The Intellectual Content of Instructional Design

Wayne A. Nelson,
Susan Magliaro,
and
Thomas M. Sherman
Division of Curriculum and Instruction
College of Education
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061

Abstract. In this paper, instructional design is described as a high-level thinking process. This characterization provides more information regarding the way instructional design is learned and actually practiced in three ways. First, analyzing instructional design from the perspective of cognitive psychology makes possible classification of design models along a continuum from general heuristics to specific algorithms. Second, the literature on planning, schema theory, development of expertise and metacognition supports the notion that instructional design models represent designers' schemata. Third, the development of design schemata can be traced over time to chart the similarities between cognitive abilities of designers. The implications drawn from this analysis can enhance understanding of the design process as well as improve strategies to teach instructional design.

Over the past three decades, instructional design has been characterized in a myriad of models which range from very specific, step-by-step algorithms to general frameworks or heuristics (e.g., Gagne & Briggs, 1979; Sherman, 1980). While these seemingly dichotomous viewpoints appear in various forms in the extant literature (see Andrews & Goodson, 1980, for a review), a reconceptualization of instructional design as a complex intellectual activity similar to other known sophisticated thinking strategies may lead to a more productive understanding of essential processes and strategies. The one ingredient which is consistent in all these models is that their purpose is to guide the decisions that designers must make during the design process.

Extending Simon's (1973) definition of design in general, we propose that instructional design is "a system of productions, in which the elements already evoked from memory and the aspects of the design already arrived at up to any given point would serve as the stimuli to evoke the next set of elements" (p. 190).

From this perspective, the steps and processes of instructional design may be appropriately considered as cognitive tools which must be acquired, developed, selected, and managed through the complex decision-making activities that constitute instructional design. By comparing instructional design to other thinking strategies such as problem solving and decision making, the research and theories of cognitive science can be employed to generate implications and testable hypotheses. Such comparisons may also provide a better understanding of how the cognitive operations which instructional designers use can be learned and developed.

Instructional Design and Complex Thinking Strategies

For many years, those who investigate high-level thinking skills have represented "problems" on a continuum anchored on the one end with "well-defined" and on the other end with "ill-structured" (Greeno, 1976; Reitman, 1965). Well-defined problems

Because designers rarely have clear goals or structured behaviors to solve problems, they typically employ heuristics to guide, organize, or pattern problem solutions.

tend to be solved using an established set of procedures expressed as rules or algorithms. These problems have clear goals and learning to solve them is often a function of time and practice. In many cases, the ultimate ability to solve well-defined problems is functional automaticity in which solution actions operate without conscious attention. In contrast, ill-structured problems have no clear goal or specific solution processes. Consequently, these problems present a different, and perhaps greater, intellectual challenge.

All instructional design problems fall somewhere along this continuum of well-defined to ill-structured (see Figure 1); however, "design" problems by and large fall toward the ill-structured anchor. Because designers rarely have clear goals or structured behaviors to solve problems, they typically employ heuristics to guide, organize, or pattern problem solutions. Similarly, scientists and artists, among others, lack specific goals and procedures to produce products. As a result, they employ heuristics, such as the scientific method, problem solving, creativity, and decision making, which provide a general model of the processes they use. Whether these heuristics operate like algorithms or serve as a theoretical schema for systematic problem exploration has been the subject of controversy (e.g., Carroll, Thomas, & Malhotra, 1980). Nonetheless, within the domain of instructional design, little attention has been given to identifying and explaining the intellectual activities associated with designing instruction.

Instructional Design Thinking

In many respects, the structure or components of instructional design are similar to those of general problem solving (see Polya, 1957). Instructional design begins with understanding the problem by establishing goals, analyzing needs and learner characteristics, and identifying the scope and content of the subject. Once the "problem space" has been defined, the designer breaks the problem into a series of subgoals that, if reached, will solve the problem. These component activities can include potential actions such as identifying and sequencing objectives, devising activities, choosing materials, and planning evaluations. In this way, an ill-structured, global design problem with broad definitions becomes more structured and specific (Simon, 1973).

In reviewing the literature, we found that design models, like problems, could be distributed along a continuum ranging from broadly defined, general heuristics to specific step-by-step algorithms. General models (e.g., Wildman, 1980) provide an overall, yet abstract, sequence of mental processes or strategies. On the other hand, algorithmic models (e.g., Annarino, 1983) are step-by-step procedures that are developed for a specific content area and outline each incremental design activity in a very concrete manner.

These examples illustrate relatively well-defined points on this instructional design continuum. Others, such as Scandura (1983) and Landa (1976) present positions which differ in specificity and structure and may be less reliably placed on such a continuum.

On the surface, design models give the designer a set of actions for planning by providing an external framework to guide the design process. But at a deeper level, design models represent a mental framework or schema of a designer's internal organization of knowledge and information about instructional design. Consequently, in addition to depicting instructional design as a product-oriented activity, instructional design models should also be considered to represent intellectual processes. These cognitive activities of designing instruction are epitomized by the interactions between the complexity of the designer's cognitive structures , the instructional design task, and the environment (see Figure 2).

Theoretical Background

Though little research on the thought processes of instructional designers exists, there is considerable evidence from cognitive psychology which supports the importance of the relationship between cognitive activities and behavioral outcomes (Ericsson & Simon, 1984). In a study of instructional designers' thought processes, Nelson (1987) compared the design models and knowledge of experts and novices. Not surprisingly, differences between experts and novices mirrored the findings from previous expert/novice comparison studies in such domains as

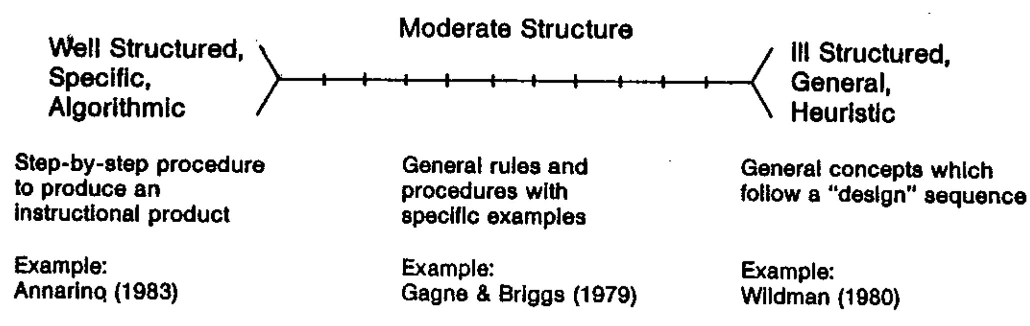

Figure 1. A continuum illustrating the relative characteristics of the structure of specific vs. general instructional designs.

chess (e.g., Chase & Simon, 1973), electronics (e.g., Egan & Schwartz, 1979), and physics (e.g., Larkin, McDermott, Simon, & Simon, 1980). That is, in comparison to novices, experts' knowledge structures are more highly organized and well-integrated, and experts are able to respond to and complete tasks more quickly and systematically. The literature on planning, schema theory, development of expertise, and metacognition provides an extensive analysis of the mental activities associated with complex intellectual tasks which may also add to our understanding of the thought processes of expert instructional designers.

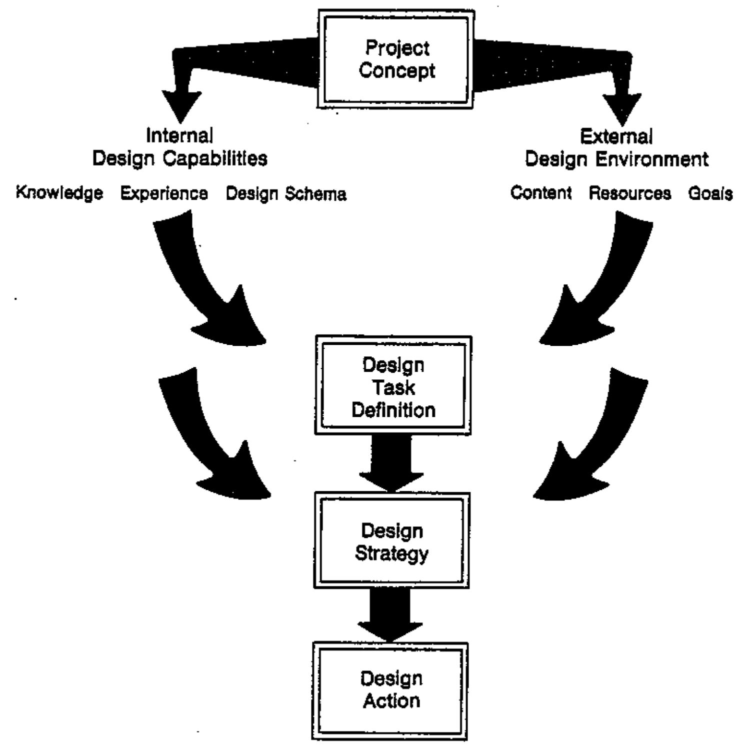

Figure 2. A representation of the interaction between the designer and the design environment.

Planning

Using their theory of human problem solving, Newell and Simon (1972) compared planning to the use of an algorithm in solving a problem. Specifically, plan construction was viewed as an auxiliary activity that aids in the solution of a problem by guiding the action in the problem space. Newell and Simon proposed that a plan is generated by: (1) abstracting details from original objects and operators; (2) forming a similar abstract problem; (3) solving this abstract problem and using the solution to provide a plan for solving the original problem; and (4) translating the plan back to the original problem space for execution.

This planning process has been modeled on computers by several researchers (Newell & Simon, 1972; Sacerdoti, 1974). Called automatic planning systems, these models operate by a process of successive refinement in which a solution is generated, compared to a standard, modified, compared again, and so on until a good (or acceptable) outcome is achieved. Though Fikes (1977) reported several methods used to represent knowledge in these systems, typically a goal state is specified and the planner matches the goal to an action that will achieve some portion of the goal. The goal is then reconsidered in light of that new action. This kind of planning process is called "top-down" because solutions are progressively developed by moving from general to specific states of abstraction.

Hayes-Roth and Hayes-Roth (1979) propose an alternative model of planning that is less rigid than the top-down process just described. Their computer-implemented planning system works at different levels of abstraction and can adopt different planning methods depending on the given problem. Decisions are made by specialists at a variety of abstraction levels including executive, meta-planning, plan-abstraction, knowledge-base, and plan levels. The planning process is controlled by the executive which selects a specialist to generate and record a decision. Depending on the specialist chosen, decisions can involve any of the five levels of abstraction. The cycle continues until a complete plan exists or until the existing plan satisfies the given criteria.

Hayes-Roth and Hayes-Roth (1979) contend that this planning method is similar to actual human planning, where decisions regarding the solution are made at a variety of levels and not in top-down fashion. In general, however, these computer models portray planning as an information-enriching activity which involves sorting and comparing multiple alternatives. The essential action is relating generated alternatives to desired goal states and judging their adequacy. Computer planning programs are often effective because of their capability to quickly and tirelessly rate the salience of information or cues and to compare alternatives; consequently, efficiency is not a problem. Humans, however, have neither the memory power nor the immediate access characteristics of computers. Thus, conceptual schemata or knowledge structures become critical components of planning in order to manage and control cognitive operations with any precision.

For example, ordinarily designers organize a design task with either a general heuristic or an algorithm. The conceptual structure serves as a means to extract needed information and relate the information to intended outcomes. Through an iterative process of

Despite having no content expertise, the designer has a set of representations based on a "design model" which can guide the development of an effective training program.

consideration and reconsideration, designers can make judgments about what information is available, needed, and missing. In many ways, the conceptual structure is a design template into which decision information is categorized, considered, and stored. Without such a schema, the design task could be overwhelming due to the volume of information associated with training on a complex task (Holmes & Sherman, 1987). Thus, the design schema provides a conceptual structure for the design task which facilitates efficient and effective management of relevant information.

Schema

According to schema theory, as people work through problems they constantly compare the knowledge stored in memory (organized in schemata) to current situations and use the comparisons to decide what to do (Mayer, 1983). If prior knowledge and experience are relevant to the task (i.e., well-defined), the problem can be comparatively easy to solve. But if individuals have no specific knowledge available with respect to the task, then they have no schemata for making evaluative judgments. In this case, general knowledge acquired in previous situations which is organized in nonspecific schemata must be accessed. These nonspecific schemata define general principles of action often expressed as heuristics.

The notion of schemata as basic, stereotypical descriptions of general knowledge was first proposed by Bartlett (1932). More recently, Alba and Hasher (1983) have elaborated the definition of a schema as a framework which "selects and actively modifies experiences in order to arrive at a coherent, unified, expectation-confirming and knowledge-consistent representation of an experience" (p. 203).

Schank and Abelson described schemata or scripts as stereotypical representations of common events, activities, or settings. Scripts may contain general information about a setting and specific information about the activities which may occur in that setting. Schank and Abelson (1977) have used scripts as the central organizational element in their research on computer understanding of natural human language.

Using this script theory, Bower, Black, and Turner (1979) studied understanding and memory of narrative text passages. Their findings indicated that subjects generally agreed on the conventional roles, props, and sequences of standard activities such as eating in a restaurant or going to a concert. Moreover, comprehension and recall of narrative passages was shown to be related to the degree to which a passage resembled an existing script. Thus, these scripts exert a powerful influence on comprehension. In effect, comprehension is a constructive activity rather than a background task of "taking meaning off a page."

Consider the position of instructional designers developing a training program on tasks which are unfamiliar to them. Despite having no content expertise, the designer has a set of representations based on a "design model" which can guide the development of an effective training program. By following this script of cognitive behavior, the designer can "think through" design procedures and initiate design activities consistent with the design model.

Jeffries, Turner, Polson, and Atwood (1981) examined the role of schemata in the design of computer software. They contend that a general design schema for software designers exists which contains abstract knowledge about the overall structure of a design as well as the processes involved in generating that design. They assume that, at least for expert software designers, design schemata would be similar, although individual differences between experts could exist because of variations in software design experience. These schemata appear very similar to what are called design models. Based on schema theory, we can expect the complexity and generality of designers' schemata to influence design conceptualizations, activities, and outcomes. We can also expect these schemata to change with experience, time, and reflection.

Development of Expertise

Piaget (1970) proposed that cognitive structures or schemata change as individuals encounter different situations and tasks. He believed that schema development follows an orderly evolution through various stages which represent qualitative change to progressively more sophisticated levels of reasoning. These changes, rather than being isolated, permeate and reorganize many different schemata. The individual plays an important role in this process by comparing new and old schemata, recognizing changes based on new experience, and actively reorganizing schemata to promote new understanding.

In a study by Pitt (1983), problem-solving schemata were shown to be essentially the same for adolescents and adults. However, significant developmental differences between age groups

Despite having no content expertise, the designer has a set of representations based on a "design model" which can guide the development of an effective training program.

consideration and reconsideration, designers can make judgments about what information is available, needed, and missing. In many ways, the conceptual structure is a design template into which decision information is categorized, considered, and stored. Without such a schema, the design task could be overwhelming due to the volume of information associated with training on a complex task (Holmes & Sherman, 1987). Thus, the design schema provides a conceptual structure for the design task which facilitates efficient and effective management of relevant information.

Schema

According to schema theory, as people work through problems they constantly compare the knowledge stored in memory (organized in schemata) to current situations and use the comparisons to decide what to do (Mayer, 1983). If prior knowledge and experience are relevant to the task (i.e., well-defined), the problem can be comparatively easy to solve. But if individuals have no specific knowledge available with respect to the task, then they have no schemata for making evaluative judgments. In this case, general knowledge acquired in previous situations which is organized in nonspecific schemata must be accessed. These nonspecific schemata define general principles of action often expressed as heuristics.

The notion of schemata as basic, stereotypical descriptions of general knowledge was first proposed by Bartlett (1932). More recently, Alba and Hasher (1983) have elaborated the definition of a schema as a framework which "selects and actively modifies experiences in order to arrive at a coherent, unified, expectation-confirming and knowledge-consistent representation of an experience" (p. 203).

Schank and Abelson described schemata or scripts as stereotypical representations of common events, activities, or settings. Scripts may contain general information about a setting and specific information about the activities which may occur in that setting. Schank and Abelson (1977) have used scripts as the central organizational element in their research on computer understanding of natural human language.

Using this script theory, Bower, Black, and Turner (1979) studied understanding and memory of narrative text passages. Their findings indicated that subjects generally agreed on the conventional roles, props, and sequences of standard activities such as eating in a restaurant or going to a concert. Moreover, comprehension and recall of narrative passages was shown to be related to the degree to which a passage resembled an existing script. Thus, these scripts exert a powerful influence on comprehension. In effect, comprehension is a constructive activity rather than a background task of "taking meaning off a page."

Consider the position of instructional designers developing a training program on tasks which are unfamiliar to them. Despite having no content expertise, the designer has a set of representations based on a "design model" which can guide the development of an effective training program. By following this script of cognitive behavior, the designer can "think through" design procedures and initiate design activities consistent with the design model.

Jeffries, Turner, Polson, and Atwood (1981) examined the role of schemata in the design of computer software. They contend that a general design schema for software designers exists which contains abstract knowledge about the overall structure of a design as well as the processes involved in generating that design. They assume that, at least for expert software designers, design schemata would be similar, although individual differences between experts could exist because of variations in software design experience. These schemata appear very similar to what are called design models. Based on schema theory, we can expect the complexity and generality of designers' schemata to influence design conceptualizations, activities, and outcomes. We can also expect these schemata to change with experience, time, and reflection.

Development of Expertise

Piaget (1970) proposed that cognitive structures or schemata change as individuals encounter different situations and tasks. He believed that schema development follows an orderly evolution through various stages which represent qualitative change to progressively more sophisticated levels of reasoning. These changes, rather than being isolated, permeate and reorganize many different schemata. The individual plays an important role in this process by comparing new and old schemata, recognizing changes based on new experience, and actively reorganizing schemata to promote new understanding.

In a study by Pitt (1983), problem-solving schemata were shown to be essentially the same for adolescents and adults. However, significant developmental differences between age groups were found in how problems were defined and in the use of problem-solving strategies. Adults were shown to use qualitatively different subroutines and deductive strategies for problem solving. For example, Pitt gave a chemistry problem to 10th graders and college juniors who had no significant background in chemistry. By measuring the subroutines of the solution process for each subject, Pitt showed that, although the 10th-grade students followed the same general heuristics and strategies as the older group, they "defined problems inadequately and generated ill-formed, inaccurate hypotheses," as well as possessed "limited ability to coordinate information or operations" (p. 547).

Other studies have also demonstrated significant differences in problem-solving abilities between experts and novices in such domains as chess (Chase & Simon, 1973), physics (Chi, Glaser, & Rees, 1982; Larkin, McDermott, Simon, & Simon, 1980), political cognition (Fiske, Kinder, & Larter, 1983), and computer programming (McKeithen, Reitman, Rueter, & Hirtle, 1981). In general, these studies show differences in task performance and problem representation based on experience in the problem domain. Experts tend to "chunk" or organize information into more highly structured patterns and to complete the task more quickly than novices. Experts also appear to represent problems differently than novices because of their superior ability to recognize patterns, infer relationships, disregard irrelevant information, and recall similar problems from past experience.

Differences between expert and novice instructional designers are likely to be similar. For example, novice designers are more likely than experts to use design models at a surface level. As a result, novices may miss important, though subtle, characteristics of performance and the developmental experiences which lead to satisfactory performance. Training programs based solely on the behaviors of established masters could ignore the similarities between components of the criterion task as well as optimum training procedures to teach trainees when to generalize and discriminate the application of learned competencies. With experience and reflection, expert instructional designers should represent design problems more accurately and in a more organized manner. Thus, expert instructional designers' schemata should be governed by qualitatively superior intellectual strategies which make possible a better understanding of problems and the process by which they may be solved. Cognitive scientists have named these higher order cognitive skills metacognition.

Metacognition

The cognitive functions of experts are likely to proceed in a hierarchical manner with superordinate processes controlling more basic subordinate processes. Kagan (1976) has identified some of these higher level metacognitive or executive functions, to include: (1) recognition of the nature of a problem and adjustment of effort to task difficulty; (2) flexibility with respect to alternatives; (3) activation of organization and rehearsal strategies; (4) control of distraction and anxiety; and (5) faith in the power of thought.

Just as there are different stages of cognitive development, similar levels can be identified with respect to models of instruction.

The role of executive functions was investigated in a study by Hayes-Roth (1980) in which the predictions made by the opportunistic model of planning (Hayes-Roth and Hayes-Roth, 1979) were tested for an errand-planning problem. The results of this study showed that people can acquire new executive strategies by instruction or experience, and adjust old or adopt new executive strategies according to the problem characteristics. There was also an indication that "knowledge of alternative executive strategies and the ability to apply them appropriately may be an important component of planning expertise" (p. 48). Though controversial (e.g., Gagne, 1987), the use of executive strategies may help to allocate cognitive resources and increase the generality of cognitive abilities. This suggests that training in the application of a variety of executive strategies may be an aid in teaching instructional design.

Implications

The apparent similarity between the characteristics of high-level modes of thought and those required for instructional design raises implications for training and future research. First, it appears that an instructional design model represents the design schema of its user. In part, this explains the proliferation of design models that has occurred (e.g., Andrews & Goodson, 1980). Each model is in effect, a schema of the specific actions required to solve the design problem it addresses. What is important, however, is that these models must match the users' cognitive abilities. Just as there are different stages of cognitive development, similar levels can be identified with respect to models of instructional design. Thus, these models tend to differ on levels of specificity, from elaborate and algorithmic to simple and generic. In practice, users must find or generate a model that matches their existing cognitive abilities.

An interesting paradox here is that, as design models become less specific, designers need more highly developed and sophisticated schemata to implement the subprocesses that are subsumed under the various stages of design. Like expert chess players, expert instructional designers must have a broad knowledge base or vocabulary of moves which is well developed and categorized in order to process information efficiently and organize it into large chunks. This intellectual structure allows designers to retrieve and use their schemata more effectively as they receive new information from the problem space.

Second, because instructional design involves systematic thinking, clearly stated purposes must follow an orderly structure to their conclusions. As a design project progresses and becomes more well defined, thinking shifts from broad considerations such as needs analysis, to more specific issues such as specification of individual lesson objectives. This is not to say that instructional design is a top-down process, however. One of the features of the planning model proposed by Hayes-Roth and Hayes-Roth (1979) is the flexibility to move from abstract to specific levels and back again. Decisions made in the initial stages of planning can affect subsequent stages, and the designer must be aware of these possibilities. This aspect of planning is controlled by the executive strategies which are elements of the designer's metacognitive structures.

Third, a broad knowledge base is needed. Beery, et al. (1981) have identified domains of knowledge which they consider essential for instructional designers. These include systematic planning, learning theory, instructional theory, and educational measurement. Basic skills in interpersonal communication, content analysis, task analysis, learner analysis, specification of objectives, and interpretation of data were also listed as important. Integration and restructuring of designers' schemata are necessary for the knowledge and skills to be used effectively, however. As Piaget's theory indicates, the process of cognitive development is continuous. As one develops expertise in a domain, skills are refined and knowledge broadened as designers gain experience working through successive projects. Again, a metacognitive awareness of designers' knowledge can assist in this development. If deficiencies are known, steps can be taken to acquire the needed information from sources outside of the designers' memory.

Finally, creativity and flexibility appear to be important, if not necessary, components of high-level thinking. Sachs (1981) has claimed that, "It is the developer's judgment, sensitivity, and inventiveness that leads to success" (p. 7). Instructional design models provide a proper framework for planning, management, and organization of information. The models often fail, however, to specify when a step is completed, or what to do to attack problems. Rote adherence to a model does not guarantee an optimum product. Many decisions are influenced by outside constraints and interactions between people and events. Therefore, the ability to be flexible and creative in response to difficulties should be of concern. For example, it appears that attitudes can be as important as the methods used. Anderson (1980) reported research suggesting that if people imagine themselves as creative, this can lead to an increase in the creative ideas they produce.

By examining the intellectual properties of instructional design strategies, designers can use models more efficiently and better understand the thinking required to design instruction.

Summary

We have characterized instructional design as an intellectual activity. As such, instructional design can be understood with reference to the way individuals think, develop design schemata, and control and use specific and general design knowledge.

By examining the intellectual properties of instructional design strategies, designers can use models more efficiently and better understand the thinking required to design instruction. In addition, examining "design" thinking makes possible analyses of the kinds and levels of activity and expertise required to use various design models and approaches. As designers become more experienced in applying knowledge and skills in a systematic way, the specific details of the design process become less important. And, as more is learned about how designers think, teaching others to design instruction can give greater emphasis to understanding design strategies. This will be made possible by revealing the purposes and structure of alternative design strategies as well as the intellectual rationale for why such strategies are necessary.

References

Alba, J. W., & Hasher, L. (1983). Is memory schematic? *Psychological Bulletin, 93*, 203-231.

Anderson, B. F. (1980). *The complete thinker.* Englewood Cliffs, NJ: Prentice-Hall, Inc.

Andrews, D. H., & Goodson, L. A. (1980). A comparative analysis of models of instructional design. *Journal of Instructional Development, 3*(4), 2-16.

Annarino, A. A. (1983). The teaching-learning process: A systematic instructional strategy. *Journal of Physical Education, Recreation and Dance, 54*(3), 51-53.

Bartlett, F. C. (1932). *Remembering: A study in experimental and social psychology.* Cambridge: Cambridge University Press.

Beery, M., Coleman, M., Durzo, J., Foshay, R., Fowler, B., Shrock, S., Schwen, T., Silber, K., Stevens, D., Terrell, B., Wileman, R., & Bratton, B. (1981). Competencies for the instructional/training development professional. *Journal of Instructional Development, 5*(1), 14-15.

Bower, G. H., Black, J. B. & Turner, T. J. (1979). Scripts in memory for text. *Cognitive Psychology, 11,* 177-220.

Carroll, J., Thomas, J., & Malhotra, A. (1980). Presentation and representation in design problem-solving. *British Journal of Psychology, 71,* 143-153.

Chase, W. G. & Simon, H. A. (1973). Perception in chess. *Cognitive Psychology, 4,* 55-81.

Chi, M. T. H., Glaser, R., & Rees, E. (1982). Expertise in problem solving. In R. J. Sternberg (Ed.), *Advances in the Psychology of Human Intelligence.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Egan, D. E., & Schwartz, B. J. (1979). Chunking in recall of symbolic drawings. *Memory and Cognition, 7,* 149-158.

Ericsson, K. A., & Simon, H. A. (1984). *Protocol analysis: Verbal reports as data.* Cambridge: The MIT Press.

Fikes, R. E. (1977). Knowledge representation in automatic planning systems. In A. K. Jones (Ed.), *Perspectives on computer science.* New York: Academic Press.

Fiske, S. T., Kinder, D. R., & Larter, W. M. (1983). The novice and the expert: Knowledge-based strategies in political cognition. *Journal of Experimental Social Psychology, 19,* 381-400.

Gagne, R. M., & Briggs, L. J. (1979). *Principles of Instructional Design* (2nd ed.). New York: Holt, Rinehart, and Winston.

Gagne, R. M. (1987, April). *International perspectives on meta-learning theories and instructional design.* Presentation as discussant at the meeting of the American Educational Research Association, Washington, D.C.

Greeno, J. G. (1976). Indefinite goals in well-structured problems. *Psychological Review, 83,* 479-491.

Hayes-Roth, B. (1980). *Flexibility in executive strategies.* (Rep. No. N-1170-ONR). Santa Monica, CA: The Rand Corp.

Hayes-Roth, B., and Hayes-Roth, F. (1979). A cognitive model of planning. *Cognitive Science, 3,* 275-310.

Holmes, G., & Sherman, T. M. (1987). *Using computers in the front-end analysis phase of instructional design.* Unpublished manuscript. Virginia Polytechnic Institute, Blacksburg, VA.

Jeffries, R., Turner, A. A., Polson, P. G., & Atwood, M. E. (1981). The process involved in designing software. In J. R. Anderson (Ed.), *Cognitive skills and their acquisition.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Kagan, J. (1976). *Cognitive development.* Washington, DC: American Psychological Association.

Landa, L. (1976). *Instructional regulation and control.* Englewood Cliffs, NJ: Educational Technology.

Larkin, J., McDermott, J., Simon, D. P., & Simon, H. A. (1980). Expert and novice differences in solving physics problems. *Science, 208,* 1335-1242.

Mayer, R. (1983). *Thinking, problem solving, and cognition.* New York: W. H. Freeman and Co.

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.

Nelson, W. A. (1987, April). *Procedural differences and knowledge organization in expert and novice instructional designers.* Paper presented at the meeting of the American Educational Research Association, Washington, D.C.

Newell, A. & Simon, H. A. (1972). *Human problem solving.* Englewood Cliffs, NJ: Prentice-Hall, Inc.

Piaget, J. (1970). *The science of education and the psychology of the child.* New York: Viking.

Pitt, R. B. (1983). Development of a general problem-solving schema in adolescence and early adulthood. *Journal of Experimental Psychology: General, 112*(4), 557-584.

Polya, G. (1957). *How to solve it* (2nd ed.). Princeton, NJ: Princeton University Press.

Reigeluth, C. M. (1983). Instructional design: What is it and why is it? In C. M. Reigeluth (Ed.), *Instructional design theories and models: An overview of their current status* (pp. 3-36). Hillsdale, NJ: Erlbaum.

Reitman, W. (1965). *Cognition and thought.* New York: Wiley.

Sacerdoti, E. D. (1974). Planning in a hierarchy of abstraction spaces. *Artificial Intelligence, 5,* 115-135.

Sachs, S. G. (1981). Practicing the art of instructional development. *NSPI Journal, 20*(7), 8-10.

Scandura, J. M. (1983). Instructional strategies based on the Structured Learning Theory. In C. M. Reigeluth (Ed.), *Instructional-design theories and models: An overview of their current status* (pp. 213-246). Hillsdale, NJ: Erlbaum.

Schank, R. C. & Abelson, R. P. (1977). *Scripts, plans, goals, and understanding.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Sherman, T. M. (1980). *Instructional decision-making.* Englewood Cliffs, NJ: Educational Technology.

Simon, H. A. (1973). The structure of ill-structured problems. *Artificial Intelligence, 4,* 181-201.

Wildman, T. M. (1980). Instructional design as a framework for unifying curriculum. *Educational Technology, 20*(3), 16-20.