Using Cognitive Models of Learning in Instructional Design

Dana S. Kay
Graduate School of Education
University of California, Berkeley

The primary goal of this research was to facilitate learning by applying a model of the changes in knowledge representations that occur with increased computer experience to instructional design. The model proposes that users' knowledge representations of computer systems evolve from being organized according to preconceptions to being organized according to the goals, plans and rules for using the system. The model was implemented in an on-line training system. It was proposed that using a model of the natural evolution of knowledge to organize the presentation of information in the training would create a better fit between learning process and learning materials and thus, facilitate learning. The model-based training system was compared to four other training systems that implemented other methods of organization that are currently used in research and industry. The results illustrated only partial validation for the model. That is, the training based directly on the model did not lead to the best peformance, but also did not lead to the worst. The best performance was observed for instructions that were organized according to tasks and the worst performance was observed for instructions that were organized according to categories of commands. It is proposed that (1) using models of learning in instructional design requires more than mapping the learning assumptions onto an instructional setting and (2) instructional design provides an important method for testing the assumptions of the learning model.

There has been a substantial amount of attention given to the investigation of the acquisition of knowledge. From this research has grown an interest in the application of cognitive models of the student to instructional design. That is, researchers have begun to examine the benefits of using knowledge-based models of learning to guide the organization of instructional information. In a recent study, Kay and Black (1985) presented a model that traced the changes in the content and organization of knowledge that occur with the acquisition of text-editing knowledge. The present study compares the performance that results from instruction that is based upon this model to the performance that results from instruction that is based upon other methods of organizing computer knowledge.

By combining the results of three studies that examined the changes in text-editing knowledge representations, Kay and Black (1985) proposed a four phase model of the evolution of knowledge as one proceeds from naive to expert computer user. In this model, phases represent snapshots of the mental structures that exist at varying levels of ability. Phase One of this model describes the knowledge representations that exist before the learning process begins. In this phase, knowledge is organized according to preconceptions about the terminology that will later refer to text-editing commands. In Phase Two, users' knowledge representations begin to reflect computer-related information. Because one of the first things that users learn is the goals that the commands accomplish, the new organization of the commands is based upon the relationships between the commands and the relevant goals that they accomplish.

Having acquired the basic text-editing goals and commands, users begin to note that there are certain commands that are frequently used together in accomplishing a goal. It is this realization that leads to another reorganization of knowledge in which actions that were organized separately in Phase Two are now combined into the plan-based representations of Phase Three. The final phase of the model describes the knowledge representations for users who are experts in text-editing. The representation proposed for this phase completes the acquisition process (aside from minor forms of tuning that will continually take place) and is similar to the GOMS (Card, Moran & Newell, 1983) account of text-editing experts. In this phase, knowledge is reorganized to incorporate (1) compound plans that are composed of the simple plans of Phase Three and (2) selection conditions for choosing the most appropriate plan for accomplishing a given goal.

Computer Instruction

Scenario Machines. There are many forms of computer instruction. One form, that has been given attention by Carroll and his colleagues, is training systems. These systems are designed to encourage users to learn by exploring the functions available in the system. Thus, users are actively engaged in the learning process. The "training wheels" interface (Carroll & Carrithers, 1984a) is a training system in which new users are blocked from accessing advanced system functions and from committing typical user errors. The motivation behind this design is to encourage "discovery learning," while sparing users the confusion and frustration that often result when new users wander into the depths of complex computer systems. By blocking off certain system states, the set of possible actions and consequences of these actions is constrained. Thus, users are free to actively participate in and control their learning.

The "Scenario Machine" (Carroll & Kay, 1985; McKendree, Schorno, & Carroll, 1985) is a more restricting version of the "training wheels" interface. In this design, not only are advanced functions and user errors blocked, but the user is directed through a single scenario that represents the basic functions of the system. The use of this type of system design provides researchers with knowledge of what the user will be required to do and thus, we can better control our investigation of user learning. The current experiment uses the Scenario Machine design as a vehicle for examining the application of cognitive models of student knowledge representations to the design of computer instruction.

Current methods of computer instruction. If we survey the commercial forms of computer instruction that are currently used, there are two prominent methods of organization that are used -- command-oriented and task-oriented. Command-oriented instruction is designed by organizing the computer information according to categories of commands that can be used in the system. The problem with this type of instructional organization is that the order of presentation is guided by categories of commands, rather than by the order in which the commands are encountered as part of an editing session. Task-oriented instruction illustrates an attempt to find a better match between order of presentation and order of use in an editing session. A task-orientation presents the editing commands according to plans for accomplishing editing tasks. Recent research has improved upon the commercial task-oriented instruction and proven it to be quite effective (Carroll, 1984; Carroll & Carrithers, 1984).

Example-based instruction is a method of instruction that has not been implemented in commercial systems. However, this type of instruction has been examined within the context of research on teaching people how to use computers and has been shown to be an important part of learning (Rissland, Valvarce, & Ashley, 1984). Example-based instruction is similar to the task-oriented instruction in that the command information is presented in accordance with its use in computer sessions or tasks. However, using only examples does not provide the user with conceptual knowledge of the system and thus, the user may have difficulty recovering from errors due to the use of incorrect commands.

Current Experiment Closer examination of the current methods of organization shows that none of these methods is based directly upon the changes in the content and organization of user representations as presented by the Kay and Black four phase model. Therefore, an on-line training system was designed for this experiment using the four phase model to guide the organization of information. The information presented in the training system is for a command-based, experimental database system. In addition to the four phase model training system, training systems implementing the other methods of instruction were also designed for the database system. By comparing the performance (i.e. time to complete tasks and errors committed) that results from using each of these training systems, we can assess each of these methods of instruction. The basic hypothesis is that for the model-based training, there will exist a better fit between the learning process and learning materials. As a result, it should be easier for users to acquire computer information. If using this model facilitates learning, then in addition to supporting the use of cognitive models in instructional design, there will also be further validation of the four phase model.

Method

Subjects

Forty Yale undergraduates participated in this study. All participants had little or no computer experience. That is, they had either never used a computer or had used the computer a few times for word processing purposes only.

Materials

Database System. A simple experimental database system was designed for the study. The system was a command-based, interactive system that contained information about popular songs. There were fifteen commands that could be used in the system and five basic tasks that sequences of these commands could accomplish. The system kept track of the keystrokes entered and the reaction time between keystrokes.

Training Systems. In addition to the basic database system, five training systems were also designed. These systems taught subjects how to use the database system by explaining the commands used to accomplish the database system tasks. The presentation of the commands within the training systems was dependent upon the method of instruction implemented in that training system. Four of the training systems were based upon the methods of organization previously described -- command-oriented, task-oriented, example-based and four phase model. The fifth training system was used as a control condition for the four phase model. Because it is believed that the ordering of the phases is important to the learning process, a training system was designed that organized the four phases in reverse order.

There were two parts to the design of these systems. In the first part, each of the five methods of organization was mapped onto the fifteen commands used in the database system. In the second part, these mappings were used to design a method of organization that generalized across the mappings and resulted in an ordering for the commands that could be applied in all the training systems. The latter part of this design process was used so that the same exercises could be used regardless of the method of instruction.

There were five parts to the training systems. The first part of the training was the same for all five training system conditions and consisted of two parts, an introduction to the database system and a description of the Quit command. This command was described separately because it is used throughout the system, regardless of the task.

In Part 2 of training, the commands used for listing, printing, and sorting the database were presented together. In Part 3, the commands for adding songs to the database were presented. In Part 4, the commands for searching in the database were presented and finally, in Part 5, the commands for changing the database were presented together. Because of the similarity in command relationships, the systems could be designed so that regardless of instructional organization, the same exercises were presented after each section.

The exercises were the same for all five training systems. In performing these exercises, subjects were taken into a modified version of the database system that had the same screen design as the full database system. However, the exercise environment differed in its response to subject input. As described previously, the system used a Scenario Machine design (Carroll and Kay, 1985) in which the system stalls until the correct command (that required for the task at hand) is input by the subject. That is, for each exercise, there is a fixed scenario for accomplishing the task and unless the subject inputs the correct command for the scenario, the system beeps and prompts the subject to input a command. This design was used to keep subjects from entering incorrect commands and getting lost in the system.

Procedure

Learning Phase. The learning phase took place on the first day of the experiment. Subjects were told that they would be following an on-line description of how to use the database system and then after each part of the training, they would demonstrate their understanding of the system by working through a series of exercises related to what they had just learned. They were also told that for each of the exercises, they would not be required to use any command that they had not been previously introduced to and that if they input a wrong command, the system would beep and then prompt them to input another command.

Subjects were also given a manual that contained the same information that was presented on the computer. They were informed that they could use this manual while working on the exercises, but that they should let the on-line system guide their training. The experimenter informed the subjects that they were to work on their own, but that if they felt that they were in trouble, they could call for help. In addition, subjects were informed that their interaction with the computer would be video-taped.

When the subjects completed the training tasks, a computer concepts test was administered to assess conceptual understanding of the database system. This test consisted of nine questions. Some of these questions asked users "how " to accomplish certain database tasks. Other questions asked subjects about the functions associated with certain database commands. Two versions of this test were used. One version was administered half way through the study (after training) and the other was given at the end of the study (after transfer).

Transfer Phase. The transfer phase took place on the second day of the study. Subjects were told that they would be using the database system again. However, in this part of the study, they would not be using the training system that guarded them against errors, but rather, they were using the complete database system. Subjects were given the manual that they had used the day before and a booklet of 12 transfer exercises of varying levels of difficulty. Four of the exercises were direct mappings from the training exercises that were part of the training phase. Four of the exercises were simple extensions from the training exercises (i.e. they combined two of the exercises presented in training). The last four were more complex exercises (i.e. they combined more than two of the exercises presented in training). Again subjects were told that their interaction with the computer would be video-taped and that they were to work on their own, but could call for help if they were in trouble.

After completing the transfer tasks, subjects were given another computer concepts test about the system. In this test, some of the questions were repeated from the training test and some of the questions were new. Regardless, the questions asked "how" to accomplish tasks in the system and "what" tasks certain combinations of commands accomplished.

Results

Performance Measures

Each time subjects used the database, five performance measures were recorded. The training systems and the database system recorded the time to complete the exercises performed while using the systems. An overall time measure for each phase was also collected by totalling across the exercise times. In addition to the time measures, two error measures were also recorded. One type of error is a "related" error in which the subjects uses an incorrect command that is related to the correct command. The second type of error is an "unrelated" error in which the command input is not related to the correct command. In most cases, the "unrelated" errors result when the user appears to be inputting random commands.

The fourth performance measure is the number of times that the subject requested aid from the experimenter (experimenter interventions). The range for this measure was zero to four interventions. The fifth performance measure is the score on the computer concepts test. This score is a number (0-9) representing the number of correct answers to the questions asked.

Performance during learning

Table 1 presents the relevant data for the learning phase. The first row of the table presents the mean time to complete training. The results of this analysis showed that the five training systems differed in their time to complete the learning exercises $F(4,35) = 4.4, p < .01$). A follow-up analysis showed that (1) the task-oriented training training system took significantly less time to complete the exercises than the four phase model, the reverse phase model and the command-oriented training systems at the .05 level and (2) the example-based training system was significantly faster than the command-oriented training system.

As mentioned previously, in addition to the reaction time measures, other performance measures were also collected -- "related" errors, "unrelated" errors, experimenter interventions, and score on the computer concepts test. The results for these measures during training are presented in the second through fifth rows in Table 1. A multivariate analysis of variance showed a significant main effect for the type of instruction used (Hotellings trace criterion, $F(16$[1]$, 102) = 2.2, p < .01$).

To localize these effects, univariate analyses of variance for the individual measures were used. A significant effect for type of training system was present for the number of "related" errors ($F(4,30) = 3.17, p < .05$) and for the score on the computer concepts test $F(4,30) = 3.58, p < .025$). Further analysis of these differences showed that for the "related" errors, the command-oriented training system lead to significantly more errors that the other four training systems. It appears that subjects using the command-oriented training system had to try more commands before finally using the correct command. In addition, the analysis of score on the computer concepts test showed that the example-based training system had significantly poorer performance on the test than the task-oriented, four phase and reverse phase training systems. Thus, although this training system lead to faster performance than some of the other training systems, the conceptual knowledge of the subjects using this training system was not as good as the knowledge extracted from the other training systems.

Table 1

Performance for Learning Phase

	Training System Treatments				
	Command	Task	Example	Four Phase	Reverse Phase
Total Time (in mins)	37.462	20.376	26.054	27.541	30.265
Related Errors	60.375	11.125	29.000	30.000	31.375
Unrelated Errors	50.250	9.875	13.875	14.625	23.875
Inter-ventions	1.250	0.500	1.125	0.750	1.625
Concepts Test	6.563	7.188	6.125	7.438	7.188

[1]The degrees of freedom in this analysis are smaller than expectedbecause an analysis of covariance was used. The covariates used and the results of these covariates will not be discussed in this paper due to lack of significant effects and space constraints.

Table 2

Performance for Transfer Phase

	Training System Treatments				
	Command	Task	Example	Four Phase	Reverse Phase
Total Time (in mins)	41.474	35.841	38.857	42.641	44.923
Related Errors	66.875	11.125	24.750	21.500	41.250
Unrelated Errors	17.000	3.000	7.375	5.125	10.250
Inter-ventions	0.375	0.000	0.125	0.103	0.250
Concepts Test	7.313	7.750	7.625	7.500	7.563

The results for the learning phase show that the command-oriented treatment took longer to complete the learning exercises and committed more errors while doing these exercises. On the other end of the continuum, the task-oriented treatment took the least amount of time. Although the example-based training system had relatively fast times and few errors, this treatment seemed to lack in conceptual knowledge of the system as is shown by the poor performance on the concepts test. It appears that using the examples to train users, eases use of the system, but does not provide users with an understanding of the system beyond the tasks at hand.

Transfer Phase

Table 2 presents the performance measures for the transfer phase. As in the learning analyses, an analysis of total task time was used first. However, there was no significant effect for training system types ($F(4,30) = .63$, ns). As in the learning phase, there were four additional performance measures recorded in addition to time to complete the transfer exercises. As can be seen from the table, the mean for experimenter interventions for the task-oriented training system was zero. Because the value for this variable can only be positive, the variance for this variable for the task-oriented training system can only be zero. Since a zero variance disturbs the homogeneity of variance for this variable, it was analyzed separately. The results of this analysis showed no significant difference between training system treatments in the number of experimenter interventions ($X^2 = 2.927$, ns).

A multivariate analysis of variance was performed on the remaining three measures. The results of this analysis showed a significant difference between the training system treatments (Hotellings trace criterion, $F(12,95) = 1.95$, $p < .05$). This main effect was localized to a significant difference in the number of "related" errors committed ($F(4,35) = 5.68$, $p < .01$). A follow-up analysis revealed that the command-oriented training system treatments committed significantly more errors that the four phase, reverse phase, and task-oriented treatments. In addition, the reverse phase treatment committed significantly more errors than the task-oriented treatment.

In the transfer phase, the results show that all subjects learned to use the system and were able to complete the exercises in roughly the same amount of time. However, when we look at the results for the errors committed, the command-oriented and reverse phase treatments committed the most errors. The system used in this study was a simple system. However, if a more complex system was used, we can imagine that committing a great deal of errors would lead to the users getting in trouble and in the end, taking longer to complete the tasks. This is an empirical question to be addressed in the future.

Discussion

The primary goal of this study was to facilitate learning and improve performance by designing instruction that is based upon an analysis of the changes in knowledge representation with the acquisition of computer knowledge. The results from the training phase of the study showed that the task-oriented and for some measures the example-based organization led to the best performance, while the command-oriented organization led to the worst. The instructions organized according to the four phase model seemed to fall in the middle with the four phase model being somewhat better than the reverse four phase organization (at least as measured by the number of errors committed during transfer). Thus, the original hypothesis that the four phase treatment would lead to the best performance is not totally supported.

Four Phase Facilitation of Learning

Utility of Phase Two. One possible explanation for the lack of support of the original hypothesis refers back to the original proposal of the model. In this presentation, Kay and Black (1985) provide several reasons for the existence of the four phases that they observed. They argued that the first phase was necessary because the only information that new users have to use in their initial attempts with a computer system are the prior knowledge associations that they bring to the computer domain. In addition, the fourth phase is necessary because past research (albeit minimal) has suggested that complex plans and selection rules are necessary for skilled computer performance (Card, Moran & Newell, 1983). The question then becomes are both the intermediate phases also necessary?

It would be difficult for users to progress directly from Phase One to Phase Four because this transition requires the use of more knowledge than human working memory is capable of holding at one time. Thus, the two intermediate phases provide a way of moving from Phase One to Phase Four within the constraints of working memory capacity. That is, these phases allow for the progression from Phase One to Phase Four to follow steps that can be handled by working memory. However, Kay and Black also suggested that it might not be necessary to have two intermediate phases. They proposed that Phase Three was a necessary precondition for Phase Four, but that Phase Two might be an artifact of the way that users currently learn to use a computer. That is, until recently, most commercial computer training materials have used a command-oriented organization and therefore, it is not surprising that novices incorporate this organization into their representation of the domain in the form of goal/action relationships.

The training system results suggest that it might indeed be the case that Phase Two is an artifact of learning materials. If we examine the task-oriented training system, we see that this system takes subjects directly from Phase One (preconceptions) to Phase Three (plans). Because of the simplicity of the database system and the plans necessary for using the system, the presentation of these plans can easily be handled within the limits of working memory. Therefore, subjects can learn these plans and begin using them immediately. In the four phase model training system, subjects are first guided through the Phase Two representation and then presented with the Phase Three plans. It is possible that because the Phase Two representation was unnecessary, this information interfered with subjects' ability to learn the plans presented later in the instructions and performance was hampered.

Process + Representation. Another possible explanation pertains to the method by which the four phase model was applied to the training system design. Recall that one of the original goals of the four phase model was to describe the changes in the content and organization of users' knowledge representations. However, in applying the four phase model to instructional design, the organization was focussed on the changes in knowledge representation and not on the processes that lead to these changes. That is, the instructions guided users through the four phases, but did not describe the acquisition processes. It is possible and probable that for the model to be used successfully, instruction needs to include process and representation information.

Mapping Learning onto Instruction An implicit assumption in the design of the four phase training system is that by mapping knowledge representations and learning processes directly onto instruction, we can design better instruction that will facilitate learning. It is possible that this assumption is not totally correct and therefore, the four phase model did not lead to the best performance. Although understanding the user may be important to instructional design, it may be the case that what we know about knowledge representation and learning needs to be reorganized into a framework that is organized around the goals of the instruction. One source of the need for this reorganization is that learning analyses work from the means to the end (i.e., they look at how people progress to a certain level of expertise), whereas, instruction works from the end (knowledge to be learned) to the means. If we consider instructional design as a problem solving task, we can propose that the problem specification should reflect the goals of the instruction and the domain knowledge that is used to solve the problems and accomplish these goals is our knowledge of the "how" and "what" of learning.

Example-based Learning

Although the example-based subjects were able to successfully perform the training tasks without an extreme number of errors, their performance on the concepts test was significantly poorer than the other training treatments. Thus, although these subjects could use the database system, they did not have a good conceptual representation of how the system worked. One possible explanation for this result is that these subjects were performing the tasks by analogically mapping them onto the examples presented in the training system. That is, these subjects were applying the commands in the examples without understanding how the commands interact.

Learning by analogy has been shown to be a prominent method used by novices in a domain (Anderson, Farrell, & Sauers, 1984). However, as suggested by the current results, once the problems from which students analogize are taken away, the students are no longer able to perform successfully. In addition, past research has shown that using analogies often leads to misconceptions in students' thinking (Douglas & Moran, 1983) because students have difficulty noting how the analogy and the problem are similar/different (Halasz & Moran, 1982).

One of the motivations behind using the example-based training system was to examine examples as a sole source of instruction since it is more often the case that examples are included as part of more detailed instruction. It appears that examples when presented by themselves provide only performance information and do not provide information at a more conceptual level. Therefore, one might propose that examples should be included in addition to other forms of information that do provide conceptual information.

Task-oriented vs. Command-oriented

Schema representations. One possible explanation for the observed results is that the task-oriented method of instruction allows the subjects to see the use of the commands as a coherent whole. That is, presenting the commands in the context of the plans that they are part of provides subjects with enough information to form at least a sketchy schema for the plan that can later be easily applied. On the other hand, presenting the commands as separate fragments of information as in the command-oriented organization forces subjects to develop the plans on their own. The error results for subjects in the command-oriented treatment suggest that this instruction leads to a strategy in which learners decide on a goal to accomplish and then try all the commands that are related to this goal until they get their desired result. As a result of this trial and error strategy, subjects obscured the coherence of the inter-command relationships. Thus, when required to perform a similar task, they were not able to remember how they had previously performed the task. The results for the transfer phase in which the command-oriented subjects were still committing significantly more errors than the other subjects lends support to this claim.

Procedural/declarative representation. Another way of explaining the training system results pertains to the procedural/declarative distinction and more specifically, its influence on the acquisition of knowledge as proposed in the ACT* theory (Anderson, 1983). Applying the procedural/declarative representation to the methods of organization used in the current study, one can propose that the task-oriented organization provides users with procedural information whereas, the command-oriented organization provides only declarative information. When we present subjects with a procedural organization of information, we are showing them the how to accomplish domain-specific tasks and thus, facilitating the learning process. When we present subjects with a declarative organization of information, we are requiring them to first use their general problem solving procedures to interpret the declarative knowledge and then use this interpretation process to develop domain-specific procedures. Thus, we are making the the learning process more difficult.

Thus, combining this distinction and the results of the current experiment suggests that using a procedural organization of knowledge can lead to better performance. There are, however, two caveats to this proposal. First, it is not necessarily the case that procedural information alone will lead to the best performance. In the current experiment, the example-based training also provided some procedural information in that it provided users with information about how the commands are used together in a sequence for a specific example. However, as was discussed previously, this organization did not provide users with a complete conceptual understanding of the computer system. One possible explanation for the difference between the task-oriented and example-based training systems is that the task-oriented organization provides more abstract procedural information that includes information about the general goals that drive the procedures and thus, drive performance.

The other caveat for interpreting the use of procedural information to facilitate learning pertains to the system used in the current experiment. The database system used is a simple computer system (i.e. requires learning for only a small set of commands that can accomplish a small number of tasks) and only requires learning simple procedures. It is not clear that procedural instruction would afford the same benefits in a more procedurally complex domain.

The goals of this research were to improve instruction and learning by using a knowledge-based analysis of learning to guide the design of the instruction and at the same time use instructional design as a method of validation for the Kay and Black four phase model. The results of the study provide only partial evidence in support of these goals. As previously described, the four phase model training system was designed by mapping the learning model directly onto instruction. The results from the training system comparisons suggest that it is important to consider instructional goals in addition to learning goals and thus, instructional design should be viewed as more than a mapping process. In addition, the results from the model-based training suggests that instructional design can and should be used as an important test of the assumptions made in proposed models of learning. Future research will address the relationship between learning models and instructional design as it pertains to these issues.

References

Anderson, J.R. (1983). The architecture of cognition. Cambridge, MA: Harvard Press.

Anderson, J.R., Farrell, R. & Sauers, R. (1984). Learning to program in LISP. Cognitive Science, 8, 87-129.

Card, S.K., Moran, T.P., & Newell, A. (1983). The psychology of human-computer interaction. Hillsdale, NJ: Lawrence Erlbaum Assoc.

Carroll, J.M. (1984). MINIMALIST training. Datamation, 30(18), 125-136.

Carroll, J.M. & Carrithers, C. (1984). Training wheels in a user interface. Communications of the ACM, 27, 800-806.

Carroll, J.M. & Kay, D.S. (1985). Prompting, feedback and error correction in the design of a Scenario Machine. Proceedings of CHI '85, San Francisco, CA, 149-153.

Douglas, S.A. & Moran, T.P. (1983). Learning text editor semantics by analogy. Proceeding of the CHI '83 Conference on Human Factors in Computing Systems 207-211.

Halasz, F. & Moran, T. (1982). Analogy considered harmful. Proceedings of Human Factors in Computer Systems Conference, National Bureau of Standards, Gaithersburg, MD.

Kay, D.S. & Black, J.B. (1985). The Evolution of knowledge representations with increasing expertise in using a system. Proceedings of the Seventh Annual Conference of the Cognitive Science Society. Irvine, CA, August.

McKendree, J., Schorno, S., & Carroll, J.M. (1985). Personal planner: The Scenario Machine as a research tool. Videotape demonstration presented at CHI '85.

Rissland, E.L., Valcarce, E.M., & Ashley, K.D. (1984). Explaining and arguing with examples. Proceeding of the AAAI, 288-294.