Title:

Research-based guidelines for computer-based instruction development. By: Overbaugh, Richard C., Journal of Research on Computing in Education, 08886504, Fall94, Vol. 27, Issue 1

Database:

Academic Search Premier

HTML Full Text

RESEARCH-BASED GUIDELINES FOR COMPUTER-BASED INSTRUCTION DEVELOPMENT

Contents

Abstract

The current generation of authoring systems provides tools for nonprogrammers to create software. This article provides guidelines for educators who wish to design custom software to meet either their own or other's needs. Many instructional design theories and related computer-based empirical research findings are merged into a prescription for the most important facet of courseware development--instructional design. The instructional design process, based on Gagne's nine events of learning, is broken into four domains: (a) instructional set, (b) teaching strategies, (c) student performance, and (d) issues unique to computer-based instruction. (Keywords: computer-assisted instruction, courseware, instructional design, software design.)

The growing popularity of easy-to-use authoring systems may encourage many educators to develop courseware either for their own use or distribution. However, these developers should draw on the experience of instructional design theorists and researchers involved in the development and evaluation of educational courseware. Roblyer (1988) defined courseware development as a three-stage process: (a) instructional design, (b) preprogramming development, and (c) programming. The central tenet of courseware development is to fully develop the instructional method and the lessons(s) before the actual programming takes place.

This article provides guidance for educators interested in developing educational software based on their content expertise and their target audience, usually their own students. It is not an inclusive prescription for all instances of instructional design. It assumes that the considerations or general course specifications that delineate the need for courseware development have already been examined; therefore, these issues will be reviewed only briefly.

This article focuses on a three-stage process by which a predetermined instructional need will lead to pedagogically sound computer-based courseware. These stages include (a) lesson design, or defining the instructional set and teaching strategies; (b) student performance, or eliciting and assessing performance and feedback principles; and (c) computer-specific design issues, or developing learner control guidelines and some miscellaneous teaching tools. Completing these three steps clearly defines the predictable aspects of the learning task, leaving only the preprogramming development and actual programming to be carried out. Most components in the instructional design stage apply to all instructional design tasks, regardless of the delivery medium. Computer technology may be an efficient and versatile tool for teaching, but all computer-based instruction must be governed by the same theories and research applicable to all teaching tasks. The following discussion provides a brief overview of predevelopment design considerations, or general course specifications, and then offers a development prescription that incorporates contemporary learning theory and instructional design theory, including special considerations for computer-based learning materials.

GENERAL COURSE SPECIFICATIONS

The instructional design process begins with identification of an instructional need, formulation of broad course objectives, and examination of the conditions chat may affect knowledge acquisition (Gagne & Briggs, 1979; Landa, 1983; Roblyer, 1988). The determination of a suitable sequence might involve the following:

1. Elaboration. Teaching a concept using a simple-to-complex pattern (Collins & Stevens, 1983; Gropper, 1983; Reigeluth & Stein, 1983; Scandura, 1983), concrete-to-abstract pattern (Collins & Stevens, 1983; Gropper, 1983) or a meaningful-to-less-meaningful pattern (Aronson & Briggs, 1983).
2. Inquiry Learning. A learning environment in which students begin with hypotheses (Collins & Stevens, 1983) and are guided through a process to prove, modify, or disprove the hypotheses.
3. Discovery Learning. Students interact with their learning environment to discover, or rediscover, principIes and concepts. Discovery learning, often considered the best method for learning, is nearly impossible to administer because of time constraints, limitations on materials, and teacher/pupil ratios (Landa, 1983).

The next step--task analysis--is the process through which the instructional content and the conditions affecting learning acquisition are divided into the smallest constituent parts (Aronson & Briggs, 1983; Carrier & Jonassen, 1988; Landa, 1983; Roblyer, 1988). It includes five interrelated domains.

1. Learning Hierarchy. An analysis that breaks final learning outcomes into their smallest prerequisite parts in one of two ways: (a) a list or group of the prerequisite parts that does not reflect the order in which they should be taught (Aronson & Briggs, 1983; Gropper, 1983; Hannum, 1988) or (b) a sequential list that reflects the order in which the parts should be taught in order to lead to the final outcomes (Gropper, 1983; Hannum, 1988). This component determines the skills and concepts to be taught as well as the target audiences' starting state (Aronson & Briggs, 1983; Gropper, 1983).
2. Student Model. A determination of the nature and characteristics of the students to be transformed (Landa, 1983), including previous knowledge (Roblyer, 1988) and motivational factors (Keller & Suzuki, 1988; Montague, 1988).
3. Instructional Environment. The delineation of factors influencing the designer's choice of teaching strategies or selection of relevant prerequisites. This includes (a) tools, or the equipment needed to effectively deliver instruction, and (b) problems, or factors that may hinder the learning process, for example, teacher/student ratios, lack of equipment, or time constraints.
4. Learner Starting Point. The determination of the point at which instruction should begin, based on the learning hierarchy, student model, and instructional environment. In computer-based instruction the learners' positions in the learning hierarchy cannot be known and CAI must, therefore, be capable of providing instruction that begins at a variety of points, Fortunately, this capability is one of the primary advantages of computer-based instruction. Courseware can be designed to meet the needs of a varied audience by offering various entry levels; the techniques range from simple student/teacher-chosen levels in an opening menu, to computer-chosen levels based on a pretest, to user levels determined by the computer and based on learner responses (Hannafin & Peck, 1988).
5. Detailed Objectives and Subject Matter/Level of Cognitive Engagement. The conversion of the learning hierarchy into detailed objectives, including subject matter type and the level of cognitive engagement required to process the new information (Gagne & Briggs, 1979; Roblyer, 1988). Subject matter type refers to the abstractness of the objective, ranging from facts to principles governing abstract concepts. The level of cognitive engagement refers to the type of processing needed to assimilate the new information, ranging from rote memorization to formulation (see Aronson & Briggs, 1983; Collins & Stevens, 1983; Gagne & Briggs, 1979; Merrill, 1983, 1988; Wager & Gagne, 1988).

COMPUTER-BASED LESSON DESIGN

Gagne's nine events of learning serve as a framework for successful courseware development (Reeves, 1986). The nine events include (a) gaining attention, (b) informing the learner of the lesson objective(s), (c) stimulating recall of prior learning, (d) presenting stimuli with distinctive features, (e) providing learning guidance, (f) eliciting performance, (g) providing feedback, (h) assessing performance, and (i) enhancing retention and learning transfer (Wager & Gagne, 1988). Although the nine events normally apply to teaching individual concepts, they also provide a logical framework for discussing general instructional design techniques and teaching strategies for computer-basexl instruction. The events can be grouped into three domains: instructional set, teaching strategies, and student performance, to which a fourth domain, issues unique to computer-based instruction, can be added. Because the present discussion focuses on general strategies for all aspects of instructional design, one sequential change will be made: the ninth event--enhancing retention and learning transfer--will be included in the second domain--teaching strategies--because the techniques for enhancing retention and generalization can be considered teaching strategies.

DOMAIN 1: INSTRUCTIONAL SET

Instructional set in well-designed computer-based instruction prepares learners to engage in the forthcoming new information. This domain includes Gagne's (1975) first three events of learning: (a) gaining the learners' attention; (b) informing learners of the objectives, here called "orienting activities"; and (c) stimulating recall of prior learning.

Gaining Attention

Gaining the learner's attention makes the student cognizant of the lesson's purpose. This may require something as simple as a title screen or as complex as a situational description or visual stimuli representing the courseware content (Wager & Gagne, 1988). Students already aware of the purpose of the lesson and seeking instruction do not need any stimulus, whereas other students may need to learn something about the upcoming lesson before recognizing the need to continue. Additionally, presenting information about the forthcoming lesson in a manner that causes students to realize the importance of learning the content makes the learning task relevant, which, in turn, raises motivation (Keller, 1983; Keller & Suzuki, 1988). To help learners use the program easily, courseware should also present explicit directions for using the program (Jonassen & Hannum, 1987).

Orienting Activities

Gagne's (1975) event called "informing learners of objectives" is here called "orienting activities" in order to include various techniques that should be included at the beginning of the lesson to relate upcoming learning to the student's present knowledge and mental state (Jonassen & Hannum, 1987; Wager & Gagne, 1988). Possible orienting activities include pre-questions, informing learners of expectations, and advance organizers.

Motivation. Presenting an overview of the lesson, including expectations for success, has two functions. First, knowing what to expect reduces user anxiety about the program which creates a relaxed mental state in which learning can more easily take place (Hannafin, Garhart, Rieber, & Phillips, 1985). Second, including expectations for success allows learners to judge whether the goals are obtainable and worth working toward. If learners perceive themselves capable of successfully completing or even working toward the prescribed skill, they will be motivated to do so (Keller, 1983; Keller & Suzuki, 1988). However, orienting activities for motivational purposes need not be included if some external event motivates students (Gagne, 1986).

Cognitive Enhancement Devices. Oftenting activities can also enhance learning by prestructuring the learners' mental set for the upcoming cognitive activities. It has been shown that nonspecific but relevant pre-questions before learning activities help facilitate new higher order structures, far transfer, and generalization (Anderson, Spiro, & Anderson, 1978; Hamilton, 1985).

Informing the learners of expectations (general, specific, or both) activate related items in long-term memory. This helps activate the learners' "mental sets" (Gagne, 1975, p. 94), which in turn enables them to assimilate the new information in a meaningful manner (Gagne & Briggs, 1979; Hannafin & Hooper, 1988).

Presenting the lesson objectives before students begin working with the content is another way to orient learners. Students who have been presented with objectives at the beginning of a lesson may be more inclined to review major points because self-review, a metacognitive skill, has been stimulated (Ho, Savenye & Haas, 1986).

In some instances, oftenting activities are not desirable (Gay, Trumbull & Smith, 1988) or are superfluous (Phillips, Hannafin & Tripp, 1988). Mayer (1984) has argued that orienting activities (for example, when lessons are already well sequenced), do not harm the learning process. Therefore, courseware directed at a varied audience should include orienting activities but should also include an option to bypass them when learners do not need them.

Stimulating Recall of Prior Learning and Supplying Missing Prerequisites

Stimulating students' recall of previous learning often proves necessary before a new concept can be learned (Aronson & Briggs, 1983; Reigeluth & Stein, 1983; Wager & Gagne, 1988). These prerequisite skills encompass knowledge or skills that students should already know. They serve as the beginning of instruction.

Stimulating recall of prior learning has been expanded here to include the process of supplying missing prerequisites. One of most common problems with classroom instruction is the failure to accurately pinpoint the correct starting state of learners. Students rarely enter a learning situation with the same amount and type of knowledge. This problem can be compounded when developers of computer-based courseware try to create effective instruction for audiences whose needs can only be estimated. However, this is a problem that computers are uniquely suited to solve. Computer courseware can be constructed to meet the needs of the intended audience by containing a "usable learning hierarchy" that addresses the needs of students with differing prerequisite skill levels. Usable learning hierarchy refers to the option of entering the learning task at the level suited to each user, based on some sort of starting-state assessment. This assessment might be an activity level specified by an instructor or student, or, in more advanced software, based on an online pretest or student-supplied answers to embedded questions. Furthermore, during the learning task, if the student, teacher, or program determines the absence of some prerequisite skill, the computer can easily teach the needed skill. For example, if a student using a chemistry simulation repeatedly makes the same mistake regarding a certain chemical, the program, in response to the student's answers or request, can branch to a tutorial about that particular chemical. The student can then return to the main program when the problem has been resolved.

DOMAIN 2: TEACHING STRATEGIES

The instructional designer has now reached the point at which learning actually begins. The teaching strategies domain consists of three events: presenting stimuli with distinctive features, providing learning guidance, and enhancing retention and transfer. These three events encompass the juncture at which the learner receives information specified by individual objectives and makes a mental model of the information. This is sometimes called the cognitive stage (Salisbury, 1988).

Presenting Stimuli With Distinctive Features

Presenting stimuli with distinctive features has two components: pedagogical (the characteristics of the material initially presented to the learner) and technical (the visual appearance of the initial concept).

Pedagogical Component. The initial objective, which has many names (e.g., exemplar, epitome, and paradigm), is that the new material should be designed so that learners can understand it and so that it maintains their interest. The most commonly recommended guideline has been to keep it simple (Aronson & Briggs, 1983; Collins & Stevens, 1983; Gropper, 1983; Reigeluth & Stein, 1983; Scandura, 1983). Other recommendations have included the suggestion that the initial presentation be meaningful or relevant (Aronson & Briggs, 1983; Keller, 1983; Keller & Suzuki, 1988) and concrete (Collins & Stevens, 1983; Gropper, 1983). If the lesson attempts to teach attitudes, a positive role model must be chosen for the initial presentation (Keller, 1983; Keller & Suzuki, 1988).

Technical Component. Presenting stimuli with distinctive features also involves the manner in which information is presented on the computer screen. The appearance of text and graphics, as well as the manner in which student attention focuses on particularly important points, should be considered. The primmed recommendation is to keep the screen as simple and uncluttered as possible because presenting too much information at one time can be confusing and overwhelming. The use of color and text formatting (e.g., inverse print, underlining, shading, arrows, and flashing) can be effective but also should be carefully considered. For example, flashing is distracting and, because it makes text difficult to read, should never be used for text presentation (Jonassen & Hannum, 1987).

Color, which is most effective when used consistently for cuing and highlighting, directs student attention to important points or relationship (Merrill, 1983; Merrill & Bunderson, 1979). Color should be used sparingly because the more color that is used, the less effective it becomes (Kanner, 1968). Studies have revealed that color cuing helps low achievers learn more but can hinder high achievers' learning (Hativa & Teper, 1988).

Providing Learning Guidance

Now that the initial concept has been presented, the courseware designer must determine how to present the learner with subsequent information so that the program will direct the learning process. If the learning task has not yet been prioritized according to an anticipated learning sequence (Aronson & Briggs, 1983; Wager & Gagne, 1988), this must be done before instruction begins. Of the three basic instructional strategies--elaboration, inquiry learning, and discovery learning--elaboration strategies require the student to do the least cognitive processing, and discovery learning strategies require the most. Because deeper levels of processing cause new information to be better encoded in the learners' memory (Jonassen, 1988), effective instruction should combine facets of all three strategies. However, instruction often includes only elaboration and inquiry techniques because of instructional environment constraints encountered with discovery learning.

Elaboration. Instructional sequences that teach concepts by initially presenting basic, understandable ideas and moving progressively toward the more complex aspects can be called "elaboration sequences." Although all elaboration sequences follow a basic-to-complex progression, there are four common types. The most universal is the general, nonspecific, simple-to-complex sequence (Collins & Stevens, 1983; Gropper, 1983; Reigeluth & Stein, 1983; Scandura, 1983). More specific designations are the concrete-to-abstract design (Collins & Stevens, 1983; Gropper, 1983; Jonassen & Hannum, 1987), meaningful-to-less-meaningful progression (Aronson & Briggs, 1983; Keller & Suzuki, 1988; Salisbury, 1988) and important-to-less-important design (Collins & Stevens, 1983). The selection of one of these elaboration sequences should be based on subject matter and should be congruent with the type of presentation chosen in the previous step.

Inquiry Learning. Because posing questions and guiding students into question generation promotes more learning and increases curiosity (Jonassen & Hannum, 1987; Keller, 1983), effective instruction combines techniques from elaboration and inquiry. In inquiry learning, one poses a rule or question and then guides the student to discover evidence to support or disprove the rule or discover an answer to the question (as opposed to discovery learning, in which no rule is stated initially) (Collins & Stevens, 1983). As is the case when using elaboration strategies, a decision must be made whether to teach students particular rules and principles or teach them how to derive rules and principles. Collins and Stevens (1983) suggested that several strategies were particularly suited for computer-based courseware, including forming and evaluating hypotheses and tracing contradictions to a conclusion.

Discovery Learning. Discovery learning arguably provides the most complete learning experience by guiding students through the complex process of building a sufficient knowledge base upon which a hypothesis can be formed, proved or disproved, and, if necessary, revised. Although discovery learning is rarely used because of time and management requirements (Landa, 1983), a teacher should at least provide a discovery experience that teaches students how to discover algorithms and then presents situations in which to apply the rules.

Method-Independent Techniques for Effective Instruction. Five effective instructional strategies--(a) counterexamples, (b) analogies, (c) contextual elaboration, (d) representational elaboration, and (e) knowledge structure enhancers--can easily be utilized in computer-based instruction. These techniques have been categorized as "method independent" because they are appropriate to all methods of instruction.

Counterexamples. One of the most frequently recommended strategies for illustrating a concept when presenting new material is the use of counterexamples (Aronson & Briggs, 1983; Collins & Stevens, 1983; Jonassen & Harmum, 1987; Merrill, 1983). When students consider counterexamples, negative examples, or alternative examples, they gain more diagnostic skills and learn to support their assertions. Therefore, both positive and negative examples should be used (Collins & Stevens, 1983).

When using counterexamples, care should be exercised to ensure similarity among all nonrelevant characteristics of the target concept and the counterexamples (Merrill, 1983). However, when the task involves discriminating between two nearly identical items, it may be necessary to exaggerate the difference. For example, when teaching students to distinguish between in-tune and out-of-tune pitches, it might be helpful to make the out-of-tune pitch very out of tune and work toward finer discriminations. This can also be considered an overall teaching strategy, similar to Gropper's (1983) "shaping." If the learning task involves motor skills, a counterexample, or incorrect method, can be performed to help students understand the correct way (Gropper, 1983). An example of this technique would be to oversheet and undersheet a jib-sail to illustrate the effects of incorrect settings on boat speed. Correct sail settings should then be easier to understand.

Analogies. Another common technique for teaching is the use of analogies and metaphors (Jonassen & Hannum, 1987; Merrill, 1983) generated either by the computer or the learner (Jonassen, 1988). Analogies help develop new cognitive structures (Hannum, 1988) by making the strange familiar and the familiar strange (Keller, 1983). The use of analogies can be thought of as an elaboration technique, but keep in mind that analogies are not essential prerequisites for the concept(s) being taught. If more than one analogy is available at a given time for the subject matter at hand, as many as are needed should be included, particularly if there are considerable individual differences among the learners (Reigeluth & Stein, 1983).

Contextual elaboration. Contextual elaboration refers to the context of the material to be learned, the historical background, or some prerequisite information that has not been previously discussed. This elaboration increases the knowledge and understanding of particular subject matter or raises the interest level (Merrill, 1983) through helpful but nonessential information (Aronson & Briggs, 1983; Gropper, 1983). These elaborations might involve information directly related to the current learning task or anecdotes and personal stories to increase curiosity and interest that will transfer to the present task (Keller, 1983).

Representational elaboration. People who do not understand information delivered by text quite often will understand it if it is presented or augmented by various visuals (Merrill, 1983). Difficult topics also sometimes become easier to understand when augmented by graphic displays (Gropper, 1983). Computer courseware can easily present information in either text mode or graphics mode but when possible, both should be used. The learner could have a choice between visual and nonvisual presentation modes, or simply have the option to view additional displays if the other mode is desired.

Knowledge-structure enhancers. Devices that facilitate an understanding of relationships among the various concepts being learned often prove helpful. Devices can be as simple as something that asks the learner to create a mental image of the concept just learned (Jonassen, 1988) or that provides a screen-generated image (Jonassen & Hannum, 1987). Asking the learner to think of an analogy or paraphrase a concept also enhances knowledge assimilation (Merrill, 1983). Another common way to represent knowledge is through cognitive or semantic mapping (Denenberg, 1988; Hannum, 1988). As learners progress through the learning task, the relationships among concepts can be depicted pictorially by either the computer or the student to help illustrate the overall structure and guide the learning sequence.

Enhancing Retention and Learning Transfer

Enhancing retention and learning transfer, which comes after eliciting and assessing performance and providing feedback, is the last step in Gagne's (1975) hierarchy. However, including retention and transfer in the teaching strategies section makes sense because similar techniques and overlapping procedures are used when discussing teaching strategies in terms of overall instructional strategies as opposed to the progression to be followed when delivering a single concept.

Simply gaining a bit of knowledge is not learning. Instead, learning requires relevant practice (Gagne, 1986); that is, a student must be able to remember and apply knowledge to the same or similar situations in the future, a skill generally accomplished through practice and elaboration (Wager & Gagne, 1988). According to Salisbury (1988), two stages--the associative and the autonomous--occur after the cognitive stage, the point at which initial learning takes place. During the associative stage, practice in strengthening concepts and relationships is provided and errors contained in initial concepts are corrected. During the autonomous stage, skills become automatic. Practice results in both assimilation and autonomy.

Varied Practice. Varied practice is essential for promoting generalization and transfer of knowledge (Gropper, 1983), particularly for the lower performance types, such as rote memory tasks and skill use. Higher level concepts are easier to remember because of the deeper processing required (Merrill, 1983). For the lower level skills, learners must have practice distinguishing between examples and nonexamples. For higher level skills, they must have opportunities to practice creating new solutions (Aronson & Briggs, 1983). If possible, practice should emphasize the practicality of the newly learned concepts or skills in a realistic, situational environment (Montague, 1988; Salisbury, 1988). Gropper (1983) also cautioned that there should not be too much or too little time between practice sessions. If students have no chance to practice during a four-hour presentation, retention of the information becomes more difficult.

Review. Review enhances retention, particularly when done prior to testing (Jonassen & Hannum, 1987). An effective review strategy involves a three-part summary, including (a) a precise statement of each idea and fact that has been taught; (b) reference examples, such as typical, easy-to-remember examples; and (c) diagnostic self-test items for each idea (Reigeluth & Stein, 1983). There are two types of review: internal, which includes material contained only in the present lesson, and cumulative, which includes all material up to and including the current lesson (Reigeluth & Stein). Review leads to increased retention of concrete information, verbal information, and rule learning (Ho, Savenye, & Haas, 1986).

Synthesizers. A synthesizer, a device for review that leads to deeper processing and better retention, can help students periodically interrelate and integrate the individual ideas that have been learned. Synthesizers compare and contrast related ideas and show how new ideas fit within the overall knowledge structure to increase meaningfulness and motivation. In general, a synthesizer presents a generality, shows the relationships among the parts and with related ideas, includes integrated reference examples, and provides diagnostic self-test practice items (Reigeluth & Stein, 1983).

DOMAIN 3: STUDENT PERFORMANCE

The events related to the student performance domain occur after the cognitive stage. These include eliciting performance, assessing performance, and providing appropriate feedback. Although performance elicitation and assessment constitute two of Gagne's (1975) nine events leading to learning single concepts, the principles guiding them are all similar. Therefore, eliciting performance and assessing performance will be discussed as a single unit. The following guidelines apply to interim or final performance in elicitation and assessment situations, but they can also apply to eliciting performance to enhance learning retention and transfer.

Eliciting Performance and Assessing Performance

Eliciting and assessing performance refers to causing students to perform, or respond to questions, throughout the instruction. Judgments on the quality of the responses are then made (Wager & Gagne, 1988). An overt response is not needed for learning to take place--an instructor can simply ask learners to make a self-estimate of their understanding. However, a response is needed if an external, criterion-based assessment is to be made (Jonassen & Hannum, 1987).

Objective-Performance Match and Active Participation. Responses to carefully designed questions will indicate or represent cognitive change (Scandura, 1983). Two major principles guide performance elicitation. First, responses should be required that match the level of processing specified by the objectives (Gropper, 1983; Jonassen & Hannum, 1987). Second, active participation is superior to reactive participation (Jonassen & Hannum). Each objective should be evaluated to determine the number and type of questions needed for accurate assessment. The simplest type of task--rote-recall items--require one question, or possibly more if concepts are paraphrased. The next level--rule application--requires as many questions as needed to cover all aspects of the objective. Deep-processing tasks--synthesis--usually dictate the type and quantity of questions (Merrill, 1983). A concept should be tested at all three levels in order to determine if it has been truly learned (Merrill). For example, if the lesson taught concepts about musical chords, the student should be able to remember the characteristics of a particular triad, use those characteristics to build his or her own examples, and be able to identify various uses of a particular chord.

Performance Realism, Time Allocation, and Acceptable Standards. Students should be required to perform in conditions that closely represent the situation in which the skill is normally required (Gropper, 1983; Montague, 1988; Salisbury, 1988). Merrill (1983) recommended that the time allowed for answering questions should increase as the level of cognitive processing deepens. For rote recall there should be no time delay; higher level tasks require some time, and synthesis tasks should have no time limit. Performance demands should be within the learners' reach in order to enhance achievement-striving behavior (Keller, 1983); therefore, designers may accept lower standards at early stages in the learning process (Gropper, 1983; Hannafin & Peck, 1988).

Instrument Reliability and Validity. When designing computer-based instruction for large-group applications, establishment of reliability and validity of the test instrument is imperative (Palumbo & Reed, 1988; Roblyer, 1988). This gives the user a sense of the thoroughness of product development and the confidence to use the supplied instrument.

Although test reliability and validity are important in all learning environments, the computer is especially well suited for this process. It patiently waits for students to absorb information and formulate responses; it provides all learners with the same level, quantity, and quality of assessment procedures; and it simulates situations that resemble or closely represent "real" situations.

Providing Feedback

Providing students with information about their own learning facilitates performance (Jonassen & Hannum, 1987); therefore, computer-based learning activities should require learner responses and provide feedback about those responses in a positive and informative manner (Aronson & Briggs, 1983). Four types of feedback are simple yes/no, correct answer feedback (yes/no plus the correct answer), explanatory (brief or lengthy explanations), and bug-related (identification of logic errors) (Schimmel, 1988). These four types of feedback lie on a continuum, in the order just given, from least to most useful for learning enhancement.

Correct/Incorrect. Simple yes/no feedback may be acceptable for rote-recall items (Merrill, 1983), but, even when presented as flashy cartoon graphics, it does not enhance motivation or achievement (Surber & Leeder, 1988). In general, simple yes/no feedback should be avoided. Similarly, simple positive comments interspersed throughout a lesson do not produce higher levels of motivation (Jonassen & Hannum, 1987).

Correct Answer. Correct answer feedback presents the correct answer to the learner and confirms a correct or incorrect response. This is the most frequently used type of feedback (Jonassen & Hannum, 1987; Merrill, 1983; Schimmel, 1988). Little effort is needed to include the correct answer along with the yes/ no response. Therefore, this should be the minimum level of feedback used in computer-based learning.

Explanatory. Explanatory feedback is appropriate for intellectual skill development because it identifies errors, provides guidance on the correct response, and gives praise for correct answers (Schimmel, 1988). Errors should be corrected according to the following priorities: errors before omissions, prior steps before later steps, shorter fixes before longer fixes, and more-important factors before less-important factors (Collins & Stevens, 1983). Providing guidance and giving praise also functions as a motivational strategy because it increases the learners' intrinsic satisfaction with the instruction by attributing success in the learning task to the student (Keller, 1983; Merrill, 1983).

Bug-Related. Bug-related feedback corrects the student by directly addressing specific procedural errors (Schimmel, 1988). This type of feedback depends on the computer's ability to identify the error(s) made by the student. This identification proves difficult because of learner inconsistencies and the complexity of a program containing bug libraries or sets of rules that match all possible student errors.

Other Feedback Issues

Feedback Control. When learning involves intellectual skills and a target audience of high achievers, the student or program should choose the appropriate type of feedback because high-ability students generally use feedback more constructively than low-ability students, who may choose a level that gives answers rather than promotes thinking skills (Schimmel, 1988). Ideally, the program should control the feedback type based on user performance, but high-ability students will probably choose an appropriate level if presented with the option.

Timing. The timing of feedback delivery can have differential effects on learning and motivation (Keller, 1983). Immediate feedback is more effective than delayed feedback for tasks requiring students to make decisions and learn new information (Jonassen & Hannum, 1987). It also motivates students to maintain the quantity of responses (Keller, 1983). To increase the quality of performance, Keller recommended that corrective feedback be given at the most useful time--usually just before subsequent opportunities to practice. End-of-session feedback is important when the learning task involves abstract concepts, especially for higher achieving learners (Jonassen & Hannum, 1987).

Relevance to Content and Learner. Feedback should be related to the content of the lesson. For example, using arbitrary point systems or money-type rewards for tasks not related to money generally do not motivate students to perform better (Keller, 1983). However, interjection of unexpected, noncontingent rewards as opposed to anticipated rewards can heighten motivation levels when used prudently and judiciously (Jonassen & Hannum, 1987; Keller).

Feedback for Low Achievers. Low achievers need more feedback than average or above-average learners (Blanchard, 1987; Schimmel, 1988). Although important for all learners, feedback that encourages low achievers by focusing on the correct portions of answers that are not completely right is especially important. Two additional help strategies include restating difficult questions in different forms to help the student connect the question with the original concept and providing pre-response help or elaboration that gives relevant information to help the student answer the question (Merrill, 1983).

Feedback Summary

Learning typically needs to be assessed by comparing student performance with criterion performance, and learners need to be informed of their successes or failures through feedback. The way the feedback is presented facilitates or debilitates further learning. Courseware can be designed to provide adaptable feedback based on learner responses and to recognize responses that are not entirely correct.

DOMAIN 4: OTHER DESIGN ISSUES

So far, the discussion has included instructional design theories and techniques that, although especially well suited for computer-based instruction, apply to all learning regardless of the delivery mode. The computer has the potential for presenting efficient, interesting, and adaptable instruction; however, other inherent issues affect courseware design. Learner control, fear reduction, cuing, mnemonics, and personalization are often part of a classroom teacher's automatic professional routine and are also important issues when designing computer-based instruction.

Learner Control

Computers can provide opportunities for learners to choose particular lessons at any given time, the number of questions to be answered, and the amount of time to be spent on particular parts of a lesson. This control over a lesson is often touted as a reason that learners will work harder and achieve more when using computer-based instruction; however, this is often not the case. Although some researchers have reported higher achievement because of learner control (Gay, Trumbull, & Smith, 1988), other investigators have not found achievement differences between students who have control over program sequence and students forced to follow a given linear path (Canino & Cicchelli, 1988; Hannafin & Colamaio, 1987; Hines & Seidman, 1988; Tennyson & Buttrey, 1980). Reigeluth and Stein (1983) asserted that only simple-to-complex sequences allow learners to make informed decisions about sequence. Therefore, learner control should be incorporated into the various elaboration strategies whenever possible, but only when the learners are properly instructed in the effective use of control strategies. Formatting options provide one means of accomplishing control. The instructional components should be clearly identified and separated to facilitate learners' selection and sequencing according to their needs and interests. This is most commonly achieved through menu-driven programs (Keller & Suzuki, 1988). Knowledge-structure enhancement devices, including stimulating review and study strategies, should also be presented to enhance learners' competency of use and to stimulate informed interaction with the content (Jonassen & Hannum, 1987; Reigeluth & Stein, 1983).

Jonassen and Hannum (1987) maintained that it is not effective to give learners control over instructional method, amount of instructional support, specific instructional events, instructional sequence, or difficulty of practice items. Variables to consider in addition to achievement include attitude, learner ability, time on task, and pacing.

Attitude. Student attitudes can be affected by the amount of control they have over a lesson. Students choosing their own program sequence usually have more positive attitudes than students without control over sequence (Canino & Cicchelli, 1988; Milheim, 1989). Also, expectations for success can be increased by providing opportunities for choice, responsibility, and control that reflect and support internal attributions for success (Jonassen & Hannum, 1987; Keller, 1983). However, a problem exists. Students with control over the lesson have better attitudes, but may achieve less, or, at best, an equivalent amount, than when they do not have control. Research on this problem has led to an answer. Courseware can be constructed to give the learner control over program sequence and branching opportunities. Instead of leaving the choice completely to the user, however, the program can be designed to offer advice based on pedagogy, or on student performance in more complex programs. This will show learners the ideal path through the learning task (Jonassen & Hannum, 1987; Ross & Morrison, 1988; Tennyson, 1984). Students who ultimately have control over a lesson and are advised about sequence by the program may not perform better than those forced to follow a predetermined sequence, but they will have a more positive attitude toward the lesson (Hannafin & Colamaio, 1987).

Learner Ability. The advisement strategy also compensates for the problem of program suitability for learners with different ability levels. Lower level students who have limited prior knowledge about a learning task are generally poor judges of their learning needs (Hannafin & Hooper, 1988). They usually select too little program support, while higher level learners usually select too much (Ross & Morrison, 1988). Lower level students usually follow advice regarding program sequence, but higher level students and those with some previous knowledge skip parts of the program they already know. Therefore, a program with advisement can be used for a wider range of learners.

Time on Task. Students in a learner-control group may attain achievement levels equal to those of students in advisement strategy groups, but the advisement-group students may complete the lesson in less time because they are advised to move on as soon as they reach criterion level for a particular concept (Tennyson & Buttrey, 1980).

Lesson Pace. A final aspect of learner control is the obvious advantage of students proceeding at their own pace. The computer allows students to choose their own pacing (Milheim, 1989). However, student pacing may not always be appropriate. The program may teach or have a student practice a skill that should be performed quickly, such as developing automaticity of response, quick assessment, or decision making. In these time-critical situations, computer pacing serves an especially important purpose.

Teaching Tools

Anxiety Reduction. Anxiety can affect a student's ability to learn. One cause of anxiety is exposure to new technology, which can result in a variety of emotional and/or physiological responses, either real or imagined (Wicks, 1977). According to Wicks, relaxation, reassurance, and learning contribute to overcoming the fear that causes anxiety. Because anxiety resulting from initial exposure to computer technology can be significantly reduced with as little as six hours of introductory instruction (Overbaugh & Reed, 1991), activities designed specifically to familiarize students with the computer as a tool may be useful. Anxiety can also be reduced by permitting group work and using shared-sum scoring, although specific task assignments for each group member are important (Keller, 1983).

Cuing. Cues refer to instructional devices employed to direct the learners' attention to certain points or relationships in memorization or rule-use tasks, either through screen design or text. Screen-based cues include the use of highlighting, color, arrows, underlining, shading, and moving text to focus learners' attention on particular points. Text-based cues clarify or elaborate on associations between concepts and examples. To avoid student dependency on cues, both types of cues should be phased out as the student learns more about the concepts (Gropper, 1983).

Mnemonics. Mnemonics can be employed to help students with rote-recall tasks (Jonassen & Hannum, 1987; Merrill, 1983). The computer presents strategies such as peg words, acronyms, and stories using key words or facts and key-letter sentences. Even better, the courseware can assist the learner in constructing and using his or her own mnemonics by providing a "note pad" window in which students can create and save their mnemonics for later use.

Personalization. Using personalization can increase interest and motivation (Jonassen & Hannum, 1987). Techniques include using students' names, background information, and interests in the presentation. This technique has been used successfully in teaching elementary children abstract concepts, but it should be phased out because the novelty wears off (Ross & Morrison, 1988).

OTHER STAGES

Clearly, the first stage--instructional design is the most time-consuming, encompassing, and rigorous task for courseware developers. After this stage is completed, the job is essentially complete except for converting all the planning into an actual piece of courseware. The second step--preprogramming development-involves only three steps: creating flow charts and story boards to aid in the actual programming; developing all support materials for the courseware, including any supplemental instructional materials; and reviewing all instructional and support materials before the actual programming begins (Roblyer, 1988).

The final stage is the relatively straightforward task of programming the courseware. The instructional designer and the programmer should work together during programming, including first-draft evaluation and revisions (Roblyer, 1988).

Research has shown that courseware can be beneficial to learners in many situations, especially when it is based on sound learning principles. Many educators without programming skills use authoring systems as tools to develop courseware. This review has examined instructional design theory and research relevant to computer-based learning in order to provide some courseware design guidelines, or prescriptions, that can be used by beginning courseware developers.

References

Anderson, R.C., Spiro, R.J, & Anderson, M.C. (1978). Schemata as scaffolding for the representation of information in connected discourse. American Educational Research Journal, 15, 433-440.

Aronson, D.T., & Briggs, L.J. (1983). Contributions of Gagne and Briggs to a prescriptive model of instruction. In C.M. Reigeluth (Ed.), Instructional-de-sign theories and models (pp. 75-100). Hillsdale, NJ: Lawrence Erlbaum Associates.

Blanchard, J.S. (1987). Instructional design issues in computer-based reading: Reinforcement and objectives. Computers in the Schools, 4(1), 25-35.

Canino, C., & Cicchelli, T. (1988). Cognitive styles, computerized treatments on mathematics achievement and reaction to treatments. Journal of Educational Computing Research, 4(3), 253-64.

Carrier, C.A., & Jonassen, D.H. (1988). Adapting courseware to accommodate individual differences. In D.H. Jonassen (Ed.), Instructional designs for microcomputer courseware (pp. 203-226). Hillsdale, NJ: Lawrence Erlbaum Associates.

Collins, A., & Stevens, A.L. (1983). A cognitive theory of inquiry teaching. In C.M. Reigeluth (Ed.), Instructional-design theories and models (pp. 247-278). Hillsdale, NJ: Lawrence Efibaum Associates.

Denenberg, S.A. (1988). Semantic network designs for courseware. In D.H. Jonassen (Ed.), Instructional designs for microcomputer courseware (pp. 307326). Hillsdale, NJ: Lawrence Erlbaum Associates.

Gagne, R.M. (1975). Essential of learning for instruction. New York: Holt, Rinehart, and Winston.

Gagne, R.M. (1986). Instructional technology--The research field. Journal of Instructional Development, 8(3), 7-14.

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

Gay, G., Trumbull, D., & Smith, J. (1988). Perceptions of control and use of control options in computer-assisted video instruction. Tech Trends, 33(5), 31-33.

Gropper, G.L. (1983). A behavioral approach to instructional prescription. In C.M. Reigeluth (Ed.), Instructional-design theories and models (pp. 37-54). Hillsdale, NJ: Lawrence Efibaum Associates.

Hamilton, R.J. (1985). A framework for the evaluation of the effectiveness of adjunct questions and objectives. Review of Educational Research, 55(1), 47-85.

Hannafin, M.J., & Colamaio, M.E. (1.987). The effects of variations in lesson control and practice on learning from interactive video. Educational Communication and Technology Journal, 35(4), 203-12.

Hannafin, M.J., Garhart, C., Rieber, L.P., & Phillips, T.L. (1985). Keeping interactive video in perspective: Tentative guidelines and cautions in the design of interactive video. In E. Miller & M.L. Mosely (Eds.), Educational media and technology yearbook (pp. 67-92). Littleton, CO: Libraries Unlimited.

Hannafin, M.J., & Hooper, S. (1988,January). Empirically-based guidelines for designing interactive video. Paper presented at the annual meeting of the American Educational Research Association, New Orleans, LA.

Hannafin, M.J., & Peck, K.L. (1988). The design, development and evaluation of instructional software. New York: Macmillan Publishing Company.

Hannum, W. (1988). Designing courseware to fit subject matter structure. In D.H. Jonassen (Ed.), Instructional designs for microcomputer courseware (pp. 275-296). Hillsdale, NJ: Lawrence Erlbaum Associates.

Hativa, N., & Teper, A. (1988). Differential effectiveness of three color treatments in learning geometric concepts via computer-guided teaching. Journal of Educational Computing Research 4(3), 303-20.

Hines, S.J., & Seidman, S.A. (1988,January). The effects of selected CAI design strategies on achievement, and an exploration of other related factors. Paper presented at the 1988 annual meeting of the Association for Educational Communications and Technology, New Orleans, LA. (ERIC ED 295 650)

Ho, C.P., Savenye, W., & Haas, N. (1986). The effects of orienting objectives and review on learning from interactive video. Journal of Computer-Based Instruction, 12(4), 126-29.

Jonassen, D.H. (1988). Integrating learning strategies into courseware to facilitate deeper processing. In D.H. Jonassen (Ed.), Instructional designs for microcomputer courseware (pp. 151-182). Hillsdale, NJ: Lawrence Erlbaum Associates.

Jonassen, D.H., & Hannum, W.C. (1987,December). Research-based principles for designing computer software. Instructional Technology, pp. 7-14.

Kanner, J.H. (1968). The instructional effectiveness of color in television: A review of the evidence. Stanford, CA: Stanford University. (ERIC ED 015 675)

Keller, J.M. (1983). Motivational design of instruction. In C.M. Reigeluth (Ed.), Instructional-design theories and models (pp. 383-436). Hillsdale, NJ: Lawrence Erlbaum Associates.

Keller, J.M., & Suzuki, K. (1988). Use of the ARCS motivation model in courseware design. In D.H. Jonassen (Ed.), Instructional designs for microcomputer courseware (pp. 401-434). Hillsdale, NJ: Lawrence Erlbaum Associates.

Landa, L.N. (1983). The algo-heuristic theory of instruction. In C.M. Reigeluth (Ed.), Instructional-design theories and models (pp. 55-74). Hillsdale, NJ: Lawrence Erlbaum Associates.

Mayer, R.E. (1984). Aids to text comprehension. Educational Psychologist, 19, 30-42. Merrill, M.D. (1983). Component display theory. In C.M. Reigeluth (Ed.), Instructional-design theories and models (pp. 279-327). Hillsdale, NJ: Lawrence Erlbaum Associates.

Merrill, M.D. (1988). Applying component display theory to the design of courseware. In D.H. Jonassen (Ed.), Instructional designs for microcomputer courseware (pp. 61-96). Hillsdale, NJ: Lawrence Erlbaum Associates.

Merrill, P.F., & Bunderson, C.V. (1979). Guidelines for employing graphics in a videodisk training system. ISD for videodisc training systems. (ERIC ED 196 413)

Milheim, W.D. (1989,February). Perceived attitudinal effects of various types of learner control in an interactive video lesson. Paper presented at the annual meeting of the Association for Educational Communications and Technology, Dallas, TX. (ERIC ED 308 828)

Montague, W.E. (1988). Promoting cognitive processing and learning by designing the learning environment. In D.H. Jonassen (Ed.), Instructional designs for microcomputer courseware (pp. 125-150). Hillsdale, NJ: Lawrence Erlbaum Associates.

Overbaugh, R.C., & Reed, W.M. (1991). The effects of prior experience and instructional format on teacher education students' computer anxiety and performance. Paper presented at the annual meeting of the American Educational Research Association, Chicago, IL.

Palumbo, D.B., & Reed, W.M. (1988). A microcomputer-based evaluation subsystem for the textual component of an instructional system. Journal of Educational Computing Research, 4(2), 167-83.

Phillips, T.L., Hannafin, M.J., & Tripp, S.D. (1988). The effects of practice and orienting activities on learning from interactive video. Educational Computing and Technology Journal, 36(1), 93-102.

Reeves, T.C. (1986). Research and evaluation models for the study of interactive video. Journal of Computer-Based Instruction, 13(4), 102-106.

Reigeluth, C.M., & Stein, E S. (1983). The elaboration theory of instruction. In C.M. Reigeluth (Ed.), Instructional-design theories and models (pp. 335-382). Hillsdale, NJ: Lawrence Erlbaum Associates.

Roblyer, M.D. (1988). Fundamental problems and principles of designing effective courseware. In D.H. Jonassen (Ed.), Instructional designs for microcomputer courseware (pp. 7-34). Hillsdale, NJ: Lawrence Erlbaum Associates.

Ross, S.M., & Morrison, B.R. (1988). Adapting instruction to learner performance and background variables. In D.H. Jonassen (Ed.), Instructional designs for microcomputer courseware (pp. 227-246). Hillsdale, NJ: Lawrence Erlbaum Associates.

Salisbury, D. (1988). Effective drill and practice strategies. In D.H. Jonassen (Ed.), Instructional designs for microcomputer courseware (pp. 103-124). Hillsdale, NJ: Lawrence Erlbaum Associates.

Scandura, J.M. (1983). Instructional strategies based on the structural learning theory. In C.M. Reigeluth (Ed.), Instructional-design theories and models (pp. 213-246). Hillsdale, NJ: Lawrence Erlbaum Associates.

Schimmel, B.J. (1988). Providing meaningful feedback in courseware. In D.H. Jonassen (Ed.), Instructional designs for microcomputer courseware (pp. 183-196). Hillsdale, NJ: Lawrence Erlbaum Associates.

Surber, J.R., & Leeder, J.A. (1988). The effect of graphic feedback on student motivation. Journal of Computer-Based Instruction, 15(1), 14-17.

Tennyson, R.D. (1984). Artificial intelligence methods in computer-based instructional systems: The Minnesota Adaptive Instructional System. Journal of Instructional Development, 7(3), 17-22.

Tennyson, R.D., & Buttrey, T. (1980). Advisement and management strategies as design variables in computer-assisted instruction. Educational Communications and Technology Journal, 28, 169-176.

Wager, W., & Gagne, R.M. (1988). Designing computer-aided instruction. In D.H. Jonassen (Ed.), Instructional designs for microcomputer courseware (pp. 35-60). Hillsdale, NJ: Lawrence Erlbaum Associates.

Wicks, R.J. (1977). Counseling strategies and intervention techniques for the human services. Philadelphia, PA: J.B. Lippincott Company.

~~~~~~~~~

By Richard C. Overbaugh, Old Dominion University

Richard C. Overbaugh received his master's degree and doctorate in education from West Virginia University. He is presently Assistant Professor of Instructional Computing at Old Dominion University in Norfolk, Virginia. (Address: Old Dominion University, Department of Educational Curriculum and Instruction, Norfolk, VA 23529-0161.)
~~~~~~~~~