

## Knowledge Acquisition and Interface Design for Learning On Demand Systems

Wayne A. Nelson

Department of Educational Leadership  
Southern Illinois University at Edwardsville

The rapid changes in our world precipitated by technology have created new problems and new challenges for education and training. A knowledge "explosion" is occurring as our society moves toward a service-oriented economy that relies on information as the major resource. Complex computer systems are beginning to dominate the workplace, causing alarming growth and change in many fields. The rapidly changing nature of the workplace, especially in fields related to information technology, requires that our knowledge be updated constantly. This characteristic of modern society poses seemingly unsolvable instructional problems involving coverage and obsolescence. The sheer amount of information to be learned is rapidly increasing, while at the same time some information becomes obsolete in light of new information. Education, therefore, must become a lifelong process that features learning of new material and skills as needed in relation to the job to be done.

Because of the problems cited above, the current model of learning in advance may no longer be feasible in our high-technology world. In many cases, learning in advance is impossible because there are simply too many things to learn. In addition, learning in advance can be time consuming, and often results in decontextualized knowledge that does not readily transfer to the work environment. The large and growing discrepancy between the amount of potentially relevant knowledge available and the amount a person can know and remember makes learning on demand an important alternative to current instructional practices. Learning on demand takes place whenever an individual must learn something new in order to perform a task or make a decision. Learning on demand is a promising approach for addressing the problems of coverage and obsolescence because learning is contextualized and integrated into the task environment rather than being relegated to a separate phase that precedes work. Learning on demand allows learners to see for themselves the usefulness of new knowledge for actual problem situations, thereby increasing the motivation for learning new information. Finally, learning on demand makes new information relevant to the task at hand, leading to more informed decision making, better quality products, and improved performance.

Successful models of learning on demand have existed where a learner could afford the luxury of a personal coach and critic who lends support to the learner's individual problem-solving activities. For a variety of reasons, this relationship is difficult to achieve in modern society between humans, but advances in computer technology now make it possible for the computer to assume the role of coach and critic in many situations where the machine is already used as a tool to complete work. With a computer-based learning on demand system, learning does not take place in a separate phase and in a separate place, but instead is integrated with working and contextualized by real problem situations. In fact, learning on demand carries the potential to truly set computer-based learning environments apart from other instructional media, and therefore may be a unique research direction where technology can make a significant difference in achieving new educational objectives and overcoming some of the basic limitations of our current instructional approaches.

### CONCEPTUAL FRAMEWORK

Learning-on-demand systems are based on recent findings in cognitive science that suggest the learning process involves knowledge construction, not knowledge absorption (Papert, 1980). Successful learners elaborate and develop self-explanations that extend the information in texts or other instructional materials. Learners use current knowledge to construct new knowledge and to restructure existing knowledge (Kintsch, 1988). Learning is highly related to the situation in which it takes place (Greeno, 1989; Lave, 1988; Suchman, 1987). No amount of knowledge of general principles accounts for or guarantees the success of action in real-world problem situations. In other words, general information needs to be elaborated and experienced in various contexts in order to devise specific courses of action that can be applied in specific problem situations. There is also overwhelming evidence that self-directed (Brown & Palinscar, 1989) and intentional learning (Bereiter & Scardmalia, 1989) are some of the most effective components of successful learning experiences.

Learning on demand systems also utilize the reflection-in-action model that describes how many practitioners proceed in their work (Schoen, 1983). Action is governed by a nonreflective thought process that proceeds until a breakdown occurs, when the practitioner realizes that nonreflective action has resulted in unanticipated consequences (positive or negative). In order for the practitioner to notice the breakdown, the situation has to "talk back" to the practitioner, so that reflection can be used to repair the breakdown and nonreflective, situated action can resume. Reflection-in-action takes place within the time period during which the decision to act has been made, but the final decision about how to act has not. Mechanisms in support of learning on demand must take advantage of these breakdown situations in order to provide information when it is most needed.

## TYPES OF LEARNING ON DEMAND SYSTEMS

Learning on demand has been described by Fischer and his colleagues (Fischer, 1991; Fischer, Lemke, Mastaglio, & Morch, 1990; Fischer, Lemke, & Schwab, 1985), who have developed two types of systems: active help systems and critiquing systems. Active help systems, also known as context-sensitive help systems, are typically embedded in a software application, interrupting the user to "volunteer" information based on an analysis of the task the user is attempting to complete. Critiquing systems can be integrated within a variety of work environments to present a reasoned opinion about a user's product or action using knowledge of domain principles to detect and critique suboptimal solutions or actions. Other researchers working in the area of electronic performance support systems (EPSS) are studying learning on demand as a form of decision support (e.g., Gery, 1991). Such systems may be designed as expert systems that identify problematic decisions and suggest alternatives, or as on-line documentation for learning about aspects of the job that are infrequently performed. Finally, "intelligent applications" are beginning to emerge that facilitate production tasks by automating certain aspects of the tasks, or by suggesting ways that the user may use the software to accomplish various goals.

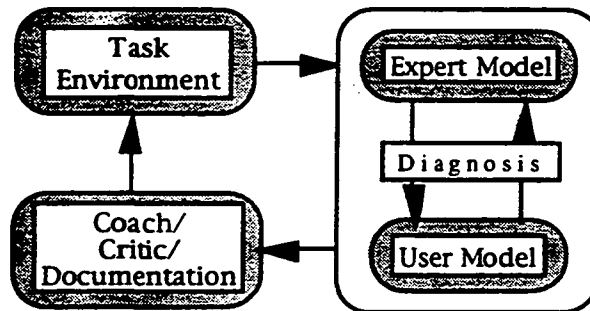
Regardless of the type of learning on demand system under development, issues of system architecture and knowledge representation still remain at the forefront. Essentially, learning on demand environments are similar to expert systems and intelligent tutoring systems. In fact, a general architecture for a learning on demand system, as depicted in Figure 1, is similar to those used for intelligent tutoring systems. A task environment (interface) where the user completes work in the domain is designed to communicate with an expert model that represents the various types of activities that can be accomplished with the system. The actions taken by the user are represented in a user model, and diagnosis of the user's actions can occur through comparison with the expert model. When errors are made (or help is requested), a coach or critic is invoked to communicate with the user within the task environment.

The types of knowledge required for learning on demand systems, and the subsequent knowledge representations within the computer, will also be similar to intelligent tutoring systems and expert systems. Considerable debate among cognitive scientists about the nature of declarative and procedural knowledge has resulted in the recognition that the appropriate knowledge representation scheme depends on the problem being addressed (Gardner, 1987). Certainly all representations can be reduced to bits in computer memory, but just as certain high-level computer programming languages are more effective for particular types of problems, the features of various knowledge representation schemes can make one type of representation more appropriate than others.

In general, high-level representations can be categorized according to representational characteristics and system functions. Semantic networks, augmented transition networks, bites, frames, scripts, and grammars have been used to construct systems which hierarchically organize information for tutoring, simulation, and parsing of natural language. These schemes use highly structured databases that allow reasoning based on links between concepts, inheritance of values from superordinate concepts, and procedures for dealing with specific objects. Modeling physical processes, problem-solving processes, and decision behavior has been achieved with behavior trees, dependency graphs, planning nets, probability matrices, and procedural nets. These networks have nodes representing various modules of a process, and links representing relations and interactions between real-world structures and processes. Rule-based production systems utilizing GAO graphs, overlays, and goal-directed productions have been employed to implement skill acquisition, tutoring and diagnosis. These techniques make comparisons between models of expert behavior and the actual student behavior in order to prescribe appropriate

intelligent tutoring. These higher-level representations have been developed for many different applications (see Wenger, 1987).

Figure 1. A general system architecture for learning on demand.



## KNOWLEDGE ACQUISITION FOR LEARNING ON DEMAND SYSTEMS

To develop expert systems or intelligent tutoring systems, the knowledge structures and decision-making processes experts utilize must be elicited. Decisions must be made about which knowledge representation scheme is most appropriate, and how the expert's reasoning might best be implemented by the computer. Knowledge engineers typically accomplish this by working in several stages where they analyze the problem domain, determine the key concepts for representation in the system, design an informal structure for the knowledge to guide subsequent knowledge acquisition activities, structure the knowledge in a formal computer-based representation, and then validate the knowledge and system performance (Nelson, 1989). Since learning on demand systems closely resemble intelligent tutoring systems, the knowledge acquisition strategies already developed for building intelligent tutors may also be appropriate for learning on demand systems.

One of the major problems in knowledge acquisition results from trying to fit the knowledge of an expert into a representational scheme which was chosen too early in the process. It is important that the content and organization of the system be matched to that of the expert, not vis versa. Consulting written references can be an effective way to form an initial knowledge base, and may be the only technique necessary for learning on demand systems that function as job aids for consulting little-used information before making decisions. If the goal is to construct a learning on demand system that monitors user actions, it will be necessary to consult with domain experts to elicit procedural knowledge and problem-solving strategies. This process should begin by having the expert tutor the knowledge engineer on the domain and terminology. During consultations with the expert, it is also important to pay attention to the expert's non-verbal cues of absorption in the task. Finally, test cases can be presented to the expert in order to elicit and refine the representation of the expert's knowledge (Evanson, 1988; Prerau, 1987).

Common methods for eliciting expert knowledge typically fall into one of two categories: interviews and protocol analyses (Cooke & McDonald, 1986). Both techniques are relatively ill-defined, time-consuming, and susceptible to communication barriers that can inhibit effectiveness, since the methods require experts to engage in introspection (Evanson, 1988). Interviews tend to be unstructured and center on spontaneous questions asked by the knowledge engineer as the expert performs the task (Hoffman, 1987). Success depends on the expert's ability to verbalize while performing, and on the knowledge engineer's ability to ask pertinent questions. Analyzing the data collected can be very complex, and may require using audio or video tapes (Ericsson & Simon, 1984). The expert's non-verbal behaviors may indicate the degree of mental effort required to solve certain problems and provide additional cues about what the expert is actually thinking.

Task analysis methods can be employed to provide information about what experts do during problem solving. As the expert performs a task, the knowledge engineer can observe the problem-solving goals and information used by the expert. Many of the tasks performed by experts fall into the general categories of interpretation, diagnosis, monitoring, prediction, planning and design (Stefik et al., 1983). These activities generally require

experts to perceive patterns, analyze similarities and differences in data, form concepts based on the relationships between data categories, make inferences, and test hypotheses. Task analysis methods can provide information about the basic knowledge and skills necessary for problem solving, but since these methods do not indicate why actions are undertaken, it is often necessary to employ cognitive task analysis to reveal more subtle aspects of the expert's reasoning (Means & Gott, 1988). Scenarios or plans derived from past experience, which experts may recall to make judgments when information is incomplete, can also be identified with such methods.

Several methods have been developed to generate specific representations of the conceptual knowledge structures of experts. Statistical procedures such as multi-dimensional scaling, cluster analysis and link-weighted networks can be used to analyze the proximity of pairwise sets of concepts in order to generate geometric representations of knowledge structures (Schvaneveldt, 1990). A less complex, easier to implement method involves the analysis of pattern notes (Buzan, 1974; Fields, 1980). The expert constructs a "map" of a subject area by free association of ideas related to the main topic. Secondary ideas are indicated by labeled lines extending from the main subject, and subordinate ideas are indicated by other extensions from these. Research has shown that knowledge structures elicited by the pattern note technique produces maps of cognitive structures very similar to those produced by statistical procedures mentioned above (Jonassen, 1987).

Another approach based on Kelly's (1955) personal construct theory utilizes repertory grid techniques to elicit and analyze personal beliefs from individuals (Boose, 1986; Hart, 1986). Constructs (attributes) and elements (instances) about a particular problem domain are accumulated through interviews (face-to-face or computer-based). Experts rate the relationships among the elements for each construct of the problem, and the ratings are assembled in a grid. The grid can be statistically analyzed to determine clusters of elements, and can be reorganized (focused) to better represent the expert's knowledge.

## **INTERFACE DESIGN FOR LEARNING ON DEMAND SYSTEMS**

The design of an interface for a learning on demand system is highly dependent on the type of task environment required and the features being developed for the application. For example, a learning on demand system for an "intelligent" word processing application will differ significantly from a control system for a nuclear reactor. The former might be designed to simply monitor the activities of the user in order to infer what the user might be trying to do when help is requested. The latter might be designed to demonstrate various operations to the user before allowing the user full control of the system, and would monitor the user at all times, regardless of expertise, to prevent fatal errors. These examples illustrate the various levels of abstraction in learning on demand environments (from real representations of the work environment to abstract representations) that Burton (1988) has described elsewhere.

Related to the level of abstraction of the interface is the notion of fidelity. It has been suggested that initial learning requires an interface that is high in fidelity, that is, the system feels, looks, acts, and seems like the actual task environment (Burton, 1988). Since learning on demand systems are in fact the actual work environments, fidelity may be less an issue than for intelligent tutoring systems that rely on simulation of the work environment. However, it is important to maintain task fidelity in many applications, both for the comfort of the user (would you like to draw with a graphics program that required you to type coordinates for every line?), and the possibilities for user modeling afforded by a high fidelity, direct manipulation interface.

There are a number of strategies borrowed from intelligent tutoring systems that can be employed in learning on demand environments to improve interaction and learning. Burton (1988) has described several forms of assistance to learners that may be adapted for learning on demand systems. Besides direct help available on request (and tailored to the current task the learner is completing), strategies might include assistance (where the system does part of the task for the user), empowering tools that record a user's actions and provide browsing of an action sequence for user reflection, modeling of the task (where the system performs the task while the user observes), coaching and critiquing interactions that are initiated by the system when a user makes an error, and direct online tutoring controlled by the system.

Finally, practical considerations for interface design also need to be incorporated into learning on demand systems. A balance between user control and system intervention must be maintained, especially in learning on

demand systems that employ coaching or critiquing. Considerations of the user's work tasks are also important, so that aspects of the learning on demand system do not interfere with the work required by the application. For example, in a decision support system, the information entered to allow the system to help in making a decision should also be available for the work requirements. It would be ridiculous to gather information on a prospective loan applicant, submit that information to an expert system for verification of the decision, then have to fill out additional forms with the same information for purposes of record keeping. Many applications have been designed without these simple, practical considerations, and not surprisingly, such applications are not well received by users.

## CONCLUSIONS

While learning on demand offers many possibilities, there are still a number of limitations. Certain essential skills may need to be acquired before work can begin, and since learning on demand is task driven, it may expose learners to isolated pieces of knowledge that require further integration and elaboration. Learners may also encounter difficulties in decontextualizing knowledge so that it can be used in new settings, and learning on demand may not support substantial restructuring of knowledge, since the additional information learned occurs only in relationship to what the learner already knows.

Nevertheless, learning on demand represents a new and relatively untested solution to the problems of learning and instruction cited earlier, and raises many questions and challenges for psychology, education and computer science. It has not been established whether learning on demand is more efficient than conventional forms of learning. Little is known about what classes of users (e. g., novices, intermediates, experts) benefit most from using an integrated learning on demand work environment. What intervention strategies should the system use for providing information without disrupting the work process? When will users suspend the work process and access relevant information? Additional research that addresses these and other questions is needed before learning on demand systems can assume the position of primary instructional role that has been predicted by advocates of these systems.

## REFERENCES

- Bereiter, C., & Scardmalia, M. (1989). Intentional learning as a goal of instruction. In L. B. Resnick (Ed.), *Knowing, learning and instruction: Essays in honor of Robert Glaser* (pp. 361-392). Hillsdale, NJ: Erlbaum.
- Boose, J. H. (1986). *Expertise transfer for expert system design*. New York: Elsevier.
- Brown, A. L., & Palinscar, A. S. (1989). Guided, cooperative learning and individual knowledge acquisition. In L. B. Resnick (Ed.), *Knowing, learning and instruction: Essays in honor of Robert Glaser* (pp. 393-451). Hillsdale, NJ: Erlbaum.
- Burton, R. R. (1988). The environment module of intelligent tutoring systems. In M. C. Polson & J. R. Richardson (Eds.), *Foundations of intelligent tutoring systems* (pp. 109-142). Hillsdale, NJ: Lawrence Erlbaum.
- Buzan, T. (1974). *Use both sides of your brain*. New York: Dutton.
- Cooke, N. M., & McDonald, J. E. (1986). A formal methodology for acquiring and representing expert knowledge. *Proceedings of the IEEE*, 74(10), 1422-1430.
- Ericsson, K. A., & Simon, H. A. (1984). *Protocol analysis*. Cambridge, MA: M. I. T. Press.
- Evanson, S. E. (1988). How to talk to an expert. *AI Expert*, 3(2), 36-41.
- Fields, A. (1980). Pattern notes. *NSPI Journal*, 19(8), 12-17, 43.

- Fischer, G. (1991). Supporting learning on demand with design environments. In L. Birnbaum (Ed.), *Proceedings of the 1991 International Conference on the Learning Sciences* (pp. 165-172). Charlottesville, VA: Association for the Advancement of Computing in Education.
- Fischer, G., Lemke, A. C., Mastaglio, T., & Morch, A. (1990). Using critics to empower users. *Human Factors in Computing Systems: CHI '90 Conference Proceedings* (pp. 337-347). New York: Association for Computing Machinery.
- Fischer, G., Lemke, A. C., & Schwab, T. (1985). Knowledge-based help systems. *Human Factors in Computing Systems: CHI '85 Conference Proceedings* (pp. 161-167). New York: Association for Computing Machinery.
- Gardner, H. (1985). *The Mind's New Science*. NY: Basic Books.
- Gery, G. (1991). *Electronic performance support systems*. Boston: Wiengarten.
- Greeno, J. G. (1989). Situations, mental models, and generative knowledge. In D. Klahr & K. Kotovsky (Eds.), *Complex information processing* (pp. 285-318). Hillsdale, NJ: Erlbaum.
- Hart, H. (1986). *Knowledge acquisition for expert systems*. New York: McGraw-Hill.
- Hoffman, R. R. (1987). The problem of extracting the knowledge of experts from the perspective of experimental psychology. *AI Magazine*, 8(2), 53-67.
- Jonassen, D. H. (1987). Assessing cognitive structure: Verifying a method using pattern notes. *Journal of Research and Development in Education*, 20(3), 1-13.
- Kelly, G. A. (1955). *The psychology of personal constructs*. New York: W. W. Norton.
- Kintsch, W. (1988). The role of knowledge in discourse comprehension: A construction-integration model. *Psychological Review*, 95, 163-182.
- Lave, J. (1988). *Cognition in practice*. Cambridge, UK: Cambridge University Press.
- Means, B., & Gott, S. P. (1988). Cognitive task analysis as a basis for tutor development: Articulating abstract knowledge representations. In J. Psozka, L. D., Massey, & S. A. Mutter (Eds.), *Intelligent tutoring systems: Lessons learned* (pp. 35-58). Hillsdale, NJ: Erlbaum.
- Nelson, W. A. (1989). Artificial intelligence knowledge acquisition techniques for instructional development. *Educational Technology Research and Development*, 37 (3), 81-94.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.
- Prerau, D. S. (1987). Knowledge acquisition in the development of a large expert system. *AI Magazine*, 8(2), 43-51.
- Schoen, D. A. (1983). *The reflective practitioner*. New York: Basic Books.
- Schvaneveldt, R. W. (Ed.). (1990). *Pathfinder associative networks: Studies in knowledge organization*. Norwood, NJ: Ablex.
- Stefik, M., Aikins, J., Balzer, R., Benoit, J., Birnbaum, L., Hayes-Roth, F., & Sacerdoti, E. (1983). Basic concepts for building expert systems. In F. Hayes-Roth, D. A. Waterman, & D. B. Lenat (Eds.), *Building expert systems*. Reading, MA: Addison-Wesley.

Suchman, L. A. (1987). *Plans and situated actions: The problem of human-machine communication*. Cambridge, UK: Cambridge University Press.

Wenger, E. (1987). *Artificial intelligence and tutoring systems*. Los Altos, CA: Morgan Kaufmann.