The Evolution of Knowledge Representations with Increasing Expertise in Using Systems

Dana S. Kay

Yale University

John B. Black

Teachers College, Columbia University

The kinds of knowledge that people have about a computer system and the form in which that knowledge is represented and stored in memory changes as people progress from being beginners to being experts at using the system. By combining a series of studies that we have conducted over the last few years, we have concluded that the knowledge representations that people have about a system progress through four phases as they become increasingly expert. In this paper, we describe these four phases of knowledge evolution during learning and justify each phase using the results of our studies.

The novice/expert paradigm has recently been given a great deal of attention. Studies have examined the differences between novices and experts in domains such as algebra (Lewis, 1981), physics (Larkin, 1981) and computer programming (e.g., Ehrlich and Soloway, 1984; McKeithen, Reitman, Rueter and Hirtle, 1981; Adelson, 1981). Although this paradigm provides insight into the conceptual and performance differences of novices and experts, it does not address the learning process that underlies the transition from novice to expert. To understand what makes an expert expert, it is necessary to examine the learning process. One method for accomplishing this goal is to investigate the organization of the domain-relevant information at various stages in learning. The assumption behind this method is that as learning progresses, the content and organization of the knowledge changes to accommodate the acquired knowledge. Because of the recent interest in computers and how to make them easier for people to learn, we chose to study learning in the computer domain by examining the development of knowledge representations appropriate for this domain.

Others have studied the differences in the knowledge representations of expert and novice programmers (e.g., Ehrlich and Soloway, 1984; McKeithen, Reitman, Rueter and Hirtle, 1981; Adelson, 1981) and the knowledge and performance of expert users of command languages (e.g., Card, Moran and Newell, 1983), but there have been no previous studies of how people's knowledge representations change as they learn to use command language systems. Our research has focused on how people learn to use text editors and, in particular, how their knowledge representations change as they become increasingly expert at using text editors. We are interested in understanding the order in which the components of the knowledge representations are acquired and how this acquisition process is reflected in the conceptualization of the text-editing knowledge. The model that we will present is an incremental model in which the user builds a complex knowledge representation by acquiring different types of knowledge at different phases in learning.

In our discussion here we will draw upon three studies to make our points: the Sebrechts, Black, Galambos, Wagner, Deck and Wikler (1984) study of people learning to use the IBM Displaywriter and the UCSD p-system text editor; the Kay and Black (1984) study of the differences in people at two different expertise levels with a local Yale text editor, and the Robertson and Black (1983) study of the timing changes between keystrokes as people learned to use a simple experimental text editor. Taken together these studies use a variety of research methods to provide converging evidence for our learning phases. Specifically, the Sebrechts, et al. study examined the changes in people's perceptions of command similarity that result from training on two commercial text editors, Kay and Black examined the differences in perceptions of similarity in different people who naturally acquired different levels of expertise with a text editor, while Robertson and Black examined inter-keystroke time evidence for how people conceptualize a simple artificial text editor as they become practiced in using it.

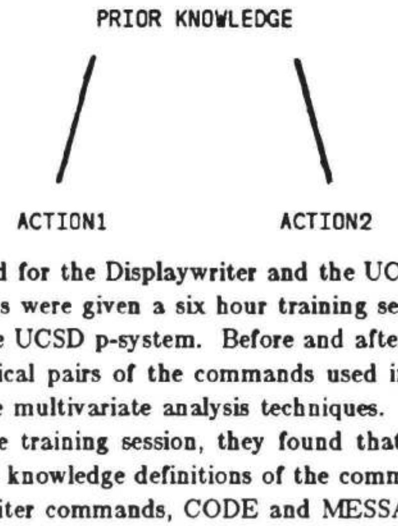
The Four Phases of Learning

The results of these studies have led us to conclude that the knowledge representations of users evolve through four phases as they become expert with a system. In particular, in the beginning user knowledge is determined by preconceptions based on the terminology used for the system, but with a little experience this changes to knowledge of what system commands are relevant to accomplishing various tasks with the system. However, with even more experience the form of the knowledge changes again to form complete plans (i.e., sequences of commands) for accomplishing goals with the system and with increasing expertise the plans become more complex and the user learns when each of these plans is most appropriately selected to accomplish the goal. In the following, we explain each of these phases in detail and describe the evidence supporting each.

Phase One: Preconceptions

Phase One represents the completely naive user who has had no experience using a text-editor. At this stage of learning, users have preconceptions about the terminology that will later refer to editing commands. As a result of prior experience with the terminology to be used in text-editing, users come to the text-editing domain with a knowledge representation that may or may not correspond to the knowledge representation that will develop as text-editing experience increases. Figure 1 presents an example of the type of knowledge structures that exist before any learning has taken place. In this structure, there are two actions (e.g. CENTER and BALANCE) that are related by prior knowledge (e.g. these are actions that make something even).

FIGURE 1: Preconception Knowledge Structures



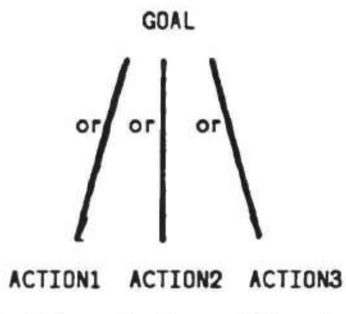
Evidence of this phase was found for the Displaywriter and the UCSD systems in the Sebrechts et al. study. In this study, naive subjects were given a six hour training session in which they learned to use either the IBM Displaywriter or the UCSD p-system. Before and after training, the subjects were asked to rate the similarity of non-identical pairs of the commands used in each system. To analyze these ratings, Sebrechts et al., used three multivariate analysis techniques. From the results of a hierarchical clustering of the ratings before the training session, they found that initially the commands for both systems clustered based upon prior knowledge definitions of the commands. For example, in the results of the clustering for the Displaywriter commands, CODE and MESSAGE and CANCEL and DELETE were clustered together. CODE and MESSAGE are related in that one can use a code to convey a message. CANCEL and DELETE are related in that they both suggest the elimination of something. In both cases, the similarity of the command names reflects prior knowledge associations between the actions. Similar types of clusters were found in the UCSD system such as FIND and GET and HEADER and MARGIN.

Although there may be some correspondence between the prior knowledge representations of the commands and the text-editing representations, in most situations, the two representations are quite different. Therefore, users in Phase One of the learning process are confronted with the task of overcoming a bias toward interpreting the commands in terms of their prior knowledge associations. To accomplish this task and achieve any level of expertise, the previously existing knowledge representations must be reorganized to accommodate the acquisition of text-editing knowledge.

Phase Two: Initial Learning

Initial text-editing knowledge can be acquired from a manual, a class, self-teaching or any combination of these. No matter what the learning method, the main goal that the user has is to overcome the prior knowledge bias that exists for the text-editing commands. We believe that the accomplishment of this goal entails (1) learning the goals relevant to text-editing and (2) learning the commands that can be used to accomplish these goals. The first part of the learning process takes place as soon as the user begins to edit and is exposed to various editing tasks. Knowledge of general goals allows for the generation of high-level goal structures that can be used to organize the editing commands. For example, if a user types the word "thee" instead of "the", then the goal or task that is instantiated is to erase the extra "e". However, before this goal can be accomplished the user must learn the actual editing command(s) that are used. That is, the user must learn the functions of the commands and selectively choose the most appropriate command(s). To continue the previous example, a user might learn that the command BACKSPACE serves to get rid of unwanted text and decide to use BACKSPACE to remove the unwanted letter. At this time, the user learns that one of the possible goals to be found in text-editing is to remove unwanted text and one command that accomplishes this goal is BACKSPACE. As a result of this learning, the user develops knowledge structures that link specific goals and commands. Figure 2a presents the knowledge structures that exist at this phase of learning. Here the goal is linked to the actions by OR links because one can use ACTION1 or ACTION2 or ACTION3 to accomplish the goal. For our previous example, the GOAL might be GET-RID-OF-TEXT and ACTION1 might refer to BACKSPACE. As more commands that get rid of text, such as DELETE and SPACE, are learned, they are added to the representation with OR links because the user knows that if you want to GET-RID-OF-TEXT, you can use BACKSPACE or DELETE or SPACE.

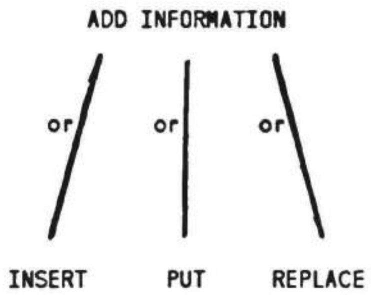
FIGURE 2a: Initial Learning Knowledge Structure



Kay and Black (1984) used a methodology similar to Sebrechts et al., but with a cross-sectional design. In this study, they examined the differences in the knowledge structures of novices and experts who naturally acquired text editing knowledge. They asked both types of users to rate the similarity of pairs of commands used in a local Yale editor and analyzed these ratings using multivariate techniques. In the results of the hierarchical clustering for the novice users, commands were clustered based upon similarity of function in the editor. For example, INSERT, PUT and REPLACE clustered together because all three commands are used to accomplish the goal of adding information to a text. This structure is represented in Figure 2b. Each of these actions could be use to accomplish the goal of adding information. Therefore, all the commands are linked to the goal by OR links. That is, if one wants to add information, INSERT or PUT or REPLACE can be used. These results will later be contrasted with the expert results in our discussion of Phase Three of the learning process.

It is important to note that although these users have acquired some text-editing knowledge, the organization of this knowledge is based upon only the "result" of the commands and not the procedure that leads to this result. Because of this narrow focus, the users do not possess the knowledge necessary to develop more complex structures such as plans. In particular, while the users know that PUT can be used to ADD INFORMATION, they do not readily know that PUT is merely the main action in a multi-action plan to accomplish this goal.

FIGURE 2b: Knowledge Structure for Adding Text



In addition to providing evidence for Phase One, the results of the hierarchical clustering in Sebrechts et al. also illustrate the transition from Phase One to Phase Two. After six hours training, users no longer clustered commands by prior knowledge associations. The clusters changed to be based more on the function of the system. For example, DELETE which was initially clustered with CANCEL, was clustered with BACKSPACE after training. In the Displaywriter system, either DELETE or BACKSPACE can be used to erase a piece of text.

Because novice users have not refined their definitions of commands, several commands are often linked to a single goal by OR links. At this level of understanding, the user will employ any one of the actions linked to the goal to accomplish the goal. For example, in the knowledge structure presented in Figure 2b, PUT, INSERT and REPLACE are conceptualized as similar because they all accomplish the goal of adding information to the text. With experience, the user will develop more complex representations of these commands that include knowledge of the processes (or plans) that lead to the results of the commands and learn that although these commands all add information to the text, the added information comes from different sources. Evidence for this change in knowledge organization will be presented in Phase Three. For example, PUT uses information from the buffer and needs PICK or DELETE to put the information into the buffer. However, INSERT needs only the information that is typed by the user.

In addition to using hierarchical clustering to examine the knowledge representations of users, Kay and Black and Sebrechts et al. also used multidimensional scaling to study the overall organization of the commands. The results of this analysis were the same in both studies. The commands were organized along (a) a dimension that differentiated system and editor commands (b) a dimension that distinguished formatting from non-formatting commands and (c) a dimension that differentiated commands to begin a sequence from commands to end a sequence. These dimensions suggest the development of a goal space for the commands. This goal space helps the user in accessing the correct goals during an editing session. For example, if one is formatting a document, it is more appropriate to have the goal of centering the text (and accessing the CENTER command) than it is to have the goal of changing a misspelled word. The third dimension (begin/end of sequence) is particularly interesting because it suggests that although users do not organize the commands by specific sequences (plans), they do understand that commands are used in sequences.

Card, Moran and Newell (1983) proposed the GOMS model to account for the text-editing behavior of experts performing routine tasks. In this model, the expert knowledge representation consisted of four components Goals, Operators, Methods and Selection rules. Using our account of initial learning, we propose that it is the Goals and Operators components of the GOMS model that are acquired first. That is, novices are able to understand the general goals involved in text-editing and the individual commands that are related to these goals. It is reasonable that the goal/action link would be the first link to be formed. During the initial learning of a system, the user is introduced to numerous command names and definitions with little reference made as to when each command should be used. If, instead of forming links between the commands and the goals that they accomplish, the first links to be formed were between commands, then the user would never know when to use each command and therefore, would have to resort to a trial-and-error method of achieving a goal. Thus, the linking of the command to a goal provides the user with some aid in using the correct command in the correct situation.

Novices seem to conceptualize the commands merely by what goals they are relevant for accomplishing. Because of this level of specificity, they have not yet acquired the procedures or plans that are associated with text-editing and thus, each text-editing task becomes a problem-solving task in which they must actively search through their representations of the commands and find the set of commands necessary to accomplish the task.

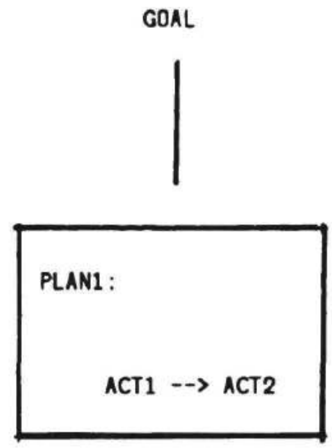
Phase Three: Plan Development

Once the users have acquired the basic editing commands and goals, they learn that there are combinations of commands that are often used together to accomplish a goal. In Phase Three, users develop the ability to form plans by combining the actions that were organized separately in Phase Two. These plans correspond to the Methods of the GOMS model. There are various ways that the transition from Phase Two to Phase Three takes place. For example, with the system that Kay and Black studied, users realize the inefficiency in repeating a command numerous times. To overcome this inefficiency they learn to use the ARGUMENT command that automatically repeats another command for the number of times specified as the argument. Once they have this knowledge, users begin to notice that there are other commands that can be used together in a sequence. This realization leads to a reorganization of the knowledge representation to accommodate the command sequences or plans that are used to accomplish goals.

This reorganization process entails the modification of the goal/action links that were formed in Phase Two. In this third phase, two types of links are formed. At one level, links are formed between the commands or actions that are used in a plan. At a higher level, links are formed between the goals and the plans that are used to achieve the goal. It is also possible to represent this knowledge as productions and explain the transition from Phase Two to Phase Three using the composition process proposed by Anderson (1983). However, since we are primarily interested in specifying the knowledge that is acquired at each phase of learning, we chose to use a network representation in which we trace the development of the links in the network because we think the networks are more perspicuous than productions.

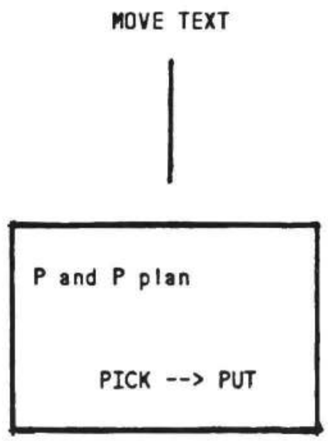
Figure 3a presents the type of knowledge structures that exist at Phase Three of the learning process. In these structures, just as in the Phase Two structures, the goals will guide the use of the commands. However, this guidance is provided by the instantiation of a plan that consists of actions, rather than invoking each action individually.

FIGURE 3a: Beginning Expertise Knowledge Structure



Again, both Kay and Black and Sebrechts et al. provide evidence for the development of Phase Three knowledge structures using the results of a hierarchical clustering analysis. As previously mentioned, the novice users organized commands as individual actions related to goals. On the other hand, expert users appeared to use a more sequence-oriented organization. That is, the perceived similarity between the command pairs was based upon the use of the commands in a plan to accomplish a given goal. For example, in the Kay and Black study, expert users clustered PICK and PUT together because these two commands are used together when a user want to accomplish the goal of moving a piece of text. This example, when contrasted with the Phase Two knowledge representation example in which novice users clustered PUT with INSERT and REPLACE because each of individual commands are used to accomplish the goal of adding information, illustrates the change in the representation of the commands from goal/action links to goal/plan links. Figure 3b presents a graphic representation for the knowledge structure that underlies this result.

FIGURE 3b: Move Text Knowledge Structure



In Phase Two we found that in addition to having specific goal/action knowledge structures, users also organized the commands in a goal space. Given the knowledge structure changes that occur from Phase Two to Phase Three, it is interesting to look at whether or not these changes influence the goal space of the users. The multidimensional scaling results for the experienced users in Kay and Black suggest that there is only one primary change in the goal space. Recall that in the initial learning phase, the dimensions of the goal space are editor/system, formatting/non-formatting and begin/end sequence. Of these three dimensions the former two were also present in the multidimensional scaling for the experienced users. However, the begin/end sequence dimension is no longer used. This result suggests that experienced users continue to organize the commands in a goal space, but now they realize that interactions with the system don't occur in as rigid an order as they originally thought so the begin/end dimension disappears. They still have sequence knowledge, but it is at the more local plan level, instead of the global begin/end sequence level.

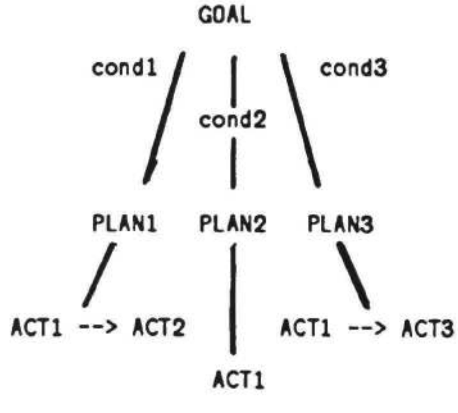
Because users begin to develop plan representations for their editing knowledge, we believe that it is during this phase in learning that users begin to mold the system to suit their own editing style by implementing these plans as macros. This molding makes their performance more efficient because using plan-macros, they can accomplish a goal with one command rather than several. We are currently testing this hypothesis by examining the number and content of macros created by users at various levels of expertise.

Phase Four: Increasing Expertise

Although the formation of simple plans results in some expertise in text editing it is not until compound plans are formed that one can accomplish more advanced tasks. Phase Four of our model accounts for this ability and represents the completion of the acquisition that results in knowledge representations similar to those proposed in the GOMS model of expert performance. In this phase, users (a) combine simple plans into more compound plans to accomplish major goals and (b) develop rules for selecting the best plan to achieve a given goal in a given situation. Once again, we observe a reorganization of the knowledge that results in the development of new links between the components of the representation. In this phase, we see a change from a one-to-one correspondence between goal and plan to a one-to-several correspondence: that is, in Phase Three each plan or sequence of actions is directly linked to a specific goal (e.g. move text) while in Phase Four, there are multiple plans that can be linked to each goal. However, because there are multiple plans, the links that connect these plans must have selection rules that tell the user under what conditions to access the plan to accomplish the goal.

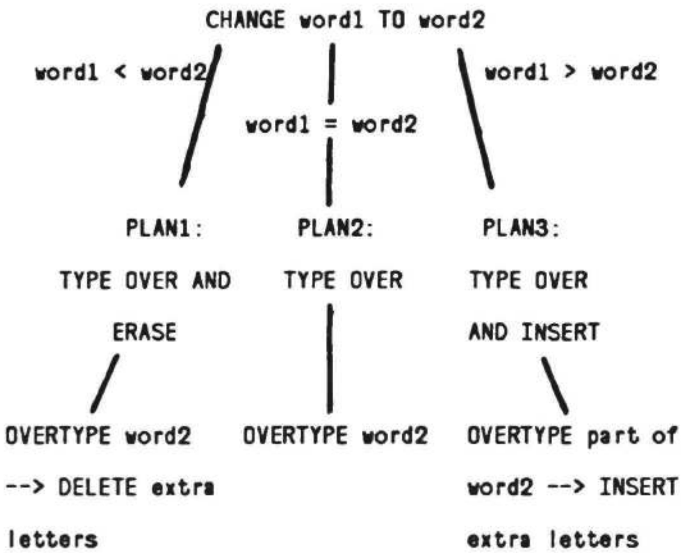
Figure 4a shows the type of knowledge structures possessed by users in Phase Four. In this type of structure, there are several plans (sequences of actions) that may be instantiated to achieve a given goal. To be sure that the correct plan is chosen from the set of applicable plans, the links that connect these plans to the goal are conditions or selection rules that must be met before a given plan is chosen. These conditions can be based upon any distinguishing feature of the plans. Thus, at the highest level of expertise, goals are linked to plans using the conditions under which these plans are invoked, whereas, goals were linked to simple plans in Phase Three and actions in Phase Two.

FIGURE 4a: Expert Knowledge Structures



Robertson and Black present evidence for the evolution of compound plans. With increased experience, the time spent pausing between simple plans decreased suggesting that users had combined the simple plans that they had learned to form compound plans. In addition, the decision time to initiate a compound plan during which subjects choose what they think is the most appropriate plan decreased as the editing session progressed suggesting the acquisition of selection rules that facilitate accessing the most appropriate plan. Figure 4b presents a concrete example of a Phase Four knowledge structure. In the editor Robertson and Black used, to change one word to another word, there are three possible plans (sequences of actions) that may be invoked. The links connecting these three plans to the goal are conditions that must be met before the plan is selected. In the structure depicted, the conditions are based upon the relationship in length between the old word and the new word.

FIGURE 4b: Knowledge Structures for Changing a Word



In addition to providing evidence for the compound plans of Phase Four, Robertson and Black also found evidence for the development of selection rules. In the beginning of the training trials, when users had to change one word to another word, the majority of the users (approximately two thirds) would DELETE the old word then INSERT the new word. However, as experience increased, these same users stopped using the DELETE-INSERT plan and began to OVERTYPE the old word with the new word. In this example, users appeared to develop a selection rule that can be stated as "If you want to change one word to another, use the OVERTYPE plan." This type of selection rule organizes the plans according to the priority that they have in efficiently accomplishing the goal.

The Learning Process

Our four learning phases describe the evolution in user behavior from problem-solving to plan instantiation. That is, initially users must interpret each action in the editing task and monitor the success or failure of the action in terms of the final goal state. However, once the user is familiar with text-editing tasks, the commands are combined into plans that are applied when appropriate. It is the latter process that accounts for the plan-boundry pauses found in the Robertson and Black study, because deciding which plan to use takes longer than the transition from one plan action to another plan action when applying the plan.

Throughout this discussion of our four phase model, we have made references to the GOMS model and the order in which the components of this model are acquired. Although we are not able to distinguish whether the Goals are acquired before, after, or simultaneously with the Operators, we are able to conclude that the Goals and Operators are acquired before the plans that are acquired before the Selection rules. In addition, we noted that as the user becomes more experienced, the components build upon one another so that the chunks of knowledge represented using these components increase in size. Thus, we were able to trace the evolution of text-editing knowledge structures and extract an acquisition process that in the end results in knowledge representations similar to those found in the GOMS model.

Why do these learning phases occur: that is, what is the learning process that causes the user of a system to progress through these phases of learning? The initial, preconception phase is clearly necessary, because new users have only their prior knowledge to help them understand and use a system in the beginning. The final, increasing-expertise phase is also clearly necessary, because compound plans for accomplishing major goals and the selection rules for choosing plans at the most appropriate times are both necessary for skilled performance (Card, Moran and Newell, 1983). But, why are the two intermediate phases necessary?

Users cannot progress directly from phase one to phase four representations, because transforming phase one representations to phase four representations requires bringing more information together at one time than human working memory is capable of holding simultaneously. Thus, the two intermediate knowledge representations allow the user to progress from phase one to phase four in bite-size chunks that correspond to the limits of human working memory. For example, to progress from phase one to phase two, the users need only learn the links between the commands and the goals that can be accomplished, they do not need to simultaneously learn the sequencing links between the commands that are needed to combine the commands into plans. Then once the phase two command-goal links are mastered, the users can progress to phase three by learning the sequencing links that combine the relevant commands into simple plans for accomplishing elementary goals. Finally, once the simple plans are well learned, the user can progress to phase four by combining these simple-plan chunks into more compound plans and learning the conditions that determine when the various plans are most appropriately selected for accomplishing the goals.

However, while an intermediate step between phases one and four is necessary, phases two and three may not both be necessary. The simple plans of phase three are clearly a necessary precondition for the compound plans and selection rules of phase four, but phase two might be an artifact of the way that systems are currently taught to new users. In particular, current instruction manuals emphasize descriptions of individual commands at the expense of describing how these commands are combined into plans. We are currently investigating whether plan-based instruction manuals will allow new users to skip phase two and progress directly to phase three, thus significantly speeding up the process of learning new systems.

We are also pursuing the model's generality and testing its predictions. In particular, we are testing the the generalization of our model to other domains. The most immediate extension from the text-editing domain is to computer programming. We have begun some preliminary work applying our model to this domain and plan to carry out a full scale longitudinal study in the near future. However, in the more distant future, we plan to try extending the model to non-computer domains (e.g. learning physics, algebra and the operation of devices).

To test the model's predictions, we are using it to guide training in text-editing. Since our model was extrapolated from our observations of the natural evolution of knowledge representations for text-editing information, we believe that using this model to train users might facilitate the learning process. We are currently designing a study to test this hypothesis by comparing the performance of users whose learning was guided by the phases in our model to the performance of users whose learning was guided by commercial training materials in which the emphasis is on the kinds of commands that can be used in the system.

Acknowledgement: This research is supported by grants from IBM, but represents the views of the authors and not necessarily the views of IBM.

REFERENCES

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

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

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

Ehrlich, K. and Soloway, E. (1984). An empirical investigation of the tacit plan knowledge in programming. in Thomas, J. and Schneider, M. (Eds.) *Human Factors in Computer Systems*, Norwood, New Jersey: Ablex Publishing Corp.

Kay, D.S. and Black, J.B. (1984). The changes in knowledge representations of computer systems with experience. *Proceedings of the Human Factors Society 28th Annual Meeting 1984*. San Antonio, TX.

Larkin, J.H. (1981). Enriching formal knowledge: A model for learning to solve textbook physics problems. In J.R. Anderson (Ed.) *Cognitive Skills and their Acquisition*, Hillsdale, NJ: Erlbaum.

Lewis, C. (1981). Skill in algebra. In J.R. Anderson (Ed.) *Cognitive Skills and their Acquisition*, Hillsdale, NJ: Erlbaum.

McKeithen, K.B., Reitman, J.S., Rueter, H.H., Hirtle, S.C. (1981). Knowledge organization and skill differences in computer programmers. *Cognitive Psychology*, **13**, 307-325.

Robertson, S.P. and Black, J.B. (1983). Planning in text editing behavior. *Proceeding of the Chi'83 Conference on Human Factors in Computing Systems*. Boston, MA.

Sebrechts, M.M., Black J.B., Galambos. J.A., Wagner, R.K., Deck, J.A., and Wikler, E.A. (1983). The effects of diagrams on learning to use a system. (Report No. 2). Learning and Using Systems, Yale University, New Haven, CT.