Cognitive Approaches to Instructional Design

Brent Wilson, David Jonassen, and Peggy Cole

University of Colorado at Denver

Full reference:


Also available at:

http://www.cudenver.edu/~bwilson

Objectives for this chapter:

1. To introduce you to some innovative methods for doing instructional design (ID), such as rapid prototyping and automated design systems.

2. To survey some examples of training models based on cognitive learning principles, such as cognitive apprenticeships and minimalist training.

3. To offer a set of guidelines for designing cognitive-based training.

The field of instructional design (ID) has enjoyed considerable success over the last two decades but is now facing some of the pains expected along with its growth. Based largely on behavioristic premises, ID is adjusting to cognitive ways of viewing the learning process. Originally a primarily linear process, ID is embracing new methods and computer design tools that allow greater flexibility in the management and order of design activities. In the present climate of change, many practitioners and theorists are unsure about "what works"; for example, how to apply ID to the design of a hypertext system or an on-line performance support system. Our purposes are (1) to review new methods and tools for doing ID, (2) to survey some promising models of training design that incorporate cognitive learning principles, then (3) to offer some guidelines for the design of training programs based on those learning principles.

Bridging learning and performance

Training is typically viewed as something done apart from a job setting, perhaps in a classroom or lab area. You learn in training settings and you work in job settings. However, the line between "on the job"
and "training" is becoming blurred. Table 1 makes this point. Clearly, learning may be supported both on the job as well as in formal training settings. Conversely, aspects of the job environment can be successfully simulated in learning settings. A comprehensive treatment of training design would necessarily include on-the-job performance support in detail (e.g., Taylor, 1991). Many of the principles we discuss below relate to training in the job context, but for purposes of delimiting scope, we give most attention to training in controlled settings designed for learning. You may wish to consult Gery (1991) for greater discussion of job-based learning systems.

Everyday Work Settings

Traditional View. Work is done; people apply their knowledge by performing tasks and solving problems.

Revised View. Work is done with appropriate tools, info sources, job performance aids, help and advisement systems; workers continue to learn on the job through mentoring, apprenticeships, internships, etc.

Learning Settings

Traditional View. People learn in classrooms; there they acquire the knowledge and skills needed to perform successfully on the job.

Revised View. People learn by introducing elements of work setting-tools, aids, help systems-into manageable and low-risk training environments. Job demands are simulated in controlled training settings.

Table 1. Different views of work and school settings.

Training versus education

Conventional wisdom says that training is more context- and job-specific than education; training is skills-based whereas education is knowledge-based. Unfortunately, this distinction has been used as an excuse for rote procedural training for many years. We believe that the distinction between training and education is not as clear-cut as many believe; when learning takes place, both knowledge and skill are acquired. Indeed, most training situations call for some degree of meaningful understanding and problem-solving capability. Educational institutions, of course, also tend to neglect meaningful learning; medical schools, for example, suffer from a reputation of inculcating basic science into students' heads, then expecting them to successfully transfer that knowledge to clinical settings later. A cognitive view of instruction would argue that both training and educational systems need a better repertoire of effective strategies to make material more meaningful and useful to learners. Thus our discussion below should relate to both technical training and educational problems; in this chapter, we use the term instruction to denote both training and education.

Changes in ID

ID as a discipline rests on the twin foundations of (1) a systems design model for managing the instructional development process and (2) theories that specify what high-quality instruction should look like.
like (Reigeluth, 1983, 1987). These foundations have served designers over the last twenty years, as ID practitioners have proceeded from public schools and higher education into military, business, and industrial training areas. The consensus is that ID works, that the models deliver on their promise of effective instruction.

At the same time, the ID community continues to examine its premises. Methodological advances such as rapid prototyping have reshaped traditional thinking about systems-based project management (Tripp & Bichelmeyer, 1990; see discussion below). Sophisticated computer-based tools are helping to automate the ID process (Wilson & Jonassen, 1990/91). On a different front, critiques from cognitive psychology have called into question many of the recipe-like behavioral prescriptions found in traditional ID theories (Bednar, Cunningham, Duffy, & Perry, 1991; Jonassen, 1991; Wilson & Cole, in press a and b; Winn, 1990). As a result of these changes, ID is clearly moving toward greater flexibility and power in its recommended processes and its specifications for instructional products.

**New ID Methods and Technologies**

From its inception, ID practice has fallen short of its ideal prescriptions. Based on cybernetic principles of general systems theory, the ideal design process relies on constant systemic feedback. Such an instructional system acts something like a thermostat, always monitoring its own effectiveness, making revisions as needed to optimize learning outcomes. These cycles of self-testing and correction are repeated during the design process as well as during implementation and maintenance of the system. In this way, ID can adapt to differences in content, setting, learner characteristics, and other factors.

In practice, however, ID methods tend to proceed in linear fashion from defined needs and goals. Once the needs are identified and goals for the instruction defined, designers rarely look back. Instead, they tend to move through planning, design, and development phases in lock-step order. This has been necessary because of the enormous cost of cycling back to previously "completed" phases. Designers, sensitive in the first place to criticisms of cost, have been loathe to truly apply the iterative cycles of review and revision prescribed by systems theory. The application of theory is further weakened when tasks are compartmentalized and team members isolated from the rest of the system; in a large project, an individual may specialize in task analysis and never interact with designers at later stages. In short, the exigencies of the situation have made the application of theory impossible. For certain kinds of well-defined content within stable training environments, the linear approach may work satisfactorily. However, the limitations of linear ID become apparent when working in ill-defined content domains, or when working with highly diverse populations.

In response to this problem, a number of techniques and technologies have been developed to allow designers greater flexibility in design activities. Several of these are discussed below.

**Rapid prototyping**

ID shares much in common with computer science, particularly the sub-area called "systems design." The traditional wisdom of computer systems designers has been to design systems in linear fashion based on defined needs goals, almost parallel to our ID processes (Maher & Ingram, 1989). However, systems designers also face the same problems of cost and rigidity. Recently, systems designers have developed a method for developing large-scale systems (Whitten, Bentley, & Barlow, 1989). At very early stages of planning, a small-scale prototype is built that exhibits key features of the intended system. This prototype is explored and tested in an effort to get a better handle on the requirements of the larger system. The prototype is then scrapped as designers start over and build the larger-scale system. This process is called rapid prototyping. Its advantage is that it allows for tryout of key concepts at early stages when costs are small and changes more easily made.
Rapid prototyping applied to ID is a technology intended to allow greater flexibility in defining the goals and form of instruction at early stages (Tripp & Bichelmeyer, 1990). Prototypes may be shallow or narrow: shallow in the sense that the entire look of a product is replicated minus some functionality, or narrow in the sense that a small segment is completed with all functionality, leaving other whole portions of the final product undeveloped.

Prototyping can be relevant to all kinds of training development projects, but its value is most apparent in the design of computer-based systems. Imagine developing a sophisticated multimedia system strictly from identified needs and goals and without early working prototypes! Easy-to-use authoring programs such as HyperCard or ToolBook are commonly used as prototyping tools because of their power and flexibility. For one project we worked on, a prototype for a multimedia weather forecasting course was first developed using SuperCard, a close cousin of HyperCard that supports color. This prototype was repeatedly tested on users and revised for four months before serving as the starting point for the final, PC-based course (Wilson & Heckman, 1992).

Rapid prototyping may be done for a variety of reasons, including:

1. to test out a user interface;
2. to test the database structure and flow of information in a training system;
3. to test the effectiveness and appeal of a particular instructional strategy;
4. to develop a model case or practice exercise that can serve as a template for others;
5. to give clients and sponsors a more concrete model of the intended instructional product;
6. to get user feedback and reactions to two competing approaches.

It should be clear that rapid prototyping can help designers break out of the linear approach to design. Tripp and Bichelmeyer (1990) also argue that rapid prototyping is more in line with how people actually solve problems in other domains, which is far from a linear process.

Automated ID systems

ID can be greatly facilitated by computers, making the process more efficient and flexible. There are three basic ways computers can help to automate ID procedures:

1. Data management. Bunderson and colleagues (Bunderson, Gibbons, Olsen, & Kearsley, 1981) described ID as a loop that begins with analysis of expert performance and ends with learners demonstrating that same expertise. In between, designers produce reams of paperwork, generating a "lexical loop." Such a process is badly in need of a database that can organize and interrelate a project's needs, goals, objectives, tests, and instruction.

2. Task support. A wide variety of production tasks can be supported by computers, ranging from graphics production to word processing to communication among team members.

3. Decision support. Computers can assist in many design decisions by providing aids such as:

--ready access to information
A number of efforts are now underway to develop comprehensive systems that automate the ID process. Most of these are in prototype form, but we look forward to sophisticated tools and systems being made available to designers in the future (Wilson & Jonassen, 1990/91).

**Formative evaluation**

Recall that systems theory requires constant self-monitoring and adjustment of the system. This is similar to the scientific method, in that we formulate hypotheses (designs) and test them, thereby supporting or altering our expectations. Formative evaluation is the primary means of doing this self-testing; at various stages, designers may try out instructional materials to improve their effectiveness. Formative evaluation may be performed in phases, beginning with expert review, one-on-one or small group trials, and tryouts with the target audience under the conditions that the materials were designed to function in. In the aforementioned weather course, early versions of the product underwent expert review by designers, scientists, and prospective learners from nearby field offices. These cycles of review resulted in significant changes in the course's form and content.

New methods and approaches to formative evaluation (Fogg, 1990; Tessmer, in press) are based on cognitive assumptions of performance. Whereas a system's evaluation in the past tended to focus on learners' success in performing the criterion task, cognitive techniques seek to uncover thinking processes as they interact with the material. For instance, think-aloud protocols (Smith & Wedman, 1988) require learners to "think out loud" as they work through instruction. Their verbal reports become a source of data for making inferences about their actual thinking processes, which in turn provide evidence concerning the effectiveness of instruction. Learners might also be asked to elaborate on their verbal reports, particularly the reasons for decisions which result in errors.

Computer-based instruction allows designers to collect records of learners' every action. This is particularly useful for systems with extensive learner control; for example, an "audit trail" of a learner's path through a hypertext lesson can suggest to designers ways to revise the lesson to make desired options more attractive. Audit trails consist of a report of all of the learners responses while navigating through an information system, including the screens visited, the lengths of times spent interacting with parts of the program, choices made, information input into the system, and so on. Their primary purpose or function has been to provide evidence for formative evaluation of materials (Misanchuk & Schwier, 1991), such as the paths selected by learners, where and why errors where committed. Designers can evaluate the frequencies and proportions of nodes visited, the proportion of learners electing options, and so on. This information can provide valuable insights into the patterns of logic or strategies that learners use while working through instructional materials. A primary advantage of such data collection, when compared with other formative evaluation techniques, is that it is unobtrusive, accurate, and easy to obtain since it is automated (Rice & Borgman, 1983).

Another method, constructive interactions, observes how pairs of learners use instructional materials together, making collaborative decisions about how to proceed through materials. The transcribed dialog provides evidence about learners' hypotheses and reasoning while interacting with instructional materials (Miyake, 1986; O'Malley, Draper & Riley, 1985). These techniques, like the think-aloud protocols, have provided designers with a richer understanding of how learners are going about learning—not just whether they fail or succeed.
Training as a sub-system

Systems theory views instruction as a system with many parts, but also as a sub-system within layers of larger systems. To be successful, then, designers must carefully consider the context or situation in which instruction is being developed. Two methods intended to ensure a proper fit between instruction and its larger context are performance analysis and environment analysis.

Performance analysis is a specific, performance-based needs assessment technique that precedes any design or development activities by analyzing the performance problems of a work organization (Gilbert, 1978; Harless, 1979; Jonassen, 1989). It begins with the assumption that "if it ain't broke, don't fix it"; that is, if no problems have occurred, no solutions are needed. Assuming that a problem in the way employees are performing has been identified, performance analysis begins by investigating the cause of the performance problem. Problems can result from a lack of skills or knowledge, lack of motivation, or an inadequate environment (physical, administrative, or logistical) in which to perform. Solutions to each of these types of problems differ. The premise of performance analysis is simple: The solution needs to fit the problem. Motivational problems may be solved by changing the work incentives or by methods aimed at increasing workers' cooperation, feelings of ownership, confidence in their ability to perform, or valuation of the tasks. Environmental problems may be solved by re-engineering the environment. Formal training is recommended only if the problem results from a lack of knowledge or skills. Training is seen as a last-resort remedy because of its relatively high cost.

Assuming that there is a verified learning need, can the instruction be successfully implemented in the context for which it is intended? Environment analysis investigates the context of any instructional system, both where the instruction will occur and how the instructional materials will be used (Tessmer, 1990). Physical factors include the facilities and equipment available, the life span of the instruction, distribution of the sites, the management and coordination available and even the physical climate. Use factors include descriptions of how and why materials will be used, characteristics of the learners who will use them, characteristics of the administrators of the materials, and important logistical factors such as the production, storage and delivery services, and the resources available to support their use. Careful attention to these factors may seem tedious, yet any one of these factors can mitigate the effectiveness of the best designed instructional programs.

Layers of necessity

Nelson, Magliaro, & Sherman (1988) noted that instructional designers can be thought of as professional problem solvers, much like doctors, lawyers, teachers, or other skilled workers. Doing ID, they say, is a high-level thinking process. Drawing on Simon's (1973; Newell & Simon, 1972) research on problem solving, they suggest that ID models serve in large part as designers' mental schemas for doing ID. That is, rather than strictly following the formal procedure, designers tend to internalize aspects of models over time: "As designers become more experienced in applying knowledge and skills in a systematic way, the specific details of the design process become less important" (Nelson, et al., 1988, p. 34).

Following this view of designers as flexible problem solvers, Tessmer and Wedman (1990) have studied the way designers make use of ID models and have suggested that, depending on the scope and constraints of a project, designers should resort to core principles or procedures. As resources permit, they may add on "layers" of additional analysis. Their "layers of necessity" model is a meta-model to aid designers in adapting ID models to the wide variety of uses to which they are put. Goal analysis, for example, is considered fundamental for any design intervention, whereas a thorough needs assessment might be considered a second-layer activity, to be accomplished when conditions permit. The layers of necessity model calls attention to the fact that designers flexibly adapt their professional knowledge; the key to its value, we believe, is in its correspondence to the way expert designers actually make on-the-job decisions. More research into actual designer behavior is therefore needed.
Cognitive Models of Training Design

Sir Alfred North Whitehead (1929) coined the term "inert knowledge" for the kind of knowledge schools typically teach. Students often fail to use knowledge gained in one setting (schools) in another key setting (on the job). Thus their knowledge is inert and is of no use to them when they need it. The conditions and expectations differ enough to make the transfer difficult. This is called the transfer problem, referring to the difficulty we all have in applying our knowledge in different situations where it could be useful.

Sometimes we lack key strategies or information to make use of our school knowledge on the job. Instruction is often at fault, neglecting to teach "when and where" to use the new knowledge. There always seems to be something important that someone forgot to tell you. Franks, Bransford, Brailey, and Purden (1990) offer a striking example from one undergraduate statistics course where the text is full of instruction and exercises on a variety of statistical tests. Near the end of the book, a section entitled "Which test do I use?" totals a mere 13 sentences. Students receive absolutely no practice deciding which statistical procedure to apply to a given problem. It is no wonder that students exit the class unsure of how to apply their new knowledge to the real world!

The cognitive models described below are all attempts to teach skills and knowledge in ways that will facilitate the successful transfer to job settings. (For a review of other cognitive teaching models, see Wilson & Cole, in press a.)

Minimalist training

Instruction or training of any kind involves a paradox. Learners don't understand instruction unless they're somewhat acquainted with the thing they are learning about. However, they are not prepared for the real thing until they've had some preliminary training. In graphic terms, trainers often must decide: "Shall I simplify the problem and lead students by the nose, or shall I throw them headfirst into the whirlpool of new information and try and hold them up as they gasp for air?" There is a continuing tension in all training between simplification and control versus exploration and exposure to real-world complexity. In-house we call this the "spoon-feeding" problem: To simplify or not to simplify instruction—that is the question! Carroll (1990) expresses it in the context of learning to use computer applications: "To learn, [users] must interact meaningfully with the system, but to interact with the system, they must first learn" (p.77).

To address this paradox, Carroll (1987, 1990) has carried on a line of research for more than ten years on what he calls "minimalist training." The problem can be put: What is the most painless way to get users up to speed using an IBM hardware/software system? Carroll and colleagues have studied minimalist training in an impressive array of experimental studies, showing clear and consistent advantages over lengthy tutorials. Three key principles of minimalist training are:

--Allow learners to start immediately on meaningfully realistic tasks.

--Reduce the amount of reading and other passive activity.

--Help make errors and error recovery less traumatic and more pedagogically productive (Carroll, 1990, p. 7).

Additional principles include:

--Encourage learners to reason about what they are doing.
--Design reading material to be read in different orders.

--Provide strong linkages between the instructional system and the target job system.

--Use learners' prior knowledge to advantage.

--Exploit specifics of the problem-solving situation (see also Nickerson, 1991).

The minimalist training approach is interesting because it reminds us how easy it is to say too much when trying to give help. Speaking in generalities, working with abstractions, reading about a task can be as much an obstacle as a help to the acquisition of new skills. Staying on the point, leaving room to reflect, staying out of the way of learners' spontaneous learning strategies are some of the relevant lessons for training design.

Cognitive apprenticeships

Gott (1988a) noted an interesting shift in modern work conditions. Due to the increased complexity and automation of production systems, the need is growing for high levels of expertise in supervising and using automated work systems. Correspondingly, the need for entry levels of expertise is declining. Workers on the job are increasingly expected to be flexible problem solvers; human intervention is often most needed at points of breakdown or malfunction. At these points, the expert is called in. Experts, however narrow the domain, do more than apply canned job aids or troubleshooting algorithms. Rather, they draw on a considerable amount of internalized knowledge in order to solve problems flexibly (Gott, 1988b). The irony is that now, when demand for true expertise is high, there are fewer on-the-job training opportunities for entry-level workers. The low-level work tends to be automated. There is often little or no chance for beginning workers to acclimatize themselves to the job, they are very quickly expected to perform like seasoned professionals. True apprenticeship experiences are becoming relatively rare. Gott calls this dilemma the lost apprenticeship: more complex job requirements with less time on the job to learn.

Borrowing on the apprenticeship theme, Collins and colleagues (Collins, Brown, & Newman, 1989; Collins, 1991) have developed a model that seeks to take the best features from traditional apprenticeships and apply them to modern training conditions. Collins (1991) believes that technology can play a major role in accruing the benefits of traditional apprenticeships while reducing the disadvantages (e.g., economic exploitation, variability of mentors). The Collins-Brown model of cognitive apprenticeship contains several instructional principles, listed below (see also Wilson & Cole, in press a.)

1. **Content:** Teach tacit, heuristic knowledge as well as textbook knowledge. There is a lot that the business school or law school, or any other school, failed to tell you. This is called tacit knowledge, and often comes in the form of a heuristic or rule of thumb. Heuristic knowledge is often utilized by experts without their conscious awareness. It is so embedded in the specifics of the problem that experts take it for granted, yet the lack of this kind of knowledge is precisely what will trip up the novice. The main way to get this tacit knowledge is by experience, yet cognitive apprenticeships need to take extra pains to ensure that this covert type of knowledge is at least partially uncovered, demystified, and taught explicitly to novices.

2. **Situated learning:** Teach knowledge and skills in contexts that reflect the way the knowledge will be useful in real life. Based on a somewhat radical model of human cognition (Brown, Collins, & Duguid, 1989), cognitive apprenticeships ground knowledge in authentic contexts. This is because knowledge and context are inseparable, they say. We know things through experience in concrete situations, so rather than fight that groundedness, why not maximize its effect by providing rich, meaningful contexts.
within which learners can try out their new knowledge and skill? These rich learning environments will then ease the transition to real job conditions.

3. **Modeling and explaining**: Show how a process unfolds and tell reasons why it happens that way. Process modeling and explaining the relation between process and underlying principles is a key part of the cognitive apprenticeship model. Instructional technologies such as stop-action video and multimedia can facilitate this modeling and explaining process.

4. **Coaching and feedback**: Observe students as they try to complete tasks and provide hints and helps when needed. The personalized attention that a one-on-one instructor provides is important for learners to pinpoint problems in performance and make needed adjustments. Again, technology can help with this, but so can peers in cooperative learning groups or instructors in traditional group instruction settings. The key is the personalized attention to performance, coupled with appropriate hints, helps, and encouraging feedback (Rossett, 1991). One key to effective coaching is to not interfere too much thereby "preventing the development ... [of] cognitive skills that allow students to detect and use their own errors" (Burton & Brown, 1979, p. 15; Brown & Van Lehn, 1980).

5. **Scaffolding and fading**: Support learners by performing parts of the task they cannot perform. Gradually reduce the amount of scaffolding, shifting more and more of the control to the learner. The amount of scaffolding can be regulated by varying equipment, task, or environment (Burton, Brown, & Fischer, 1984). Burton, Brown, and Fischer (1984) offer additional guidelines, such as:

   -- intervening if the learner endangers himself or herself,

   -- being "aware that coaching is more important at the beginning of the acquisition phase than later on" (p. 150),

   -- deciding when to move on so that the learner doesn't develop wrong conceptions or habits that have to be unlearned at a later stage.

6. **Articulation and reflection**: Have students think about and give reasons for their actions, thus making their tacit knowledge more explicit. Students need opportunities to look back over their efforts and analyze their own performance. Talking about one's plans and activities as they solve problems can help learners develop more appropriate mental models of expert performance.

7. **Exploration**: Encourage students to try out different strategies and observe their effects. This gives learners practice using their existing knowledge and helps to tie it in with problems in need of solution. If students develop misconceptions, confront them with anomalies and counter-examples (Collins & Stevens, 1983).

8. **Sequence**: Proceed in an order from simple to complex, with increasing diversity. Increasing diversity means you explore the full domain of interest. Teach the underlying principle first, then fine-tune the application of that principle to specific performance contexts (see also Wilson & Cole, in press b).

Programs incorporating these principles have been shown to be successful in a variety of academic settings, particularly in basic skills instruction (Collins, Brown, & Newman, 1989; Glaser & Bassok, 1989). There is some ambivalence, however, about how to apply the model to training problems. The metaphor of apprenticeship suggests a continuing training experience in an authentic setting. Very often, designers have neither the time nor the resources to develop full-blown apprenticeship experiences for learners. Demand for content coverage is often high. Must all training be a cognitive apprenticeship? We believe the answer is no, but even short-term interventions in less-than-ideal settings can strive to incorporate individual cognitive apprenticeship principles.
Cases, Microworlds, and Simulations

We turn below to techniques that can help bridge the gap between job and school settings.

Case-based approaches. Drawing on the well-established use of the case method in business, law, and medical schools, Graf (1991) proposes a model using focused case materials to provide a bridge for students between ID theory and practice. Cases may be real or fictional. Use of information-rich case materials allows students to practice any of the tasks typically addressed in ID courses—from conducting needs assessment to formulating evaluation plans. Because the cases are "focused," they are more efficient and manageable than actual field experience. Through a case method, students are exposed to a wide range of contexts and can view cases from multiple perspectives.

Schank and Jona (1991) propose a different approach to the use of cases based on these observations: (1) Experts, such as doctors and lawyers, make decisions in new situations by comparing them to previous cases; (2) when experts teach, they often tell stories; and (3) "learning takes place on a need-to-know basis" (p. 17). The Schank model:

1. places "students in a situation that they find inherently interesting,"
2. gives them a task that is "complex enough that all the information is not immediately available,"
3. teaches each "student what he or she needs to know, or might consider while doing the task, at precisely the points in the task at which the student becomes interested in knowing this information" (p. 17).

We believe that case-based instruction will continue to be influential, particularly in professional education and advanced training settings.

Functional context training. Montague (1988) provides evidence for the effectiveness of "functional context training," a spiraling method which begins with familiar objects about which learners have intuitive knowledge and moves to progressively more complicated but still familiar objects. For example, an introductory course for electronics technicians starts with a flashlight and proceeds to a table lamp, a curling iron, an AC adaptor, and a soldering iron. Instruction is situated in realistic settings; it helps students develop appropriate mental models and procedures by integrating several domains of knowledge at once: problem solving, basic electricity/electronics knowledge, mental models of devices, language processing, and mathematics.

Microworlds. Burton, Brown, and Fischer (1984) use skiing instruction as a pretext for developing a model for designing skill-based "microworlds." A microworld is a controlled (often computer-based) learning environment where a student is able to try out new skills and knowledge. It is a practice-oriented simulation. Like the approaches described above:

1. instruction proceeds from simple to complex skills,
2. knowledge, skills, and attitudes are integrated through problem-solving activities, and
3. instruction is situated in rich and meaningful settings.

Complexity is controlled by manipulating the equipment (e.g., the length of the skis), the task (e.g., gliding down hill, making easy turns, making more difficult turns), and the environment (e.g., a gentle slope feeding into an uphill slope, a steep slope, moguls). Burton et al.'s microworld model is a...
precursor of the cognitive apprenticeship model; for example, it incorporates modeling, coaching, fading, reflection, exploration, and encouraging the learner to debug his or her knowledge.

**Simulations.** Computer simulations provide an opportunity for the learner to act on realistic scenarios without attendant dangers and inefficiency (time and money). Flight simulators have been used for years to train pilots; students can be exposed to a wide array of flight conditions (e.g., fog, cross winds, down drafts, engine flare out) without endangering themselves or others. Students in medical technology can practice "time-consuming" laboratory tests in seconds rather than hours or days. Simulations facilitate exploration and reflection and can incorporate on-line modeling, coaching, and explanations. Furthermore, the level of task difficulty can be adjusted.

**Cognitive flexibility theory**

Spiro and colleagues (Spiro, Coulson, Feltovich, & Anderson, 1988; Spiro, Feltovich, Jacobson, & Coulson, 1991; Spiro & Jehng, 1990; Spiro, Vispoel, Schmitz, Samarapungavan, & Boerger, 1987) studied the learning behavior of students in medical school and found an interesting phenomenon. Because the quantity of material was so large, students would look for anything to make it more memorable, some kind of hook or key idea. Unfortunately, students over-simplified, stubbornly relying on a simplistic analogy presented during instruction, even when the data were more complex. Spiro termed this tendency a "reductive bias," that is, the inclination to learn things as simpler or easier than they really are.

Training for professionals requires advanced levels of knowledge (Jonassen, in press), where learners must solve complex, context-dependent problems. Cases tend to be unique, unpredictable, and ill-structured. In order to prepare learners to acquire needed skills, learners need instructional conditions that stress the interconnectedness of knowledge and different perspectives on the information, since it should be obvious that bosses nor clients rarely agree completely. So learners need to develop flexible knowledge.

Cognitive flexibility theory is an "an integrated theory of learning, mental representation, and instruction" (Spiro, Feltovich, Jacobson, & Coulson, 1991, p. 28). Cognitive flexibility theory provides a number of heuristics for designing instruction that avoids over-simplifying instruction by providing real-world cases, providing multiple representations of the content in order to enhance transfer, and requiring knowledge construction by learners, not knowledge regurgitation. A primary teaching strategy is the use of hypertexts that allow students considerable control as they explore and browse through a content domain.

**Cognitive Principles of Training Design**

**The help metaphor**

Training can be approached from many perspectives. Performance technologists (e.g., Stolovich, in press) like to think of training as a strategic investment contributing to the corporate bottom line. Another perspective is to think of training as a kind of service to people. Trainers offer a service to learners intended to improve their functioning in an environment. Thus training can be thought of as a kind of help, similar to help systems in a computer environment or aid provided by a social services agency (Inouye, 1992).

The quality of training, like help, can be judged on its effectiveness and its efficiency. To understand this perspective, think of a software company such as WordPerfect Corporation. Let's say WordPerfect offers a support system, where users having difficulties with software can phone and get help from a specialist. WordPerfect managers can evaluate the quality of help offered in a number of ways:
a. **Effectiveness**—solving the customer's problem, also known as the *power* of the help according to Inouye (1992). Does the help work?

- **Availability** of help. Is there someone around who can answer the caller's question?

- **Relevance** to the problem at hand. Is the information provided pertinent to the customer's immediate problem?

- **Understandability**. Is the help clear to the user? Are the instructions executable? Is the user able to take corrective action based on the help?

b. **Efficiency**—the timeliness and affordability of providing the help.

- **Mean time to help**. How long does the customer have to wait after phoning in for assistance until their problem is satisfactorily solved? Is the help message brief and to the point?

- **Cost**. What are the resources needed to provide the help? What are the costs to WordPerfect? To the customer?

Each time training is offered, there is by definition some problem to be solved, some goal to be reached. Even though not all training problems are as neatly quantifiable as on-line assistance systems, there are clear implications for training design which reinforce the lessons learned from the cognitive training models above. Training is most effective when it:

-- Responds to an immediate performance need. Training should seek to create "teaching moments" wherein the learner is trying to solve a problem, clearly needs assistance, and is highly receptive to assistance that will help him/her perform better.

-- Seeks to meet those teaching moments with relevant, clear instructional messages and practice opportunities.

-- Doesn't give too much or too little help. Excessive training wastes money, and it interferes with the learners' developing cognitive skills in detecting and learning from their own errors (Burton & Brown, 1979).

-- Doesn't get in the learners' way. People spontaneously apply a set of cognitive strategies to any situation or problem. Training should work with those strategies rather than compete with them.

The additional guidelines offered below are based on our review of cognitive training models and on the literature in cognitive learning and teaching methods. The guidelines are not mutually exclusive; for example, only one guideline specifically addresses motivation, yet every guideline affects motivation. We believe that designers can make use of these or similar guidelines as they seek creative solutions to problems in the design of all kinds of training and instruction.

**Foster a learning culture.**

1. Offer training, within an overall culture that encourages cooperation, risk-taking, and growth.

2. Get learners' buy-in and commitment in achieving training goals.

**Motivate learners.**
3. Demonstrate the value of the training to the learners and cultivate their sense of confidence in their ability to master the objectives.

**Make training problem-centered.**

4. Draw on authentic needs and contexts; make requirements of learning tasks similar to important requirements of job tasks.

5. Encourage learners' active construction of meaning, drawing on their existing knowledge (Resnick, 1983).

6. Teach multiple learning outcomes together (Gagne & Merrill, 1990).

7. Sequence instruction so that learners can immediately benefit from what they learn by applying it to real-world tasks.

**Help learners assume control of their learning.**

8. Provide coaching.

9. Provide scaffolding and support in performing complex tasks.
   a. Adjust tools (equipment), task, and environment.
   b. Provide timely access to information and expertise.
   c. Provide timely access to performance feedback.
   d. Utilize group problem-solving methods.
   e. Provide help only when the learner is at an impasse and only enough help for the learner to complete the task.

10. Fade support.

11. Minimize mean time to help (i.e., provide "just-in-time" training).

12. Encourage learners to reflect on their actions.


14. Encourage learners to detect and learn from their errors.

**Provide meaningful "practice."**

15. Provide opportunities for learners to apply what they've learned in authentic contexts. If it is not feasible to practice on real tasks, provide cases or simulations.


**Author Notes**

http://carbon.cudenver.edu/~bwilson/training.html
Minor portions of this chapter have been adapted from Wilson & Cole (in press a) and Wilson & Cole (in press b). We wish to thank Peter Beckschi for his editorial comments and support of this chapter.

References


presented at the annual meeting of the Association for Educational Communications and Technology, Orlando, FL, February 13-17, 1991.


### Glossary

**Case method.** The presentation of real or fictional situations or problems to learners to analyze, to discuss, and to recommend actions to be taken.

**Coaching.** A technique of cognitive apprenticeship whereby the instructor observes students as they try to complete tasks and provides hints, help and feedback as needed.

**Cognitive apprenticeship.** An instructional model that seeks to emulate the opportunities for extended practice on authentic tasks that apprentices have while working under a master craftsman.

**Cognitive flexibility theory.** A theory of learning for advanced knowledge. Advanced knowledge is seen as less rule-based and rigid than introductory knowledge. The theory recommends approaching content from multiple perspectives through multiple analogies and the use of hypertext instruction.

**Cognitive psychology.** "The scientific analysis of human mental processes and memory structures in order to understand human behavior" (Mayer, 1990, p. 1).

**Education.** Instruction which emphasizes far-transfer learning objectives; traditionally knowledge-based instruction which is not tied to a specific job, as opposed to training.

**Effectiveness.** A measure of whether a procedure or action achieves its purpose.

**Efficiency.** A measure of the timeliness and affordability of an action.

**Environment analysis.** The context of any instructional system, both where the instruction will occur and how the instructional materials will be used.

**Fading.** A technique of cognitive apprenticeship whereby the instructor gradually withdraws support and transfers full control of a performance to the learner.

**Formative evaluation.** On-going evaluation of instruction with the purpose of improvement.
Functional context training. A model of instruction that works from simple, familiar tasks and proceeds to more complex tasks with ample opportunities for practice.

Heuristic. A rule of thumb or guideline (as opposed to an invariant procedure). Heuristics may not always achieve the desired outcome, but are extremely valuable to problem-solving processes.

Hypertext. Non-linear text. Image a computer screen with a word in bold. You click on the word and it "zooms in" to greater detail. Hypertext allows you to zoom in and zoom out of subjects and make connections between topics. Hypertext programs are useful for instruction and for information access.

Inert knowledge. Knowledge a learner has acquired but fails to activate in appropriate situations.

Instructional design. The activity of planning and designing for instruction. Also, a discipline associated with the activity.

Layers of necessity model. A model of instructional design and development which prioritizes the needs of a project into layers; "each layer being a self-contained model." Additional layers are developed as time and resources allow (Tessmer & Wedman, 1990, p. 79).

Microworld. A computer-based simulation with opportunities for manipulation of content and practice of skills.

Minimalist training. An instructional approach which seeks to provide the minimum amount of instruction needed to help the learner master a skill. It emphasizes active learning and meaningful learning tasks.

Performance analysis. A specific, performance-based needs assessment technique that precedes any design or development activities by analyzing the performance problems of a work organization.

Performance support systems. Computer program that aids the user in doing a task. Examples include help systems, job aids, and expert system advisors.

Problem solving. The creative application of "various rules, procedures, techniques, or principles to solve complex problems where there is no single correct. . . answer" (Tessmer, Jonassen, & Caverly, p. 4).

Rapid prototyping. In a design process, early development of a small-scale prototype used to test out certain key features of the design. Most useful for large-scale or projects.

Scaffolding. A technique of cognitive apprenticeship whereby the instructor performs parts of a task that the learner is not yet able to perform.

Simulation. "A simulation is a simulated real life scenario displayed on the computer, which the student has to act upon" (Tessmer, Jonassen, & Caverly, 1989, p. 89).

Spoon-feeding problem. The dilemma in training between (1) how much to simplify and control the learning situation and (2) how much to provide for exploration and exposure to real-world complexity.

Training. Instruction which emphasizes job-specific, near-transfer learning objectives; traditionally skills-based instruction, as opposed to education.