DOCUMENT RESUME

ED 335 003 IR 015 170

AUTHOR	Orey, Michael A.; Nelson, Wayne A.
TITLE	Using Intelligent Tutoring Design Principles To Integrate Cognitive Theory into Computer-Based Instruction.
PUB DATE	91
NOTE	14p.; In: Proceedings of Selected Research Presentations at the Annual Convention of the Association for Educational Communications and Technology; see IR 015 132.
PUB TYPE	Information Analyses (070) -- Reports - Research/Technical (143) -- Speeches/Conference Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	Behaviorism; *Computer Assisted Instruction; *Epistemology; Guidelines; *Instructional Design; Instructional Development; Instructional Systems; *Learning Processes; *Models
IDENTIFIERS	*Intelligent Tutoring Systems

ABSTRACT

Arguing that the evolution of intelligent tutoring systems better reflects the recent theoretical developments of cognitive science than traditional computer-based instruction (CBI), this paper describes a general model for an intelligent tutoring system and suggests ways to improve CBI using design principles derived from research in cognitive science rather than behavioral psychology. Differences between the two types of instructional systems are explained, and four major components incorporated in all intelligent tutoring systems are identified: (1) the user interface, which provides the means for two-way communication where the learner is engaged in some activity while the system is interpreting that activity in order to make a meaningful response; (2) the learner model, which is a representation of the errors or misconceptions that commonly occur when learners are exposed to the content; (3) the expert model, which is typically a database that represents the knowledge of an expert in the domain; and (4) pedagogic knowledge, which receives the results of a comparison of the learner's present knowledge state with knowledge in the expert model and makes decisions about what, when, and how information is to be communicated to the learner. Each of these components is then discussed in the context of design principles and general guidelines for designers of CBI derived from research in cognitive science and intelligent tutoring systems. It is concluded that, in order for developers of CBI to integrate these findings into their practice, a new perspective, which will require different assumptions about learning than the instructional design models currently in use, is necessary. (36 references) (BBM)

Title:

Using Intelligent Tutoring Design Principles to Integrate Cognitive Theory into Computer-based Instruction

Authors:

Michael A. Orey
Wayne A. Nelson

Using Intelligent Tutoring Design Principles to Integrate Cognitive Theory into the Design of Computer-Based Instruction

The field of computer-based instruction, like other areas of instruction, has been struggling to embrace a new theoretical base. Recently, cognitive psychology has made great progress in describing the structures and processes of human cognition, and many ideas for improving instruction by relying on cognitive learning theories have been proposed (Merrill, Li & Jones, 1990a; Hannafin & Rieber, 1989a; Hannafin & Rieber, 1989b; Low, 1980; Salomon, 1985; Wildman, & Burton, 1981; Winn, 1988). Glaser (1989) has noted that cognitive theories should now address the acquisition of structures and processes, stating that "[t]he study of learning can now take its cue from this knowledge [descriptive theories], and principled investigations of instruction can be a tool of major importance for the interactive growth of learning theory and its applications" (p. 38).

The problems we are having with integration of cognitive theories into the design of computer-based instruction stem from the models we use to design instruction. Current instructional design models, since they are founded in behavioral psychology, do not adequately support the kinds of design activities and decisions necessary for instruction based in cognitive conceptions of learning. Instructional systems design models employ a systematic process for the design of instructional systems, and prescribe various activities for development of computer-based instruction such as needs assessment, development of objectives, selection of strategies, and formative evaluation. The instructional product which results from using such models typically emphasizes a frame-based approach derived from behavioral theories of human learning. For a variety of reasons, such systems and theories are now seen as largely ineffective for instruction. In order to design a new generation of computer-based instruction which incorporates principles of cognitive learning, new models for the design of computer-based instruction are necessary to guide decision-making. In addition to fitting new ideas into old models, as has been suggested by Park, Perez, & Seidel (1987), or developing a completely new model, as proposed by Merrill, Li, and Jones (1990b), we believe it is more feasible to borrow and adapt a model which has already proven effective. That model is the intelligent tutoring system.

Unlike traditional computer-based instruction, the evolution of intelligent tutoring systems better reflects the recent theoretical developments of cognitive science. Various forms of computer-based instruction today are incapable of performing well from a cognitive perspective because the design models utilized are not conducive to the development of flexible instruction which can adapt to the learner as the instruction occurs. As Winn (1988) has noted, traditional instructional design models require that all decisions about instruction be made and tested before the instruction is implemented. Winn (1989) suggests, therefore, that intelligent tutoring systems may be the instructional medium of the future because of the dynamic cognitive modeling capabilities which allow the system to make adaptive instructional decisions as the learner uses the system. In the following pages, a general model for an intelligent tutoring system is described, along with suggestions for the improvement of computer-based instruction using design principles derived from research in cognitive science.

A Model of an Intelligent Tutoring System

Intelligent tutoring systems derive historically from computer-based instruction, but since there are basic differences in theoretical perspectives, they represent two poles along a continuum of computer-based instructional systems. Differences between the two types of instructional systems are primarily structural characteristics: intelligent tutoring systems encode knowledge, while computer-based instruction encodes instructional decisions based on knowledge (Wenger, 1987). Subject matter is separated from teaching method in an intelligent tutoring system, thereby separating content from presentation techniques (Clancey, 1984). Intelligent tutoring systems are based on the idea that natural learning occurs through context-based performance, and the goal of research with intelligent tutoring systems is to develop programs which understand student misconceptions and provide appropriate instruction as a master teacher would (Loser & Kurtz, 1989).

The main advantage of the intelligent tutoring systems model, however, is that the underlying assumptions about learning and instruction differ from those of current instructional design models. Regardless of the particular model (c.f. Hartley & Sleeman, 1973; Hayes-Roth & Thorndyke, 1985; Park & Seidel, 1989; Wenger, 1987), all intelligent tutoring systems incorporate at least four major components: 1) the user interface; 2) the learner model; 3) the expert model, and; 4) pedagogic knowledge. As depicted in Figure 1, the learner and expert models are often combined into an area where diagnosis occurs through comparison between correct knowledge states (the expert model) and incorrect knowledge states (the learner model).

The *interface* provides the means for a two-way communication, where the learner is engaged in some activity while the system is interpreting the learner's activity so that a meaningful response can be made. The *expert model* is typically a database of correct knowledge states for a given domain that is organized in some form of declarative or procedural knowledge that represents the knowledge of an expert in the domain. In many intelligent tutoring systems, expert knowledge may be supplied to the system, generated dynamically, or both. The *learner model* is a representation of the errors or misconceptions that commonly occur when learners are exposed to the content, but may also include other information such as a curriculum map of the learner's sequence through the curriculum. During an instructional session, the system monitors the performance of the learner, attempting to ascertain the knowledge that the learner possesses. This diagnostic process is accomplished by comparing the learner's present knowledge state with knowledge in the expert model. The results of the comparison are then passed to the *pedagogical model*, where decisions are made about what, when, and how information is to be communicated to the learner.

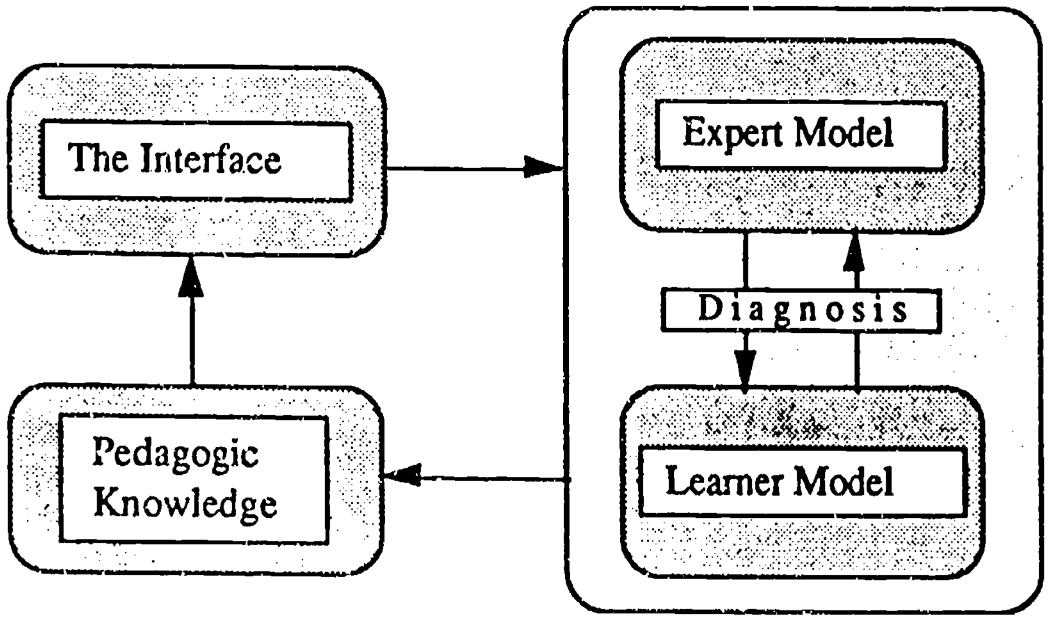

Figure 1. Components of an Intelligent Tutoring System.

Design Principles for Computer-Based Instruction

It is our contention that many of the aspects of the components of an intelligent tutoring system presented in Figure 1 can be included in any type of computer-based instructional software. Regardless of the type of software being developed, the interface should be given prime consideration, and the content needs to be clearly defined and represented (the expert model). The learner is always considered, regardless of the delivery system (the learner model) and instructional software must contain knowledge about how to provide instruction (the pedagogical model). Therefore, the design and development of computer-based instruction and intelligent tutoring systems are not vastly different. The following discussion, while not intended to be an exhaustive review of the literature, presents general guidelines for designers of computer-based instruction derived from research in cognitive science and intelligent tutoring systems.

The interface

Many principles based on cognitive theories have been proposed for user interface design. These priciples constitute the area of human-computer interaction (c.f. Card, Moran, & Newell, 1983; Norman & Draper, 1986; Shneiderman, 1987). The major problem in designing a good interface is the delay in communication with the learner that occurs between the time the system is designed and the time the learner actually uses the system. The designer must anticipate the possible actions a learner may take, and devise ways to handle the actions before the learner ever interacts with the software. This requires attention to aspects of learning, task performance, subjective satisfaction, and retention of information related to operation of the interface. Many researchers suggest that a good interface needs to be designed using some kind of iterative process for development, testing, and revision of the interface components (Shneiderman, 1987).

It is important to remember that the learner is not only learning the content, but is also learning how to operate the software. Therefore, one of the primary considerations in designing a user interface should be ease of use. Screen design, use of special function keys, menu selection, and feedback on errors are just some of the aspects which need to be considered (c. f. Jay, 1983). The key in designing these and other components is consistency. The learner should be able to rely on the same body of knowledge about operating the software as they move from one screen to another or, ideally, from one program to another. Consistent interface design will also help to reduce the cognitive load on the learner (Norman & Draper, 1986; Shneiderman, 1987), thereby freeing more processing capacity for the learning task and minimizing interference between the learning task and operation of the software (Anderson, Boyle, & Reiser, 1985; Anderson & Reiser, 1985).

Interaction style is also an important part of interface design. An interaction style which is consistent with the learners knowledge of computers is most desirable. There are many possibilities, including command-driven, menu selection, and direct manipulation interfaces. Depending on the learner's expertise, one style might be more appropriate than another. For example, if the learner is a skilled computer user, a command-driven interaction style may be the most appropriate, but if the learner is relatively inexperienced, direct manipulation of objects with a mouse may be more desirable. The selection of an interaction style should also be made based on the context of the learning activities. Menu-driven software may not be appropriate for many tasks which involve simulations, problem solving, or similar learner activities.

Metaphor can also be a powerful device in an effective interface, providing students with "comparisons which can help them learn" (Carroll & Mack, 1985; p. 40). Much of the applications software currently being marketed makes use of metaphor to help users be more productive. We have windows, trash cans, desktops, buttons, cards, folders, and pages. Such metaphors enable learners to build on their past experiences, and to map concepts in a well known domain to similar concepts and relations in a new domain. Complex metaphors may require sophisticated graphics capabilities and direct manipulation, but the benefits to the learners are worth the efforts. There are many metaphors in the classroom which are familiar to students, and which could be incorporated into an effective interface for learning.

The interface is not only a means for input and output of information, but can also supply important data about the learners. Depending on the domain, data from the interface can be used to monitor the learning process as it unfolds. If the content is process-oriented, the interface should be designed to monitor the learner's progress. For example, Orey and Burton (1990) designed an interface which supplied the system with more information about the child's subtraction process than just the answer to a problem. These data allowed the system to make decisions about the nature of the learner's subtraction errors. If the content is declarative, the interface should facilitate retention and organization. For example, the way in which an expert organizes knowledge in the field of her expertise might be depicted in some graphical form such as a content map, with the intention that the learner will develop a similar organization in obtaining domain proficiency.

The Expert Model, the Learner Model and Diagnosis

Experts in a domain organize knowledge and control problem-solving processes differently than novices (Glaser, 1989). The goal of instruction from a cognitive perspective, then, should be to replicate the knowledge structures and processes of the expert in the mind of the learner (Wildman & Burton, 1981). In order to do so, the domain expert's knowledge must be mapped into symbols a computer can store and manipulate, and presented to the learner in an organized manner. While there are many types of knowledge representation schemes, including semantic networks, production systems, scripts and/or frames, the appropriate form for knowledge representation is determined, in part, by the kinds of diagnostic procedures being implemented.

Van Lehn (1988) describes the notion of bandwidth in diagnosis, which is the correspondence between observable actions of the learner and mental states within the learner. High bandwidth is the ideal case of a one-to-one correspondence between the learner's mental states and observable actions. Low bandwidth, on the other hand, exists when there are multiple mental states between the observable actions performed by the learner. Where the system fits on the continuum between high bandwidth and low bandwidth is largely determined by the interface. For example, Orey and Burton (1990) used a process oriented interface to capture as many actions as possible while the learner performed whole number subtraction. On the other hand, Brown and Burton (1978) used a system that only examined the completed answers to an entire subtraction test. In the latter case, there are a multitude of mental states that could occur before the system is able to diagnose errors. As a result of the differences in bandwidth diagnostic levels between these two systems, the diagnostic strategies and knowledge representation were also entirely different.

The designer of computer-based instruction can make use of knowledge representation and diagnostic principles when implementing high bandwidth diagnosis. First, an interface to support this level of diagnosis needs to be designed. Second, the most common types of errors that may occur need to be identified in order to construct a partial "model" of the learner that can be used for diagnosis. The type of error made by the learner can then be determined with extended conditional statements that are checked at the point of the error (context-specific error checking). In this way, a "mini" high bandwidth diagnostic system which uses an approximation of a production system can be implemented. For example, a system for teaching hypothesis testing in statistics could present a problem, and then allow the learner to plug values from the problem statement into formulas. If the learner attempts to place a value in an incorrect position in the formula, the system could respond with appropriate instruction.

Simulation software could also benefit from monitoring and diagnosis of user errors. For example, suppose the simulation involves the operation of a nuclear power plant. The instantiation of a major error might result in the melt down of the simulated power plant, which may have dramatic effects on the learner's memory of the error. The error which caused the melt down, if properly communicated to the user, would hopefully not occur in the real world environment of running the power plant.

The Pedagogical Model

The pedagogical model contains knowledge necessary for making decisions about the teaching tactics available to the system. Since there tends to be considerable overlap between the functions of the various components of an intelligent tutoring system, the decisions and actions of the pedagogical model are highly dependent on the results of the diagnostic process. In general, the pedagogical model must decide when to present information to the learner, how to present the information, and what information to present. There have been a variety of pedagogical approaches employed in intelligent tutoring systems, but most of the current systems tend to implement only one pedagogical strategy. Intelligent tutoring systems, therefore, do not yet possess a rich repertoire of tutoring strategies from which to select. This deficiency exists, in part, because so little is known about how to teach, especially in a one-to-one setting common in tutoring (Ohlsson, 1987).

In general, pedagogical strategies are dependent on the overall context of the learning environment embodied in the intelligent tutoring system. Three general types of environments have been implemented: systems which monitor student activity within a problem-solving domain, systems which employ mixed-initiative dialogue between learner and tutor, and systems which employ guided discovery or coaching (Wenger, 1987). The choice of tutoring environment is dictated by the nature of the content to be learned, the knowledge and experience of the learner, and the assumptions about learning inherent in the underlying theory on which the system is based.

Current research with intelligent tutoring systems primarily focuses on context-based environments where students learn by working on specific, real-world problems, because cognitive theories have shown that knowledge and expertise is acquired by active application of knowledge during problem solving (Glaser, 1989). Further, learning strategies which make use of modeling of specific and appropriate procedures and strategies for problem-solving, and which strengthen existing knowledge through practice that minimizes error, have been successfully applied in intelligent tutoring systems research (Anderson, Boyle, & Reiser, 1985; White & Fredricksen, 1990). While these strategies are not significantly different from those already employed in some computer-based instruction, the use of context-based strategies supporting situated cognition (Brown, Collins, & Duguid, 1989) is becoming an important factor in the development of intelligent tutoring systems. Such strategies are not emphasized in current computer-based instruction, especially drill and practice and tutorial software.

Within a particular learning environment, a variety of teaching tactics can be implemented. Some of the possible tactics are presented in Table 1. Interested readers are referred to Wenger (1987), Ohlsson (1987), and Winne (1989) for more detailed discussions of some common teaching tactics utilized in intelligent tutoring systems. The selection and use of teaching tactics are governed by the characteristics of the learning environment, along with the general pedagogical approach. Various teaching tactics can be selected using an opportunistic approach (results of diagnosis reveal when opportunities for intervention are appropriate), or a plan-based approach (the focus of diagnosis is to monitor teaching plans and goals). Of course, a human tutor probably uses some combination of opportunistic and plan-based interventions, and research is currently focusing on the development of more sophisticated pedagogical techniques combining aspects of both pedagogical approaches (Wenger, 1987).

Table 1. *Some teaching tactics employed in intelligent tutoring systems.*

Presenting Information	Monitoring Performance	Detecting Errors
Socratic dialogue	Guided practice	Reveal errors as occur
Demonstration	Annotated practice	Marking for explanation
Priming	Hints	Probability thresholds
Associative Links	Prompt self-review	Bug repair
Curriculum Maps	Prompt self-annotation	
Issues and Examples	Evaluate hypotheses	
Answer Questions	Coaching	
	Interactive simulation (Microworld)	

Many opportunities for integrating pedagogical strategies derived from research on intelligent tutoring systems are available for designers of computer-based instruction. For example, the work on conceptual models, as discussed by Mayer (1989), can provide an effective framework for applying cognitive theories of learning in computer-based instruction systems. As an illustration of this approach, consider the study conducted by Wertheimer (1959) which examined the differences between teaching children the formulae for the area of different geometrical shapes versus teaching children the conceptual explanation of those same formulae (see Figure 2).

Initially the learner is given a paper parallelogram and a pair of scissors (Step 1 of Figure 2). The child is then instructed to cut off the triangle on one side of the parallelogram (Steps 2 and 3) and place it on the other side (Step 4). From this experience, the learner understands that the formula for the area of a parallelogram is $A=B\bullet H$ (where A is the area, B is the base and H is the height, or in Figure 2, $A=7\bullet 3=21$). Direct manipulation or animation could be used to implement this strategy in computer-based instruction (Orey, 1985). The learner could be coached through the process, learning to "grab" the triangle with a mouse and place it on the other side of the parallelogram. Such an approach allows the learner more control of the environment, and is consistent with the pedagogical strategies employed in intelligent tutoring systems.

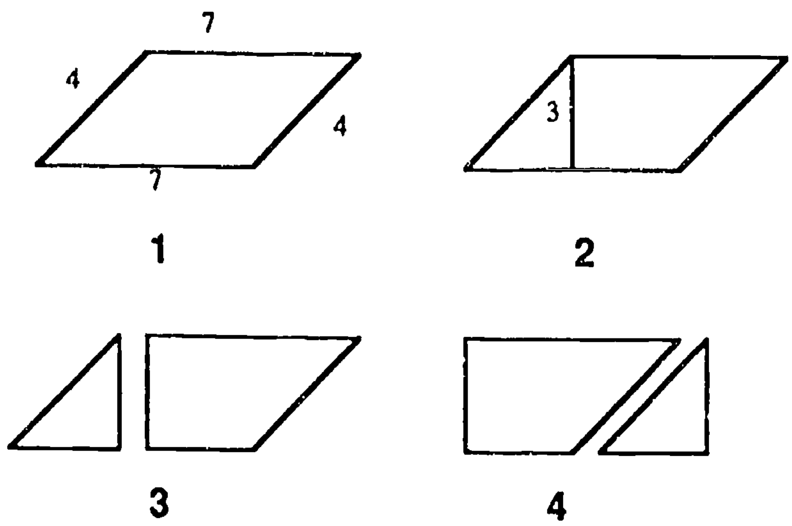

Figure 2. Wertheimer's (1959) conceptual model for teaching area.

Pedagogical knowledge, then, can be as important in computer-based instruction as it is in an intelligent tutoring system. Numerous decisions need to be made regarding teaching tactics, sequencing and media (graphics, text, sound, or video). Certainly, more traditional approaches to computer-based instruction can be used as components of the pedagogical knowledge base (see Jonassen, 1988). Metacognitive considerations may also be incorporated within the pedagogical knowledge of a computer-based instructional system. That is, communication with the learner can be used to induce reflection on the learning process or to suggest alternative problem-solving strategies (Brown, 1975). In whatever manner, it is important for the designer to remember that learners are capable of controlling their own learning to some extent (Linn & Clancey, 1990; Winne, 1989).

Conclusions

We have sketched an approach to the design of computer-based instruction derived from cognitive science, in particular, the design principles used in intelligent tutoring systems. As our analysis unfolded, we were struck by the idea that there were many similarities between intelligent tutoring systems and computer-based instruction that has been developed from a cognitive perspective. That is to say, rather than a dichotomy existing where the two poles are computer-based instruction and intelligent tutoring systems, actually there is a continuum anchored at one end by traditional computer-based instruction developed from an instructional systems design perspective (such as that found in many training settings), and at the other end by the "ideal" intelligent tutoring system. Some intelligent tutoring systems are not quite the ideal, and some computer-based instruction has been developed from a cognitive perspective. These systems overlap in the middle of the continuum.

In order for developers of computer-based instruction to integrate the findings of cognitive science into their practice, however, a new perspective is necessary. This perspective will require different assumptions about learning than those embodied in instructional design models currently used for designing computer-based instruction; assumptions that form the basis of the intelligent tutoring systems reviewed in the previous pages. It is time to apply these principles to the design and testing of computer-based instruction in real learning settings.

References

Anderson, J. R., Boyle, C. F. & Reiser, B. J. (1985). Intelligent tutoring systems. *Science*, *228*, 456-462.

Anderson, J. R. & Reiser, B. J. (1985). The lisp tutor. *Byte*, *10*(4), 159-175.

Brown, A. L. (1975). The development of memory: Knowing, knowing about knowing, and knowing how to know. In H. W. Reese (Ed.), *Advances in Child Development and Behavior*. Vol 10. New York: Academic Press.

Brown, J.S., & Burton, R.R. (1978). Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science, 2,* 155-192.

Brown, J. S., Collins, A., & Duguid, P. (1989). Situated cognition and the culture of learning. *Educational Researcher, 18*(1), 32-42.

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

Carroll, J. M., & Mack, R. L. (1985). Metaphor, computing systems, and active learning. *International Journal of Man-Machine Studies, 22*, 39-57.

Clancey, W. J. (1984). Methodology for building an intelligent tutoring system. In W. Kintsch, J. R. Miller, & P. G. Polson (Eds.), *Method and tactics in cognitive science.* (pp. 51-83). Hillsdale, NJ: Erlbaum.

Hannafin, M. J., & Rieber, L. P. (1989a). Psychological foundations of instructional design for emerging computer-based instructional technologies: Part I. *Educational Technology Research and Development, 37*(2), 91-101.

Hannafin, M. J., & Rieber, L. P. (1989b). Psychological foundations of instructional design for emerging computer-based instructional technologies: Part II. *Educational Technology Research and Development, 37*(2), 102-114.

Hartley, J. R. & Sleeman, D. H. (1973). Towards intelligent teaching systems. *International Journal of Man-Machine Studies, 5*, 215-236.

Hayes-Roth, B., & Thorndyke, P. W. (1985). Paradigms for intelligent systems. *Educational Psychologist, 20*(4), 231-141.

Jay, T. B. (1983). The cognitive approach to computer courseware design and evaluation. *Educational Technology, 23*(1), 22-26.

Jonassen, D. H. (Ed.). (1988). *Instructional designs for microcomputer courseware.* Hillsdale, NJ: Erlbaum.

Linn, M. C., & Clancey, M. J. (1990). Designing instruction to take advantage of recent advances in understanding cognition. *Academic Computing, 4*(7), 20-23.

Loser, M., & Kurtz, B. (1989). Quadratic Grapher: An intelligent tutoring system for graphing quadratic equations. In H. Maurer (Ed.), *Computer Assisted Learning: Proceedings of the 2nd International Conference on Computer Assisted Learning.* (pp. 346-358). Berlin: Springer-Verlag.

Low, W. C. (1981). Changes in instructional development: The aftermath of an information processing takeover in psychology. *Journal of Instructional Development, 4*(2), 10-18.

Mayer, R. E. (1989). Models for understanding. *Review of Educational Research, 59*(1), 43-64.

Merrill, M. D., Li, Z., & Jones, M. K. (1990a). Limitations of first generation instructional design. *Educational Technology, 30*(1), 7-11.

Merrill, M. D., Li, Z., & Jones, M. K. (1990b). Second generation instructional design. *Educational Technology, 30*(2), 7-14.

Norman, D. A., & Draper, S. W. (Eds.). (1986). *User centered system design.* Hillsdale, NJ: Erlbaum.

Ohlsson, S. (1987). Some principles of intelligent tutoring. In R. W. Lawler & M. Yazdani (Eds.), *Artificial intelligence and education (Vol. 1).* Norwood, NJ: Ablex.

Orey, M. A. (1985). *Geometric area.* Unpublished software program.

Orey, M. A., & Burton, J. K. (1990). POSIT: Process oriented subtraction-interface for tutoring. *Journal of Artificial Intelligence in Education, 1*(2), 77-105.

Park, O., Perez, R. S., & Seidel, R. J. (1987). Intelligent CAI: Old wine in new bottles, or a new vintage? In G. Kearsley (Ed.), *Artificial Intelligence & Instruction: Applications and Methods*. Reading, MA: Addison Wesley.

Park, O., & Seidel, R. J. (1989). A multidisciplinary model for development of intelligent computer-assisted instruction. *Educational Technology Research and Development, 37*(3), 72-80.

Salomon, G. (1985). Information technologies: What you see is not (always) what you get. *Educational Psychologist, 20*(4), 207-216.

Shneiderman, B. (1988). *Designing the user interface: Strategies for effective human-computer interaction*. Reading, MA: Addison-Wesley.

Van Lehn, K. (1988). Student modeling. In M. Polson and J. Richardson (Eds.), *Foundations of Intelligent Tutoring Systems*. (pp. 55-78) Hillsdale, NJ: Erlbaum.

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

Wertheimer, M. (1959). *Productive thinking*. New York: Harper and Row.

White, B. Y., & Fredricksen, J. R. (1990). Causal model progressions as a foundation for intelligent learning environments. *Artificial Intelligence, 42*(1), 99-157.

Wildman, T. M., & Burton, J. K. (1981). Integrating learning theory with instructional design. *Journal of Instructional Development*. *4*(3), 5-14.

Winn, W. (1988, January). *The theoretical foundations of educational technology and future directions for the field*. Paper presented at the annual meeting of the Association for Educational Communications and Technology. New Orleans, LA.

Winn, W. (1989). Toward a rationale and theoretical basis for educational technology. *Educational Technology Research and Development*. *37*(1), 35-46.

Winne, P. H. (1989). Theories of instruction and of intelligence for designing artificially intelligent tutoring systems. *Educational Psychologist, 24*(3), 229-259.