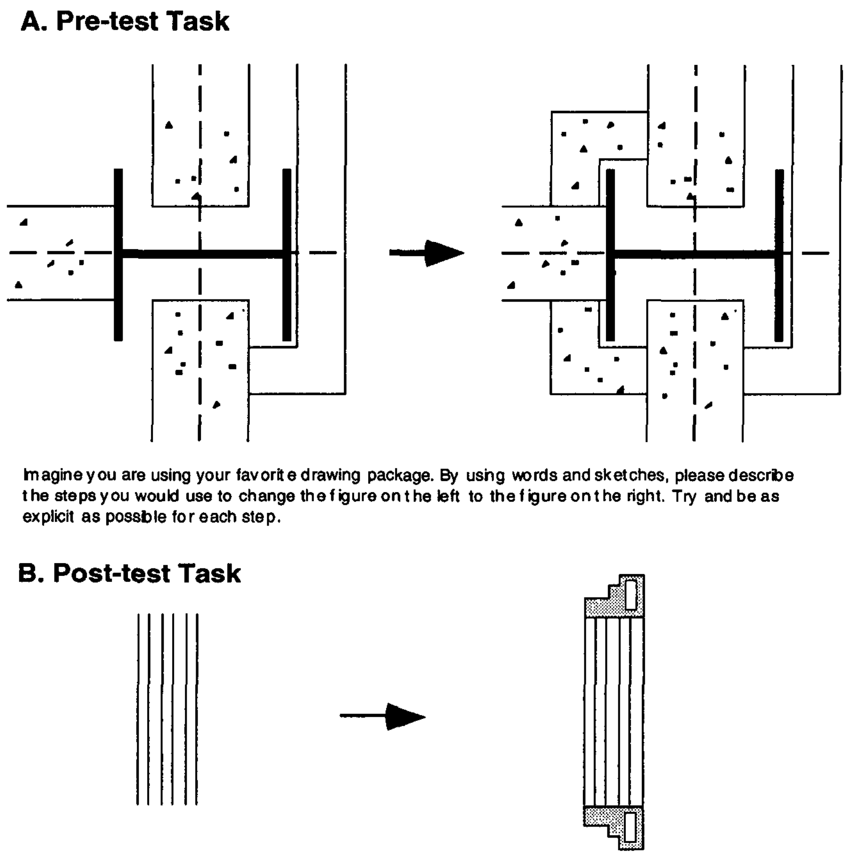
Figure 4. The types of knowledge required in the Learning to See and Leaning to Do components for abstract and concrete strategies. The gray cells represent the knowledge expressed in action sequences.

Action sequences are pedagogically important because they make the *process* public and therefore open for inspection. Furthermore, it allows the instructor to provide timely feedback to students. To motivate students to think about the process as well as their final products, they were informed that their final grade would depend on their action sequences as well as the quality of the final drawings. This approach is a significant departure from traditional training approaches, which concentrate on giving feedback mainly on the final product. To understand the efficacy of this approach, we tested its effect in a pilot experiment.

EXPERIMENTAL DESIGN

The Strategic Use of CAD course was offered in the graduate program in the School of Architecture at CMU in the spring semester of 1998; it was open to advanced undergraduates. Eight students (seven graduates and one undergraduate) took the course for credit. The small size of the class is typical of graduate-level courses offered at the university. Six of the students were architecture majors and two of them were HCI graduate students. While all the students took part in the experiment, one student had to be dropped from the analysis as she had attended some lectures when the course was offered in the previous year, which invalidated her pre-test scores. Taking part in the experiment did not affect the students' course grade.

The experiment consisted of a pre-test conducted on the first day of classes and a post-test conducted on the last day of classes. The goal of the experiment was to investigate how the course affected the students' ability to decompose a drawing task so that efficient CAD strategies could be applied. As the experiment was not a test of command knowledge, the students were asked to describe with words and pictures the steps they would take to perform a drawing



A. Pre-test Task

Imagine you are using your favorite drawing package. By using words and sketches, please describe the steps you would use to change the figure on the left to the figure on the right. Try and be as explicit as possible for each step.

B. Post-test Task

Imagine you are using MicroStation. By using words and sketches, please describe the steps you would use to change the figure on the left to the figure on the right. Try and be as explicit as possible for each step.

Figure 5. A pre-test task (A) and its isomorph in the post-test (B). The pre-test task is identical to a task that a user performed inefficiently in our ethnographic study (see [1, 4, 9] for a detailed analysis of this task). This task requires an application of the DAM strategy described in Figure 1B. The drawings presented here are half the size of the original tasks presented to the students.

task when using their favorite CAD or drawing application. Students were instructed to use 8-1/2"x11" sheets of unruled paper provided by us to describe the steps. Students could choose their own writing instruments to perform the experiment.

The pre- and post-test each consisted of three drawing tasks[1]. The first task required the DAM strategy, the second required the ADM strategy, and the third task was designed to test if the students could use the DAM strategy recursively. The post-test drawing tasks were isomorphs of those in the pre-test. Figure 5 shows an example of a task and instruction in the pre-test, and an isomorph of the same task in the post-test. These tasks were similar to those observed in our ethnographic study [2] where real-world architects missed opportunities to use aggregation strategies. The students were given no time limit, but all completed the tasks within the 80 minute time slot of the class.

DATA ENCODING

We established the following criteria for the successful application of each strategy.

1. The DAM strategy was used successfully if all details were completed in the first drawing before it was aggregated and mirror-copied to create any subsequent copies. The Detail stage itself did not have to be correct, but any attempt to draw elements in the copy was considered a non-use of the strategy.

[1] A fourth task, which tested knowledge of a strategy called LAMM [4], had to be dropped from the analysis because the difference between the before and after drawings in the pre-test was too subtle; this caused five of the seven students to misunderstand the pre-test task.

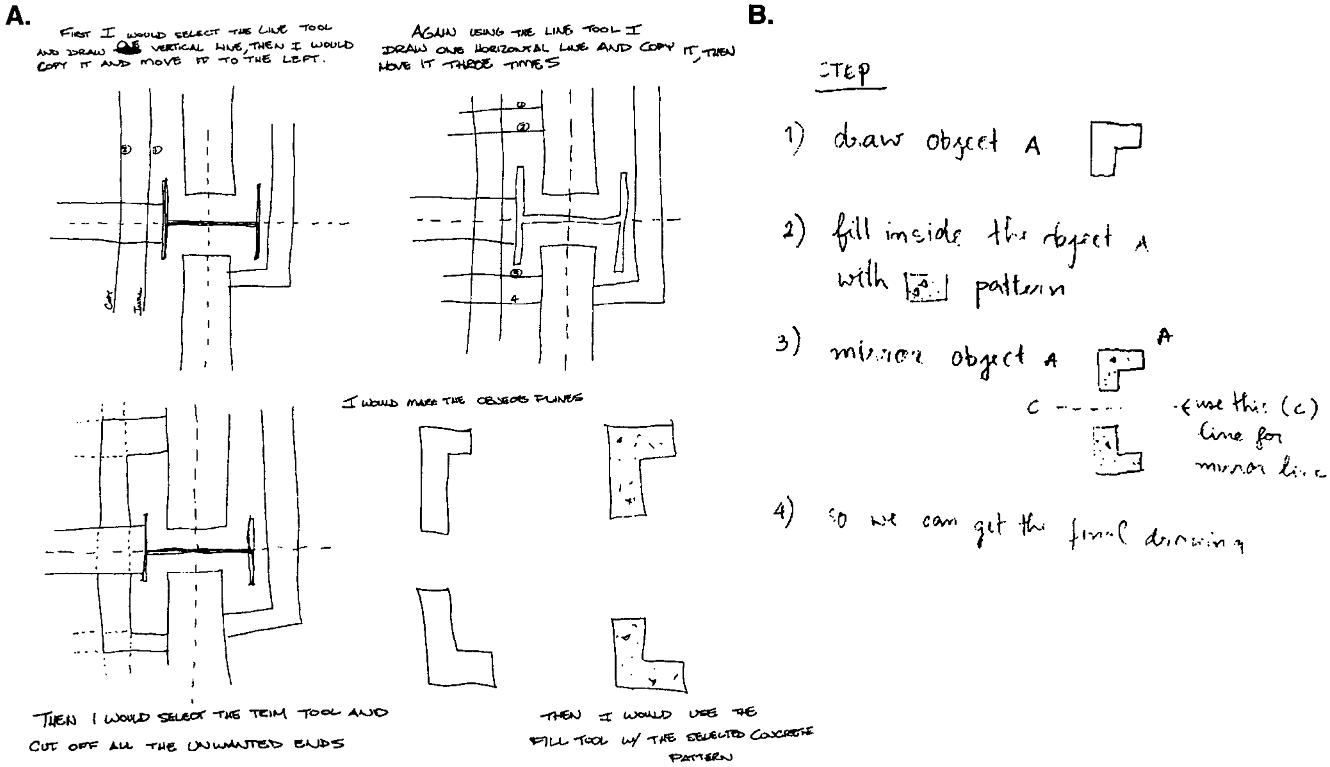

A.
FIRST I WOULD SELECT THE LINE TOOL AND DRAW ONE VERTICAL LINE, THEN I WOULD COPY IT AND MOVE IT TO THE LEFT.
AGAIN USING THE LINE TOOL I DRAW ONE HORIZONTAL LINE AND COPY IT, THEN MOVE IT THREE TIMES
I WOULD MAKE THE OBJECTS PLINES
THEN I WOULD SELECT THE TRIM TOOL AND CUT OFF ALL THE UNWANTED ENDS
THEN I WOULD USE THE FILL TOOL W/ THE SELECTED CONCRETE PATTERN
B.
STEP
1) draw object A
2) fill inside the object A with pattern
3) mirror object A
A
C -----
use this (c) line for mirror line
4) so we can get the final drawing

Figure 6. Two attempts by students in the experiment to do the pre-test task (described in Figure 5A) requiring the DAM strategy. A score of 0 was given to the attempt shown in (A), and a score of 1 was given to the attempt shown in (B).

2. The ADM strategy was used successfully if there was any attempt to aggregate elements in such a way as to exclude the exceptions. Attempts to modify each element individually to avoid the exceptions was considered a non-use of this strategy. The type of commands to perform the aggregation was irrelevant.

3. The recursive use of the DAM strategy was considered successful if the strategy was used in at least two levels as shown in Figure 2B.

Based on the above criteria, a score of 1 was given if the appropriate strategy was used, and a score of 0 if the strategy was not used. This coarse-grained analysis reflected the main goal of the experiment, which was not so much to see if the students could execute commands or draw accurately, but rather to determine if the students could recognize opportunities to use abstract strategies and plan a course of action. The criteria also reflected the temporal aspect of the aggregation strategies (do A before B). Our GOMS analysis of similar tasks has shown that when such temporal criteria are violated, low-level interactions proliferate leading to many errors and an increase in execution time [1, 4, 5].

Figure 6A shows an example of how a student decomposed the task requiring the DAM strategy described in Figure 1B. The steps show how he planned to draw both the "L" shapes using lines similar to those that would be drawn using a T-square in manual drafting. This is similar to the Sequence-by-Operation strategy shown in Figure 1A. As the student did not recognize the opportunity to use the DAM strategy, he was given a score of 0 for that task. Figure 6B shows how another student attempted the same pre-test task. Here she correctly saw the symmetrical composition of the task and demonstrated the use of the DAM strategy by first drawing all the details in the first "L" shape, and then aggregating and mirroring the shape to create the copy. She was therefore given a score of 1 for that task.

DATA ANALYSIS

As shown in Figure 7, two (S4 and S7) of the seven students demonstrated knowledge of all the strategies in the pre-test, while one student (S5) demonstrated knowledge of none. The rest of the students showed knowledge of at least one aggregation strategy, but not of all. To determine if there was an overall positive effect of the class, the Wilcoxon test[1] was performed on the data collapsed over tasks. A one-tailed test showed a significant improvement of the post-test scores over the pre-test scores ($p<0.05$).

Of the 11 cases where students did not exhibit knowledge of a strategy in the pre-test, only 2 did not display that

[1] The Wilcoxon test is a distribution-free, non-parametric test designed for nominal data and is similar to the paired sample T-test designed for continuous data.

	DAM Task		ADM Task		Rec. DAM Task	
Studnt	Pre	Post	Pre	Post	Pre	Post
S1	0	1	1	1	0	1
S2	1	1	0	0	0	1
S3	0	1	0	0	1	1
S4	1	1	1	0	1	1
S5	0	1	0	1	0	1
S6	0	1	1	1	0	1
S7	1	1	1	1	1	1
Total	**3**	**7**	**4**	**4**	**3**	**7**

Figure 7. The pre- and post-test scores for seven students performing three tasks. The scores represent the unsuccessful (0) and successful (1) use of aggregation strategies.

knowledge in the post-test. These cases show up in the table as a pair of 0s in the pre- and post-test scores for the same task (S2 and S3 in the ADM task). Furthermore, there was only one case where a strategy known by a user in the pre-test was not used in the post-test (S4 in the ADM task). It is interesting to note that all three cases occurred in the task requiring the ADM strategy. We therefore performed a more detailed analysis to determine whether the ADM strategy was taught differently from the DAM strategy.

The analysis revealed that the DAM strategy was introduced in the first week and emphasized for homework assignments throughout the course. Furthermore, the recursive DAM strategy was explicitly taught in the third week. In contrast, the ADM strategy was taught for only a small portion of the eighth week. Therefore, the DAM and ADM strategies received proportionately different times in lectures, demonstrations, and practice in homework assignments.

DISCUSSION OF RESULTS

The results provide encouraging evidence that the Strategic Use of CAD course improved the students' ability to recognize opportunities for the use of efficient strategies. While the students came into the course with mixed abilities to recognize these opportunities, the post-test scores show a far more uniform ability at the end of the course. Furthermore, with the exception of one student in one task, all students who could recognize opportunities to use strategies at the start of the course did not loose that ability at the end of the course. The course therefore helped those who did not know the strategies and did not hurt those who knew them. Because students had no discernible problems showing the steps they would use to do the pre-test task, we are confident that these results were not caused by an improved ability to describe action sequences over the semester.

Because we had carefully chosen the DAM task to be identical to the one we had observed in our ethnographic study (as shown in Figure 5A and described in [4, 9]), the study helped confirm our long-standing belief that the carry-over of manual drafting strategies to CAD was not unique to our ethnographic data. In fact, four of the seven students failed to use the DAM strategy in the pre-test while all used it in the post-test. We found the same proportion of pre-test and post-test scores in the recursive DAM task.

The three missed opportunities to use the ADM strategy and subsequent analysis suggest that students may not have learned the conditions for its use because we did not spend enough time in describing and demonstrating the ADM strategy. This has been observed in other studies that taught strategies. For example, several strategies in Shoenfeld's [12] experiments on teaching problem-solving strategies in mathematics were also not used in transfer experiments. Singley and Anderson [14] suggest that the strategies which were least used in Shoenfeld's study lacked explicit conditions for their use, and therefore did not provide the information important for their selection. Thus, if either explicit conditions for strategy selection do not exist (as in Shoenfeld's study), or sufficient time is not taken to teach them (as seems to be the case in our course), students may fail to use the strategies. In the next iteration of the course, we intend to provide explicit conditional information for the ADM strategy.

Clearly, our pilot experiment is limited by the small number of students that registered for the graduate-level course, a constraint beyond our control. In addition, the pilot experiment was limited by the absence of a control condition. This could lead to the rival hypothesis that the observed improvements in recognizing opportunities to use strategies could have occurred just by virtue of learning how to use a CAD system for a semester, irrespective of the training approach. However, we have evidence against such a claim. First, although one of the students in the experiment had taken a command-based AutoCAD™ (another sophisticated CAD system) course at CMU in the preceding semester, he still used the inefficient Sequence-by-Operation strategy in the pre-test (as shown in Figure 6A). While he had mastered the use of aggregation commands in AutoCAD™, the command-based course had not helped him to recognize opportunities to use them with efficient strategies. Second, our ethnographic study also demonstrates the ineffectiveness of command-based training to make users efficient: despite formal training through their CAD vendor, the users in the study exhibited similar forms of inefficient usage.

Both limitations of this pilot experiment (the small number of participants and the lack of a control condition) will be addressed in our current research that aims to design and test a strategy-based course for other complex applications (such as word processors and spreadsheets). This new course will be taught to approximately 130 incoming freshmen next year with the same number of students in a traditional command-based control condition.

CONCLUSION

Based on converging evidence starting from the Einstellung effect down to our own studies on experienced CAD usage, we have begun to recognize the widespread and persistent occurrence of inefficient methods and their negative effects on performance. Our research over several years has explored different approaches to counter-act this phenomenon. We have focused on the systematic identification of strategic knowledge and the dissemination of that knowledge to users. Insights from the research led directly to the design of the Strategic Use of CAD course.

The course design made two departures from conventional training. The first was to focus on the teaching of strategies in addition to commands as neither can achieve efficient usage on its own. The second departure was to make students think consciously about process before they attempted their tasks. Both relied on a pedagogical device we called action sequences.

The results of our pilot experiment suggest that the strategic approach had a positive effect on the students' ability to recognize opportunities for using efficient abstract strategies and to develop a plan to use them. Furthermore, the course did not negatively affect existing knowledge of strategies. While the experimental tasks tested the learning of abstract strategies for relatively small drawings, future experiments should investigate whether the strategic approach also helps users apply abstract strategies in the context of large complex drawings requiring deep decompositions. In addition, future research should investigate the link between abstract strategies which are general to any CAD package, and concrete strategies which are specific to a particular package. Finally, we need a systematic understanding of trade-offs between strategies because we have observed situations where the choice between competing strategies is not as clear-cut as those we have discussed in this paper.

Although we have explored other ways to disseminate strategic knowledge, such as active assistance as well as better management and peer interaction [1, 2, 3], we believe that there can be no replacement for the explicit training and practice of strategic knowledge to use complex computer systems. Our current research extends the strategic approach to the teaching of other complex computer applications. We intend to study if users are able to transfer abstract strategies across applications and if they retain this knowledge over time. The hope is that this research will provide a way to counter-act the persistence of inefficient usage, which has plagued modern computer usage for many years.

ACKNOWLEDGMENTS

This research was supported by the National Science Foundation, Award# IRI-9457628 and EIA-9812607. 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 NSF or the U. S. Government. The authors acknowledge the contributions of students who took part in the experiments, in addition to A. Corbett, M. Lovett, and G. Vallabha. Bentley Systems, Inc. supplied the academic edition of MicroStation™.

REFERENCES

1. Bhavnani, S. K. *How Architects Draw with Computers: A Cognitive Analysis of Real-World CAD Interactions*, unpublished Ph.D. dissertation, 1998, Carnegie Mellon University, Pittsburgh.
2. Bhavnani, S.K., Flemming, U., Forsythe, D.E., Garrett, J.H., Shaw, D.S., and Tsai, A. CAD Usage in an Architectural Office: From Observations to Active Assistance. *Automation in Construction* 5 (1996), 243-255.
3. Bhavnani, S.K., and John, B.E. Exploring the Unrealized Potential of Computer-Aided Drafting. *Proceedings of CHI'96* (1996), 332-339.
4. Bhavnani, S.K., and John, B.E. From Sufficient to Efficient Usage: An Analysis of Strategic Knowledge. *Proceedings of CHI'97* (1997), 91-98.
5. Bhavnani, S.K., and John, B.E. Delegation and Circumvention: Two Faces of Efficiency. *Proceedings of CHI'98* (1998), 273-280.
6. Card, S.K., Moran, T.P., and Newell, A. *The Psychology of Human-Computer Interaction.* Hillsdale, NJ: Lawrence Erlbaum Associates, 1983.
7. Cragg, P.B. and King, M. Spreadsheet Modeling Abuse: An Opportunity for OR? *Journal of the Operational Research Society* 44 (1993), 743-752.
8. 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.
9. Flemming, U., Bhavnani, S.K., and John, B.E. Mismatched Metaphor: User vs. System Model in Computer-Aided Drafting. *Design Studies* 18 (1997), 349-368.
10. Luchins, A. S. Mechanization in Problem Solving. *Psychological Monographs*, (1942), 54 (248).
11. 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.
12. Schoenfeld, A. H. Mathematical Problem Solving. New York: Academic Press, 1985.
13. Siegler, R.S., and Jenkins, E. *How Children Discover New Strategies*. Lawrence Erlbaum Associates, New Jersey, 1989.
14. Singley, M.K., and Anderson, J. R. *The Transfer of Cognitive Skill.* Harvard University Press, Cambridge, Massachusetts, 1989.