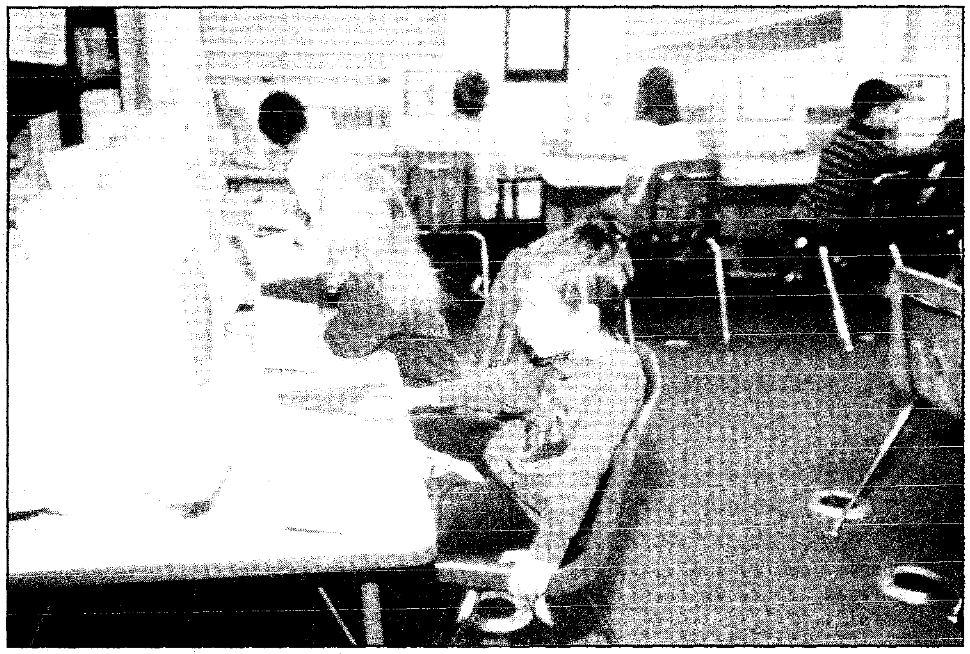
4 CHAPTER

Integrating Instructional Software into Teaching and Learning



The fact that individuals bind themselves with strong emotional ties to machines ought not to be surprising. The instruments [we] use become . . . extensions of [our] bodies.

Joseph Weizenbaum in
Computer Power and Human Reason (1976, p. 9)

This chapter covers the following topics:

- Definitions, issues, integration strategies, and technology integration ideas based on a directed instructional model for:
  - Drill-and-practice functions
  - Tutorial functions
- Definitions, issues, integration strategies, and technology integration ideas based on both directed and constructivist models for:
  - Simulation functions
  - Instructional game functions
  - Problem-solving functions
- Characteristics and uses of integrated learning systems (ILSs)
- Criteria and methods for evaluating and selecting software

Objectives

1. For each description of a classroom need for instructional materials, identify one or more types of instructional software functions that could meet the need.
2. Plan lesson activities that integrate instructional software using a directed learning strategy.
3. Plan lesson activities that integrate instructional software using a constructivist learning strategy.

Introduction

What Is Instructional Software?

The *Newware* example described in Chapter 2 (Figure 2.6) illustrates an example of what Weizenbaum (1976) called using computers as "extensions of [our] bodies." Such uses have a long history in education. From the time people began to recognize the potential power of a computer to do tasks quickly and systematically, they also began exploring and experimenting with its capability to emulate and improve on the functions of a human teacher. If computer programs could be written to do essentially anything, why could not computers be programmed to teach? Many educators and developers pursued this goal of the computer as teacher during the 1960s and 1970s. Some, like William Norris (1977) who developed Control Data's PLATO teaching systems, believed that computer-based education was the only logical alternative to education's "outdated, labor-intensive ways" (p. 451). Norris believed that education could become more productive if computers were to take over much of the traditional role of teachers.

Today, after about 30 years of development and experimentation, there is less talk of computers replacing teachers, but programs like the one described in the Figure 2.6 example still exist that perform various teaching functions. While these programs are not alternatives to human teachers, as envisioned by Norris, they can enhance teaching and learning in many ways. This chapter shows how programs like the one in the Figure 2.6 lesson empowers teachers, rather than replaces them.

Products written in computer languages (e.g., Basic, Assembler, C++, Java) and which are designed and developed to perform tasks are called *applications software* or *programs*. Instructional software (or courseware) is applications software that is designed specifically to deliver or assist with student instruction on a topic. Although software such as word processing and spreadsheet programs also can enhance instructional activities, this textbook differentiates between such tools and instructional software. Software tools serve a variety of purposes other than teaching; instructional software packages are programs developed for the *sole purpose* of delivering instruction or supporting learning activities.

Problems in Identifying and Classifying Software Functions

Computer-assisted instruction (CAI) originated in the early days of educational technology as a name for instructional software, and the term is still in common use. However, some kinds of instructional software are designed with more constructivist purposes in mind and do not actually deliver instruction per se; therefore many people consider the term CAI outdated and misleading. Teachers may hear instructional software referred to as *computer-based instruction (CBI)*, *computer-based learning (CBL)*, or *computer-assisted learning* along with more generic terms such as *software learning tools*.

Names for the types of instructional software functions also vary, but they are usually identified as drill and practice, tutorial, simulation, instructional game, and problem solving. Although these terms originated because each type had clearly different characteristics and uses, much of today's software defies easy classification for three reasons:

1. **Developers use terms interchangeably.** There seems to be no consensus among developers for what terms to use to describe various types of programs. Some developers refer to a drill program that gives extensive feedback as a *tutorial*. Others refer to *simulations* or *problem-solving* functions as *games*.
2. **Packages contain more than one activity.** Many software packages contain several different activities, each of which serves a different purpose. For example, a program like *Millie's Math House* has a number of straight drill activities along with some problem-solving and game activities.
3. **Software is becoming multimedia.** Tergan (1998) notes that since more software is incorporating hypermedia and multimedia environments (including Internet links), it makes it more difficult to analyze learner–system interactions, isolate instructional roles in a software package, and identify the type of software functions involved.

In light of these issues, educators who use software for instruction will find it useful to analyze all of the activities in a package and classify each one according to its instructional functions. For example, one may not be able to refer to an entire package as a tutorial or a drill, but it is possible and desirable to identify a particular activity according to whether it provides skill practice or opportunities for solving problems. As this chapter will show, each software function serves different purposes during learning and, consequently, has its own appropriate integration strategies.

Insights on Software Classifications and Integration Strategies

Gagné, Wager, and Rojas (1981) suggested a way to look at courseware that can help educators analyze a given product as to its instructional function(s) and design appropriate integration strategies that make use of these functions. Gagné et al. observed that drills, tutorials, and simulations each accomplish a different combination of the Events of Instruction (see them in Chapter 3, InSight 3.3 on Gagné's principles). The nine events are a set of guidelines identified by Gagné that can help teachers arrange optimal "conditions for learning" for various types of knowledge and skills. By determining which of the events a courseware package fulfills, educators can determine the teaching role it serves and where it might fit in the instructional process. The five common courseware types accomplish the following functions:

- **Drills (or drill and practice)** allow learners to work problems or answer questions and get feedback on correctness (accomplishes events 6 and 7).
- **Tutorials** act like human tutors by providing all the information and instructional activities a learner needs to master a topic: information summaries, explanation, practice routines, feedback, and assessment. (Gagné et al. say that a tutorial should accomplish all nine events. However, depending on how it is implemented, it can accomplish at least events 3 through 8).
- **Simulations** model real or imagined systems to show how those systems or similar ones work (accomplishes events 6 and 7, and usually also accomplishes events 2, 4, and 5).
- **Instructional games** are designed to increase motivation by adding game rules to learning activities; usually either drills or simulations (usually accomplishes events 1, 6, and 7).
- **Problem-solving programs** teach directly, through explanation and/or practice, the steps involved in solving problems or help learners acquire problem-solving skills by giving them opportunities to solve problems (can accomplish events 3 through 7 and event 9).

When a teacher evaluates a courseware package for possible use, a recommended strategy is to analyze and identify which Events of Instruction each activity accomplishes, classify it as to type(s), and then design one or more integration strategies that make effective use of its functions. For example, the software and integration example described in Figure 2.6 accomplishes events 5, 6, and 7.

Programming Languages as Instructional Software

This chapter focuses on classroom uses of instructional software, while Chapters 5 and 6 address productivity and instructional uses of the resources known as software tools. However, programming languages may be considered a hybrid software, merging the capabilities of both instructional software and tools. Programming languages were created to develop computer programs that make computers do various tasks. For example, word processing programs are written in programming languages as are drills, tutorials, and other forms of instructional software. However, teachers also use programming languages as a tool to teach as well as to develop programs. (See Technology Integration Idea 4.1.)

One of the most widely known of the programming languages used for instruction is Logo. The work of Seymour Papert (see Chapter 3) and his colleagues at the Massachusetts Institute of Technology made Logo "widely used throughout the world as an introductory programming language and mathematical learning environment for students in elementary and secondary schools" (p. 615). Papert hoped that it would become "a context which is to learning mathematics what living in France is to learning French" (p. 6).

Although not as popular as it was in the 1980s, Logo and its derivative materials such as *Microworlds* software

TECHNOLOGY INTEGRATION IDEA

4.1

Using Programming Languages

TITLE: Problem Solving à la vos Savant

CONTENT AREA/TOPIC: Logic and organizational skills

GRADE LEVEL: Middle to high school

NETS FOR STUDENTS: Standards 1, 6

DESCRIPTION: For high school students, beginning programming is a good way to gain practice in organizational skills that are fundamental to problem solving. This activity is based on a problem from Marilyn vos Savant's *Parade Magazine* column "Ask Marilyn." The column posed this question: "Suppose you are on a game show and you're given a choice of three doors. Behind one is a car; behind the others, goats. You pick a door—say No. 1—and the host, who knows what's behind the other doors, opens another door—say No. 3—which has a goat. He then says to you, 'Do you want to pick door 2?' Is it to your advantage to switch your choice?" (p. 12). Marilyn vos Savant said the odds of winning the car increase from 1-in-3 to 2-in-3 by switching doors. Ask students if they agree. Then show them how to create a program to test their solution.

Source: Wagner, P. (1992). Gamer 1, 2, 3: The vas (sic) Savant challenge. *The Computing Teacher, 19*(5), 12–14.

still are used for many instructional purposes. Logo often is used to introduce young children to problem solving through programming and to explore concepts in content areas such as mathematics, science, and language arts (Galas, 1998; Gonsalves & Lopez, 1998; Weinstein, 1999). Logo also led the way for other programming resources to be used in the same instructional ways. Ploger and Vedova (1999) describe a "programming system" called *Boxer*, which combines a programming language with text processing and graphics capabilities to help students learn number sense concepts.

Drill and Practice Activities

Drill and Practice: Definition and Characteristics

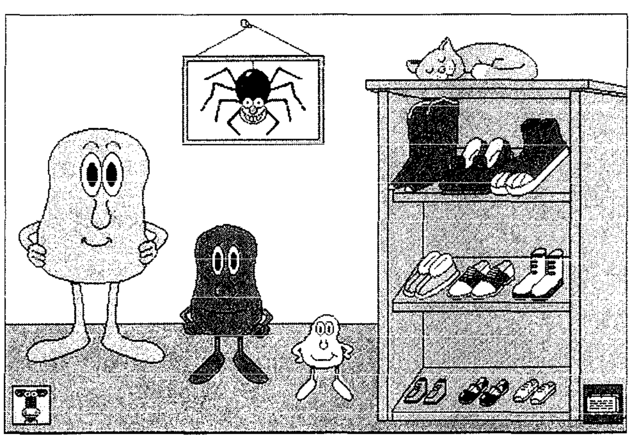
Drill and practice activities provide exercises in which students work example items, usually one at a time, and receive feedback on their correctness. Programs vary considerably in the kind of feedback they provide in response to student input. They range from a simple display like "OK" or "No, try again" to elaborate animated displays or verbal explanations. Some programs simply present the next item if the student answers correctly.

Types of drill and practice are sometimes distinguished by the sophistication with which the program tailors the practice session to student needs (Merrill & Salisbury, 1984). The most basic drill and practice function often is described as a *flashcard activity*. A student sees a set number of questions or problems on the screen and answers one at a time. Examples of instructional software that reflect this type of function are shown in Figures 4.1 and 4.2.

A more sophisticated form of drill and practice moves students on to advanced questions after they get a number of questions correct at some predetermined mastery level; it may also send them back to lower levels if they answer a certain number wrong. Some programs automatically review questions that students get wrong before going on to other levels. Movement between levels often is transparent to students since the program may do it automatically without any indication of what it is doing. Sometimes, however, the program may congratulate students on good progress before proceeding to the next level, or it may allow them to choose their next activities.

Figure 4.1 Drill and Practice in the Word Concepts *small, medium,* and *large* from *Edmark's Millie's Math House*

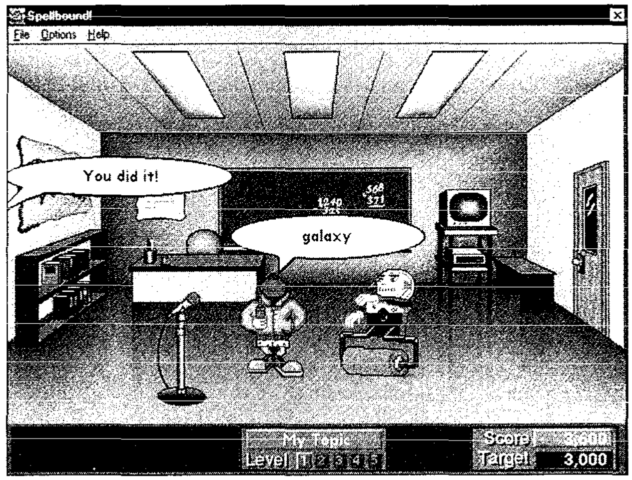
Student selects shoes from the shelf that fit each foot.



Source: Used by permission of Edmark Corporation.

Figure 4.2 Drill and Practice in Spelling Words from Encore Education Software's *Elementary Advantage*

After reviewing a list of various spelling words and definitions, the computer pronounces a word as its definition appears on the screen. The student types in the word, and the computer tells whether the spelling is correct.



Source: Used by permission of Encore Education Software.

In addition to meeting general criteria for good instructional courseware (see listing and discussion of courseware criteria in Figure 4.12 later in this chapter), well-designed drill and practice programs should also meet other criteria:

Control over the presentation rate. Unless the questions are part of a timed review, students should have as much time as they wish to answer and examine the feedback before proceeding to later questions. If the program provides no specific feedback for correct answers, it usually is acceptable to present later questions without any further entries from students.

Appropriate feedback for correct answers. Although some courseware designers stress the importance of positive feedback for correct answers, not all programs provide it. If students' answers are timed, or if their session time is limited, they may find it more motivating simply to move quickly to later questions. Positive feedback should not be so elaborate and time consuming that it detracts from the lesson's purpose. No matter how attractive the display, students tend to tire of it after a while and it ceases to motivate them.

Better reinforcement for correct answers. Some programs inadvertently motivate students to get wrong answers. This happens when a program gives more exciting or interesting feedback for wrong answers than for correct ones. The most famous example of this design error occurred in an early version of a popular microcomputer-based math drill series. Each correct answer got a smiling face, but two or more wrong answers produced a full-screen, animated crying face that students found very amusing. Consequently, many students tried to answer incorrectly to see it. The company corrected this flaw, but this classic error still exists today in other programs.

Issues Related to Drill and Practice

Drill and practice courseware activities were among the earliest and most well-recognized instructional uses of computers and are still used extensively in schools. These activities have frequently been shown to allow the effective rehearsal students need to transfer newly learned information into long-term memory (Merrill & Salisbury, 1984; Salisbury, 1990). However, drill and practice is also the most maligned of the courseware activities, sometimes informally referred to among its critics as "drill and kill." This derision results, in part, from perceived overuse. Many authors have criticized teachers for presenting drills for overly long periods or for teaching functions that drills are ill suited to accomplish. For example, teachers may expose students to drill and practice courseware as a way of introducing new concepts rather than just practicing and reinforcing familiar ones.

But probably the most common reason for the virulent criticism of drill and practice courseware is its identification as an easily targeted icon for what many people consider an outmoded approach to teaching. Critics claim that introducing isolated skills and directing students to practice them directly contradicts the trend toward restructured curriculum in which students learn and use skills in an integrated way within the context of their own projects that specifically require the skills.

Although curriculum increasingly emphasizes problem solving and higher order skills, teachers still give students on-paper practice (e.g., worksheets or exercises) for many skills to help them learn and remember correct procedures. Many teachers feel that such practice gives students more rapid recall and use of basic skills as prerequisites to advanced concepts. They like students to have what Gagné (1982) and Bloom (1986) call *automaticity* or *automatic recall* of these lower order skills to help them master higher order ones faster and more easily. Kahn (1998–1999) cites drill and practice as a worthwhile software substitute for paper worksheets. The usefulness of drill programs in providing this kind of practice has been well documented, but the programs seem especially popular among teachers of students with learning disabilities (Hasselbring, 1988; Higgins & Boone, 1993; Okolo, 1992). The following are examples of skills for which students could use a drill program to gain necessary proficiency:

- Automatic recall of arithmetic facts is required for most higher level mathematics ranging from long division to algebra.
- Keyboard proficiency is a prerequisite for assignments that require extensive typing.
- Graded compositions require rapid recall and application of correct sentence structure, spelling, and principles of grammar and usage.
- Many schools still require students to memorize facts such as states and capitals and names of planets.
- College entrance exams and other standardized tests require quick recall of many facts

Despite the increasing emphasis on problem solving and higher order skills, it is likely that some form of drill and practice courseware probably will be useful in many classrooms for some time to come. Such programs address needs for these and other required skills. Rather than ignoring drill and practice software as outmoded, teachers should seek to select and use these kinds of programs for uses they can best accomplish.

How to Use Drill and Practice in Teaching

Benefits of drill functions. Drill and practice programs may be used whenever teachers feel the need for on-paper exercises such as worksheets. Drill courseware provides several acknowledged benefits as compared to paper exercises:

- **Immediate feedback.** When students practice skills on paper, they frequently do not know until much later whether or not they did their work correctly. To quote a common saying, "Practice does not make perfect; practice makes permanent." As they complete work incorrectly, students may actually be memorizing the wrong skills. Drill and practice courseware informs them immediately whether or not their responses are accurate, so they can make quick corrections. This helps both "debugging" (identifying errors in their procedures) and retention (helps to place the skills in long-term memory for ready access later).
- **Motivation.** Many students refuse to do the practice they need on paper, either because they failed so much that the whole idea is abhorrent, or they have poor handwriting skills, or simply dislike writing. In these cases, computer-based practice may motivate students to do the practice they need. Computers don't get impatient or give disgusted looks when a student gives a wrong answer.
- **Saving teacher time.** Since teachers do not have to present or grade drill and practice, students can do this activity essentially on their own while the teacher addresses other student needs.

Classroom applications of drill functions. On some occasions, even the most creative and innovative teacher may take advantage of the benefits of drill and practice courseware to have students practice using isolated skills.

■ **To supplement or replace worksheets and homework exercises.** Whenever students have difficulty with higher order tasks ranging from reading and writing to mathematics, teachers may have to stop and identify specific prerequisite skills that these students lack and provide the instruction and practice they need to go forward. In these cases, learning may require a rehearsal activity to make sure information is stored in long-term memory so students can retrieve it easily. Drills' motivation, immediate feedback, and self-pacing can make it more productive for students to practice required skills on the computer rather than on paper.

■ **In preparation for tests.** Despite the new emphasis on student portfolios and other authentic assessment measures, students can expect to take several kinds of objective examinations in their education careers. When they need to prepare to demonstrate mastery of specific skills in important examinations (e.g., for end-of-year grades or for college entrance), drill and practice courseware can help them focus on their deficiencies and correct them. An example integration strategy for drill functions is shown in Technology Integration Idea 4.2.

Guidelines for using drill and practice. Observe the following guidelines when designing integration strategies for drill and practice functions:

■ **Set time limits.** Teachers should limit the time devoted to drill assignments to 10 to 15 minutes per day. This ensures that students will not become bored and that the drill and practice strategy will retain its effectiveness. Also, teachers should be sure students have been introduced previously to the concepts underlying the drills; drill courseware should serve mainly to debug and to help students retain their grasp of familiar concepts.

■ **Assign individually.** Because self-pacing and personalized feedback are among the most powerful benefits of drills, these activities usually work best for individual computer use. However, some teachers with limited technology resources have found other, ingenious ways to capitalize on the motivational and immediate feedback capabilities of drills. If all students in a class benefit from practice in a skill using a drill program, the teacher may divide them into small groups to compete with each other for the best group scores. The class could even be divided into two groups for a "relay race" competition over which group can complete the assignment the fastest with the most correct answers.

■ **Use learning stations.** If not all students need the kind of practice that a drill provides, the teacher may make courseware one of several learning stations to serve students with identified weaknesses in one or more key skills. The key to using drill and practice appropriately is to match its inherent capabilities with the identified learning needs of individual students.

Tutorial Activities

Tutorials: Definition and Characteristics

Tutorial courseware uses the computer to deliver an entire instructional sequence similar to a teacher's classroom instruction on the topics. This instruction usually is expected to be complete enough to stand alone; the student should be able to learn the topic without any help or other materials from outside the courseware. Unlike other courseware activities, tutorials are true teaching courseware. Gagné et al. (1981) stated that good tutorial courseware should address all instructional events. (See the discussion of Gagné's Events of Instruction in Chapter 3.) Gagné et al. show how a tutorial may vary its strategies to accomplish events for different kinds of learning ranging from verbal information to complex applications of rules and problem solving.

TECHNOLOGY INTEGRATION IDEA 4.2

Integration of Drill and Practice Courseware

TITLE: Traffic Officer

CONTENT AREA/TOPIC: Language arts—letter cases

GRADE LEVEL: Elementary

NETS FOR STUDENTS: Standard 3

DESCRIPTION: Each student is given either an upper- or lowercase letter card. Students identify letters as a full group. One child is chosen as the traffic officer. Two areas are designated as parking lots, one for uppercase letters and one for lowercase letters. The traffic officer directs the other students' "cars" to the correct parking lot according to letter case. Students will be guided through this activity by the teacher. After letters are sorted ("cars are parked"), the class counts the number of letters in each group. The teacher introduces the class to the parking lot activity on the *Muppet Word Book* disk. At centers or workshops, individual students or small groups of students practice classifying letter cases with the *Muppet Word Book* parking lot activity and the traffic officer game. Students complete the traffic officer worksheet as an assessment activity.

Source: Cobb, L., Dunn, A., and Henry, G. (1991). Name games. *The Florida Technology in Education Quarterly, 3*(2), 63–66.

People may confuse drill activities with tutorial ones for two reasons. First, drill courseware may provide elaborate feedback that reviewers may mistake for tutorial explanations required by Gagné's events 4 and 5. Even courseware developers may claim that a package is a tutorial when it is, in fact, a drill activity with detailed feedback. Second, a good tutorial should include one or more practice sequences to address events 6 and 7, so reviewers easily become confused about the primary purpose of the package.

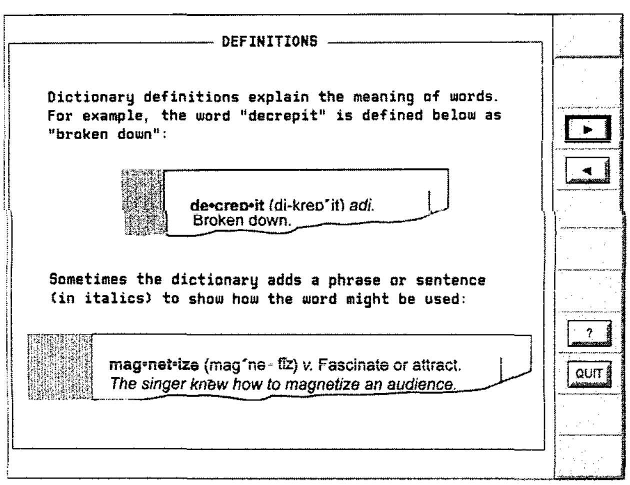
Tutorials often are categorized as linear and branching tutorials (Alessi & Trollip, 2001). A simple, linear tutorial gives the same instructional sequence of explanation, practice, and feedback to all learners regardless of differences in their performance. A more sophisticated, branching tutorial directs learners along alternate paths depending on how they respond to questions and whether or not they show mastery of certain parts of the material. Even branching tutorials can range in complexity by the amount of branching they allow and how fully they diagnose the kinds of instruction a student needs.

Some tutorials also have computer-management capabilities; teachers may "tell" such a program at what level to start for a student and get reports on each student's progress through the instruction. Although a tutorial program does not need these components, data collection and management features often make it more useful to teachers.

As the description of Events of Instruction implies, tutorials are most often geared toward learners who can read fairly well, usually older students or adults. Since tutorial instruction is expected to stand alone, it is difficult to explain or give appropriate guidance on-screen to a nonreader. However, some tutorials aimed at younger learners have found clever ways to explain and demonstrate concepts with graphics, succinct phrases or sentences, or audio directions coupled with screen devices.

Figure 4.3 Tutorial on Study Skills from *SkillsBank Study Skills*

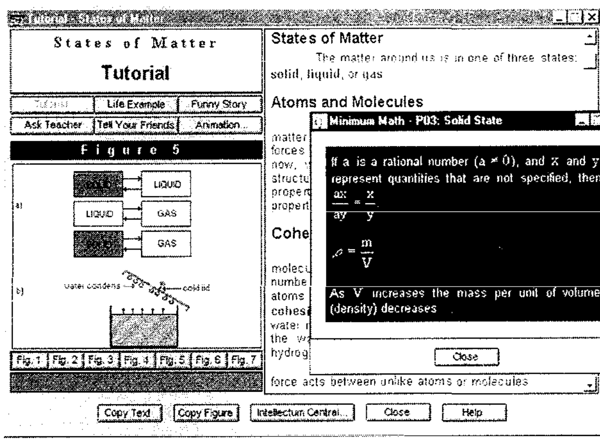
In this "finding definitions" study skill, the student sees a sequence of screens with explanations on how to locate and use dictionary definitions. After reading this description and seeing correct examples, the student is given practice items.



Source: *SkillsBank Study Skills* © 1996 SkillsBank Corporation. Used by permission.

Figure 4.4 Tutorial on States of Matter from Intellectum Plus's *PhysicaElementa: States of Matter*

The student sees a sequence of screens with explanations, descriptions, and animated examples of solids, liquids, gases, and how they can change states. After reading this description, students can view animated demonstrations of these principles.



Source: Used by permission of Intellectum Plus.

Some of the best tutorial courseware activities are in packages that accompany newly purchased computers or applications software, for example, *Introduction to Microsoft Works*. While tutorials are found more frequently on mainframe or file server systems than on microcomputers, some good tutorials are available on stand-alone systems. Examples of these tutorials are given in Figures 4.3 and 4.4.

Being a good teacher is a difficult assignment for any human, let alone a computer. However, courseware must accomplish this task to fulfill tutorial functions. In addition to meeting general criteria for good instructional courseware, well-designed tutorial programs should also meet several additional standards:

- **Extensive interactivity.** The most frequent criticism of tutorials is that they are "page-turners," that is, they ask students to do very little other than read. Good tutorials, like good teachers, should require students to give frequent and thoughtful responses to questions and problems and they should supply appropriate practice and feedback to guide students' learning.
- **Thorough user control.** *User control* refers to several aspects of the program. First, students should always be able to control the rate at which text appears on the screen. The program should not go on to the next information or activity screen until the user presses a key or gives some other indication of having completed the necessary reading. Next, the program should offer students the flexibility to review explanations, examples, or sequences of

instruction or move ahead to other instruction. The program should also provide frequent opportunities for students to exit as desired.

- **Appropriate and comprehensive teaching sequence.** The program's structure should provide a suggested or required sequence of instruction that builds on concepts and covers the content adequately. It should provide sufficient explanation and examples in both original and remedial sequences. In sum, it should compare favorably to an expert teacher's presentation sequence for the topic.
- **Adequate answer-judging and feedback capabilities.** Whenever possible, programs should allow students to answer in natural language and should accept all correct answers and possible variations of correct answers. They should also give appropriate corrective feedback when needed, supplying this feedback after only one or two tries rather than frustrating students by making them keep trying indefinitely to answer something they may not know.

Although some authors insist that graphics form part of tutorial instruction (Baek & Layne, 1988), others emphasize judicious use of graphics to avoid interfering with the purpose of the instruction (Eiser, 1988). Eiser is among those who recommend evaluation and record keeping on student performance as part of any tutorial.

Issues Related to Tutorials

Tutorials attract the same criticism as drill and practice for teacher-directed methods; that is, they deliver traditional instruction in skills rather than letting students create learning experiences through generative learning and development projects. Also, since good tutorials are difficult to design and program, critics charge that tutorials represent trivial or even counterproductive uses of the computer. A number of tutorials fail to meet criteria for good programs of this kind, thus contributing to this perception.

Tutorials are difficult to find, even for those who want to use them. Software publishers describe fewer packages as tutorials than any other kind of microcomputer courseware. Part of the reason for this comes from the difficulty and expense of designing and developing them. A well-designed tutorial sequence emerges from extensive research into how to teach the topic well, and its requirements for programming and graphics can become fairly involved. Designers must know what learning tasks the topic requires, what sequence students should follow, how best to explain and demonstrate essential concepts, common errors that students are likely to display, and how to provide instruction and feedback to correct those errors. Tutorials can be large, so they often work slowly on microcomputers. Larger tutorials must be delivered via integrated learning systems or other networked systems, making them expensive.

These problems become still more difficult because teachers frequently disagree about what they should teach for a given topic, how to teach it most effectively, and in what order to present learning tasks. A teacher may choose not to purchase a tutorial with a sound instructional sequence because it does not cover the topic the way he or she presents it. Not surprisingly, courseware companies tend to avoid programs that are problematic both to develop and market.

How to Use Tutorials in Teaching

Benefits of tutorial functions. It is unfortunate that microcomputer tutorials are so rare; a well-designed tutorial on a nontrivial topic can be a valuable instructional tool. Since a tutorial can include drill and practice routines, helpful features include the same ones as for drills (immediate feedback to learners and time savings) plus the additional benefit of self-contained, self-paced substitutes for teacher presentations.

Classroom applications of tutorial functions. Self-instructional tutorials should in no way threaten teachers, since few conceivable situations make a computer preferable to an expert teacher. However, the tutorial's unique capability of presenting an entire interactive instructional sequence can assist in several classroom situations:

- **Self-paced reviews of instruction.** On many occasions, students need repeated instruction on a topic after the teacher's initial presentation. Some students may be slower to understand concepts and need additional time on them. Others seem to learn better in a self-paced mode without the pressure to move at the same pace as the rest of the class. Still others may need review before a test. Teachers can help these students by providing tutorials at learning stations to review previously presented material while the teacher works with other students.
- **Alternative learning strategies.** Tutorials also provide alternative means of presenting material to support various learning strategies. Some students, typically advanced ones, prefer to structure their own learning activities and proceed on their own. A good tutorial allows students to glean much background material prior to meeting with a teacher or others to do assessment and/or further work assignments.
- **Instruction when teachers are unavailable.** Some students have problems when they surge ahead of their class rather than falling behind it. The teacher cannot leave the rest of the class to provide the instruction that such an advanced student needs. Many schools, especially those in rural areas, may not offer certain courses because they cannot justify the expense of hiring a teacher for comparatively few students who will need physics, German, trigonometry, or other lower demand courses. Well-designed tutorial courses, especially in combination with other methods such as distance learning, can help meet these students' needs.

Guidelines for using tutorials. Like drill and practice functions, tutorial functions are designed primarily to serve individuals. Depending on which of the above strategies it promotes, a tutorial may form a classroom learning station or may be available for checkout at any time in a library/media center. Many successful uses of tutorials have been documented over the years (Arnett, 2000; CAI in Music, 1994; Cann & Seale, 1999; Graham, 1994, 1998; Kraemer, 1990; Murray et al., 1988; Steinberg & Oberem, 2000), but microcomputer tutorials that fulfill the functions listed still are found but rarely in classroom use. Although they have considerable value and are popular in military and industrial training, schools and colleges have never fully tapped their potential as teaching resources. The expense of developing them and difficulty of marketing them may be to blame for this situation. However, recent trends toward combining tutorial courseware with video media and distance education may bring tutorial functions into more common use (see Technology Integration Idea 4.3).

Simulation Activities

Simulations: Definition and Characteristics

A simulation is a computerized model of a real or imagined system designed to teach how a system works. Unlike tutorial and drill and practice activities in which the structure is built into the package, learners usually must create their own sequence for using simulations. The person using the courseware usually chooses tasks and the order in which to do them. Alessi and Trollip (2001) identify two main types of simulations: those that teach about something and those that teach how to do something. They further divide the "about" simulations into physical and iterative types and they divide the "how to" simulations into procedural and situational types.

Physical simulations. Users manipulate objects or phenomena represented on the screen. For example, students see selections of chemicals with instructions to combine them to see the result or they may see how various electrical circuits operate. (See Figure 4.5.)

Iterative simulations. These speed up or slow down processes that usually either take so long or happen so quickly that students could not ordinarily see the events unfold. For example, courseware may show the effects of changes in demographic variables on population growth or the effects of environmental factors on ecosystems. Alessi and Trollip (2001) refer to this type as "iterative" because the student can run it over and over again with different values, observing the results each time. Biological simulations like those on genetics are popular, since they help students experiment with natural laws like the laws of genetics by pairing animals with given characteristics and showing the resulting offspring.

Procedural simulations. These activities teach the appropriate sequences of steps to perform certain procedures. They include diagnostic programs, in which students try to identify the sources of medical or mechanical problems, and flight simulators, in which students simulate piloting an airplane or other vehicle (e.g., *Microsoft Flight Simulator* shown in Chapter 2).

Situational simulations. These programs give students hypothetical problem situations and ask them to react. Some simulations allow for various successful strategies such as letting students play the stock market or operate businesses. Others have most desirable and least desirable options such as choices when encountering a potentially volatile classroom situation. (See Figure 4.6.)

4.3

Integration of Tutorial Courseware

TITLE: Interactive Algebra

CONTENT AREA/TOPIC: Mathematics/algebra

GRADE LEVEL: Middle to high school

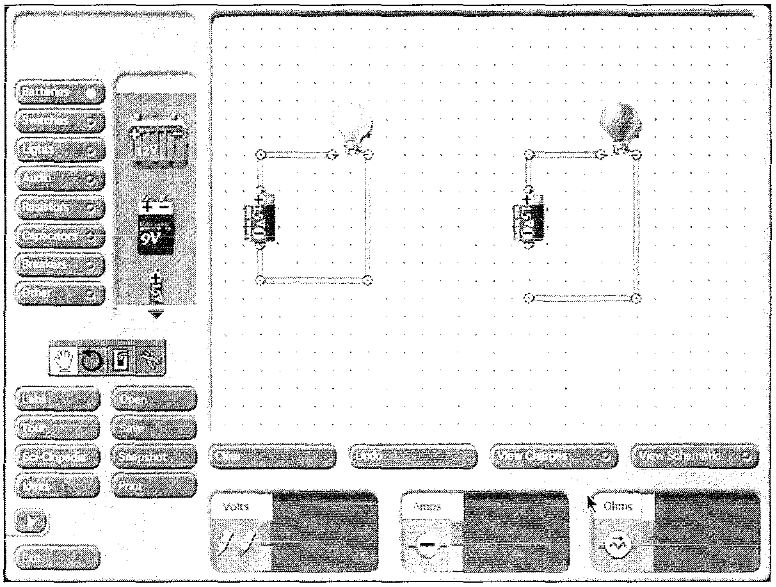
NETS FOR STUDENTS: Standards 3, 4

DESCRIPTION: Students can use this software as a review after the material has been introduced in class to help ensure that they understand the concepts and are prepared for taking end-of-year exams. Teachers introduce each of the algebra concepts in the usual way. However, instead of assigning worksheets or other practice to make sure students understand and remember concepts, the teacher assigns each student to go through the software tutorial sequence on the topic. Then the teacher holds a cooperative round-table review. This review is especially important for more complex concepts such as "What do the slope and x-y coordinates in a linear equation mean?" As students state one thing they remember about the topic, the teacher or one of the students enters the contribution into a computer connected to a projection system for all to see. Students continue adding information, asking questions, and giving comments about the topic, until the teacher feels that they have an adequate comprehension of the concepts.

Source: Based on ideas from the American Education Corporation's *Advanced Learning System Teacher Resource Guide* (1998).

Figure 4.5 Simulation on Electricity from Edmark's *Virtual Labs: Electricity*

This program provides images of batteries, switches, resistors, and elements necessary to create and test electrical circuits. The student selects elements, places them on a simulated board, and tests them. The program illustrates how they work when properly assembled.

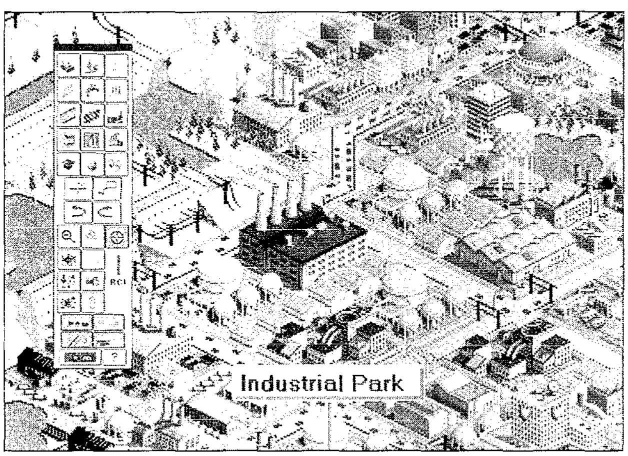


Source: Used by permission of Edmark Corporation.

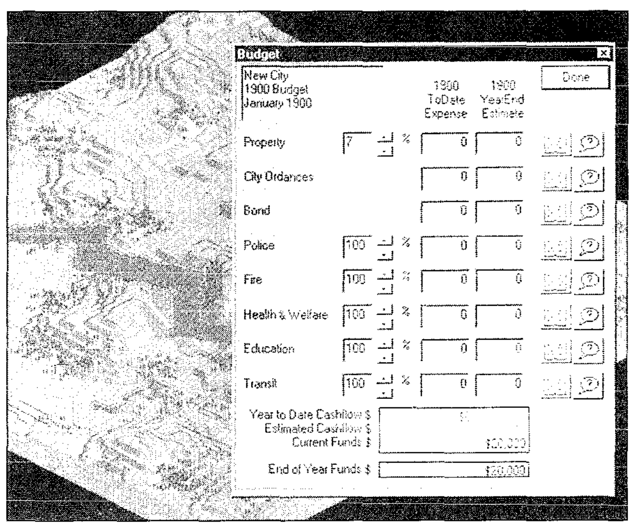
These types only clarify the various forms a simulation might take. *Teachers need not classify a given simulation into one of these categories.* They need to know only that all simulations show students what happens in given situations when they choose certain actions. Simulations usually emphasize learning about the system itself, rather than learning general problem-solving strategies. For example, a program called *The Factory* has students build products by selecting machines and placing them in the correct sequence. Since the program emphasizes solving problems in correct sequence rather than manufacturing in factories, it should probably be called a problem-solving activity rather than a

Figure 4.6 Simulation from MAXIS/Electronic Arts' *SimCity 2000*™

This popular simulation lets users build their own cities, create a budget for them, populate them, and run them, including responding to intermittent disasters.



Source: Images courtesy of Electronic Arts™.



simulation. Programs such as *SimCity* (MAXIS/Electronic Arts), which let students design their own cities, provide more accurate examples of building simulations (Adams, 1998).

Since simulations promote such widely varied purposes, it is difficult to provide specific criteria for selecting high-quality ones. By one frequently cited criterion, *fidelity,* a more realistic and accurate representation of a system makes a better simulation (Reigeluth & Schwartz, 1989). However, even this is not a criterion for judging all simulations (Alessi, 1988). Reigeluth and Schwartz describe some design concerns for simulations based on instructional theory. They list important simulation components including a scenario, a model, and an instructional overlay that lets learners interact with the program. Since the screen often presents no set sequence of steps, simulations—more than most courseware—need good accompanying documentation. A set of clear directions helps the teacher learn how to use the program and show the students how to use it rapidly and easily.

Issues Related to Simulations

Most educators acknowledge the instructional usefulness of simulations; however, some are concerned about the accuracy of the programs' models. For example, when students see simplified versions of these systems in a controlled situation, they may get inaccurate or imprecise perspectives on the systems' complexity. Students may feel they know all about how to react to situations because they have experienced simulated versions of them. Many educators feel especially strongly that situational simulations must be followed at some point by real experiences. Many teachers of very young children feel that learners at early stages of their cognitive development should experience things first with their five senses rather than on computer screens.

Some simulations are viewed as complicated ways to teach very simple concepts that could just as easily be demonstrated on paper, with manipulatives, or with real objects. For example, students usually are delighted with the simulation of the food chain called *Odell Lake,* a program that lets students see what animals prey on what other animals in a hypothetical lake. However, some wonder whether or not such a computer simulation is necessary or even desirable to teach this concept. Hasselbring and Goin (1993) point out that students can often master the activities of a simulation without actually developing effective problem-solving skills; on the contrary, such applications actually can encourage counterproductive behaviors. For example, some simulations initially provide little information with which to solve problems, and students are reduced to "trial-and-error guessing rather than systematic analysis of available information" (p. 156). Teachers must carefully structure integration strategies so that students will not use simulations in inappropriate ways.

Simulations are considered among the most potentially powerful computer courseware resources; however, as with most courseware, their usefulness depends largely on the program's purpose and how well it fits in with the purpose of the lesson and student needs. Teachers are responsible for recognizing the unique instructional value of each simulation and using it to best advantage.

How to Use Simulations in Teaching

Benefits of simulation functions. Simulations have long been recognized for their unique teaching capabilities. Depending on the topic, a simulation can provide one or more of the following benefits (Alessi & Trollip, 2001):

- **Compress time.** This feature is important whenever students study the growth or development of living things (e.g., pairing animals to observe the characteristics of their offspring) or other processes that take a long time (e.g., the movement of a glacier). A simulation can make something happen in seconds that normally takes days, months, or longer. Consequently, feedback is faster than in real life and students can cover more variations of the activity in a shorter time.
- **Slow down processes.** Conversely, a simulation can also model processes normally invisible to the human eye because they happen so quickly. For example, physical education students can study the slowed-down movement of muscles and limbs as a simulated athlete throws a ball or swings a golf club.
- **Get students involved.** Simulations can capture students' attention by placing them in charge of things and asking that most motivating of questions: "What would *you* do?" The results of their choices can be immediate and graphic. It also allows users to interact with the program instead of just seeing its output.
- **Make experimentation safe.** Whenever learning involves physical danger, simulations are the strategy of choice. This is true any time students are learning to drive vehicles, handle volatile substances, or react to potentially dangerous situations. They can experiment with strategies in simulated environments that might result in personal injury to themselves or others in real life.
- **Make the impossible possible.** This is the most powerful feature of a simulation. Very often, teachers simply cannot give students access to the resources or the situations that simulations can. Simulations can show students what it would be like to walk on the moon or to react to emergencies in a nuclear power plant. They can see cells mutating or hold countrywide elections. They can even design new societies or planets and see the results of their choices.
- **Save money and other resources.** Many school systems are finding dissections of animals on a computer screen much less expensive than on real frogs or cats and just as instructional. (It also is easier on the animals!) Depending on the subject, a simulated experiment may be just as effective as a learning experience but at a fraction of the cost.

■ **Repeat with variations.** Unlike real life, simulations let students repeat events as many times as they wish and with unlimited variations. They can pair any number of cats or make endless airplane landings in a variety of conditions to compare the results of each set of choices.

■ **Make situations controllable.** Real-life situations often are confusing, especially to those seeing them for the first time. When many things happen at once, students have difficulty focusing on the operation of individual components. Who could understand the operation of a stock market by looking at the real thing without some introduction? Simulations can isolate parts of activities and control the background noise. This makes it easier for students to see what is happening later when all the parts come together in the actual activity.

Classroom applications of simulation functions. Real systems are usually preferable to simulations, but a simulation can suffice when a teacher considers the real situation too time consuming, dangerous, expensive, or unrealistic for a classroom presentation. Simulations should be considered in the following situations, keeping in mind that the real activity is preferable:

■ **In place of or as supplements to lab experiments.** When adequate lab materials are not available, teachers should try to locate computer simulations of the required experiments. Many teachers find that simulations offer effective supplements to real labs, either to prepare students for making good use of the actual labs, or as follow-ups with variations on the original experiments without using up consumable materials. Some simulations actually allow users to perform experiments that they could not otherwise manage or that would be too dangerous for students (see Technology Integration Idea 4.4).

■ **In place of or as supplements to role playing.** When students take on the roles of characters in situations, computer simulations can spark students' imaginations and interests in the activities. However, many students either refuse to role play in front of a class or get too enthusiastic and disrupt the classroom. Computerized simulation can take the personal embarrassment and logistical problems out of the learning experience and make classroom role playing more controllable.

■ **In place of or as supplements to field trips.** Seeing an activity in the real setting can be a valuable experience, especially for young children. Sometimes, however, desired locations are not within reach of the school and a simulated experience of all or part of the process is the next best thing. As with labs, simulations provide good introductions or follow-ups to field trips.

■ **Introducing a new topic.** Courseware that allows students to explore the elements of an environment in a hands-on manner frequently provides students' first in-depth contact with a topic. This seems to accomplish several purposes. First, it is a nonthreatening way to introduce new terms and unfamiliar settings. Students know that they are not being graded, so they feel less pressure than usual to learn everything right away. A simulation can become simply a get-acquainted look at a topic. Simulations can also build students' initial interest in a topic. Highly graphic, hands-on activities draw them into the topic and whet their appetite to learn more. Finally, some software helps students see how certain prerequisite skills relate to the topic; this may motivate students more strongly to learn the skills than if the skills were introduced in isolation from the problems to which they apply. An example of this is *Decisions! Decisions!* software by Tom Snyder on social studies topics such as the U.S. Constitution and elections.

■ **Fostering exploration and process learning.** Teachers often use content-free simulation/problem-solving software as motivation for students to explore their own cognitive processes. Since this kind of courseware

TECHNOLOGY

4.4

Directed Integration of Simulation Courseware

TITLE: Earthquake!

CONTENT AREA/TOPIC: Earth science concepts

GRADE LEVEL: Elementary

NETS FOR STUDENTS: Standards 3, 5, 6

DESCRIPTION: Students learn to work cooperatively with a group to perform an experiment to explain the movement of an earthquake. The teacher begins with a basic review of an earthquake and its causes and discusses tectonic plates and the role they play in the earth's crust. Students build a simulated building, start up the software, and set up the seismoscope according to directions in the instruction manual. They simulate earthquakes by shaking the table as the software collects data. Students review data gathered and discuss how they show that an earthquake has occurred.

Source: Smith, K. (1992). Earthquake! *The Florida Technology in Education Quarterly, 4*(2), 68–70.

requires students to learn no specific content, it is easier to get them to concentrate on problem-solving steps and strategies. However, with content-free products, it is even more important than usual that teachers draw comparisons between skills from the courseware activities and those in the content areas to which they want to transfer the experience. For example, *The Incredible Laboratory* (Sunburst) presents an implicit emphasis on science process skills that the teacher may want to point out. These kinds of activities may be introduced at any time, but it seems more fruitful to use them just prior to content area activities that will require the same processes.

- **Encouraging cooperation and group work.** Sometimes a simulated demonstration can capture students' attention quickly and effectively and interest them in working together on a product. For example, a simulation on immigration or colonization might be the "grabber" a teacher needs to launch a group project in a social studies unit (see Technology Integration Idea 4.5).

Guidelines for using simulation functions. Simulations offer more versatile implementation than tutorials or drills. They usually work equally effectively with a whole class, small groups, or individuals. A teacher may choose to introduce a lesson to the class by displaying a simulation or to divide the class into small groups and let each solve problems. Because they instigate discussion and cooperative work so well, simulations usually are considered more appropriate for pairs and small groups than for individuals. However, individual use certainly is not precluded.

The market offers many simulations, but it often is difficult to locate one on a desired topic. The field of science seems to include more simulations than any other area (Andaloro, 1991; Mintz, 1993; Richards, 1992; Ronen, 1992; Simmons & Lunetta, 1993; Smith, 1992), but use of simulations is also popular in social sciences topics (Adams, 1998; Allen, 1993; Clinton, 1991; Estes, 1994). However, more simulations currently are in development and feature videodisc and online supplements to combine the control, safety, and interactive features of computer simulations with the visual impact of pictures of real-life devices and processes.

Instructional Games

Instructional Games: Definition and Characteristics

Instructional games are courseware whose function is to increase motivation by adding game rules to learning activities. Even though teachers often use them in the same way as drill and practice or simulation courseware, games usually are listed as a separate courseware activity because their instructional connotation to students is slightly different. When students know they will play a game, they expect a fun and entertaining activity because of the challenge of the competition and the potential for winning (Randel, Morris, Wetzel, & Whitehill, 1992). Naturally, classroom instruction should not consist entirely of these kinds of activities, no matter how instructional or motivational they are. Teachers intersperse games with other activities to hold attention or to give rewards for accomplishing other activities.

As with simulations, the categories described here merely illustrate the various forms an instructional game may take. Teachers should not feel that they have to classify specific games into categories. But it is important to recognize the common characteristics that set instructional games apart from other types of courseware: game rules,

TECHNOLOGY INTEGRATION IDEA 4.5

Constructivist Integration of Simulation Courseware

TITLE: Community Planning Projects with *SimCity*

CONTENT AREA/TOPIC: Social studies—citizenship/group cooperation in social projects

GRADE LEVEL: 7

NETS FOR STUDENTS: 3, 5, 6

DESCRIPTION: This lesson helps raise students' awareness of responsibility to become informed citizens, shows how they can participate in local decision making, and gives practice in working cooperatively with a group to carry out a social project. The software allows the development of a comprehensive plan for the township. Have a representative from the county planning and zoning office talk with the students about factors of concern to community development. The teacher then introduces students to the *SimCity* software. They form teams of four to five students each and start their own community planning projects with *SimCity*. Each group meets every week to discuss and develop its plan. As a group, they decide how to select and place features such as roads, homes, and utilities. After recording their decisions on paper, they enter them into the program and observe the results. They discuss feedback that the program gives them on areas such as taxes, crime rates, and public opinion. After their plans are complete, each group presents its plan to the teacher and explains and defends their choices.

Source: Jacobson, P. (1992). Save the cities! *SimCity* in grades 2–5. *The Computing Teacher, 20*(2), 14–15.

elements of competition or challenge, and amusing or entertaining formats. These elements generate a set of mental and emotional expectations in students that make game-based instructional activities different from nongame ones.

Since instructional games often amount to drills or simulations overlaid with game rules (see Figure 4.7), the same criteria, such as better reinforcement for correct answers than for incorrect ones, should apply to most games. When Malone (1980) examined the evidence on what makes things fun to learn, he found that the most popular games included elements of adventure and uncertainty and levels of complexity matched to learners' abilities. However, teachers should examine instructional games carefully for their value as both educational and motivational tools. Teachers should also assess the amount of physical dexterity that games require of students and make sure that students will not be frustrated instead of motivated by the activities. Games that call for violence or combat need careful screening, not only to avoid parent criticism, but also because girls often perceive the attraction of these activities differently than boys and because such games sometimes depict females as targets of violence.

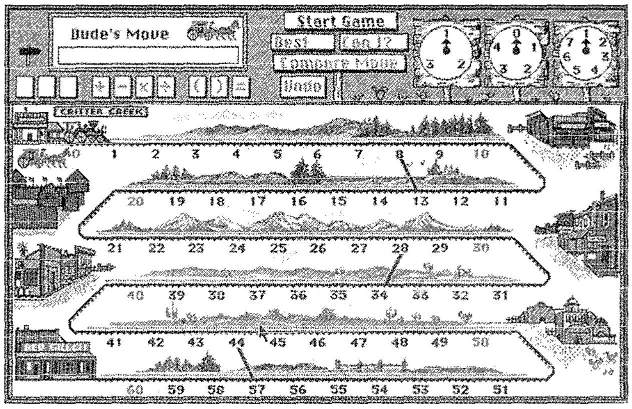
Issues Related to Instructional Games

A classroom without elements of games and fun would be a dry, barren landscape for students to traverse. In their review of the effectiveness of games for educational purposes, Randel et al. (1992) found "[the fact] that games are more interesting than traditional instruction is both a basic for using them as well as a consistent finding" (p. 270). They also observed that retention over time favors the use of simulations/games. Yet many educators believe that games, especially computer-based ones, are overused and misused (McGinley, 1991). Other teachers believe games convince students that they are escaping from learning, and that they draw attention away from the intrinsic value and motivation of learning. Critics also feel that winning the game becomes a student's primary focus and the instructional purpose is lost in the pursuit of this goal. Observers disagree about whether getting lost in the game is a benefit or a problem.

Some teachers believe that any time they can sneak learning in under the guise of a game, it is altogether a good thing (McGinley, 1991). Other teachers believe that students can become confused about which part of the activity is the game and which part is the skill; they may then have difficulty transferring their skill to later nongame situations. For example, the teacher's manual for Sunburst's *How the West Was One + Three × Four* reminds teachers that some students can confuse the math operations rules with the game rules and that teachers must help them recognize the need to focus on math rules and use them outside the game. Recent studies seem to indicate that instructional games can be useful in fostering higher order skills, but that their usefulness hinges on how teachers

Figure 4.7 Instructional Game from Sunburst Communications' *How the West Was One + Three × Four*

The program provides practice in math order of operations. The student tries to move a vehicle along steps on a trail by answering math problems that combine numbers using math order of operations.



Source: Used by permission of Sunburst Communications.

employ them (Henderson, Klemes, & Eshet, 2000; Rieber, Smith, & Noah, 1998).

Although students obviously find many computer games exciting and stimulating, educational value sometimes is difficult to pinpoint. Teachers must try to balance the motivation that instructional games bring to learning against the classroom time they take away from nongame strategies. For example, students may become immersed in the challenge of the *Carmen Sandiego* series, but more efficient ways to teach geography may be just as motivating. Successful uses of games have been reported in many content areas (Flowers, 1993; Muckerheide, Mogill, & Mogill, 1999; Trotter, 1991).

How to Use Instructional Games in Teaching

Several kinds of instructional opportunities invite teachers to take advantage of the motivational qualities of games:

- **In place of worksheets and exercises.** This role resembles that of drill and practice (see Technology Integration Idea 4.6).
- **To foster cooperation and group work.** Like simulations, many instructional games serve as the basis for or introductions to group work. A game's interactive and motivational qualities help interest students in the topic and present opportunities for competition among groups.
- **As a reward.** Perhaps the most common use of games is to reward good work. This is a valid role for instructional courseware, but teachers should avoid overuse of it. Otherwise, the game can lose its motivational value and become an "electronic babysitter." Some schools actually bar games from classrooms for fear that they will overemphasize the need for students to be entertained.

TECHNOLOGY INTEGRATION IDEA 4.6

Integration of Instructional Game Courseware: Practicing Math Rules of Operation

TITLE: Please! Please! Remember My Dear Aunt Sally!

CONTENT AREA/TOPIC: Mathematics—order of mathematics operations

GRADE LEVEL: Middle school

NETS FOR STUDENTS: Standards 1, 3

DESCRIPTION: Before beginning instruction, use worksheets to ensure that all students know basic math operations and symbols (+, −, ×, ÷); reteach if necessary. Demonstrate the game to the students and let them play it as a group activity. Make sure the students know that the order of operations is a mathematical rule that always applies and not a game rule applicable only to this program. Explain the mnemonic "Please, Please Remember My Dear Aunt Sally" and the meaning of the order of operations. Worksheets help students practice and remember these rules; the teacher checks and assists with their work. When all students seem able to work the problems, demonstrate *HTWWO* and review the rules. Students practice one game as a group. Make arrangements for students to practice their skills on the game in the classroom or computer lab for a large group activity. Students help each other play the game against the computer. Give special recognition to students who win the most games in a period.

Source: Culpeper, G., Myers, E., and Roblyer, M. D. (1991). Please, please remember my dear Aunt Sally! *The Florida Technology in Education Quarterly, 3*(2), 87–88.

Problem-Solving Courseware

Problem-Solving Courseware: Definition and Characteristics

Teachers may find the topic of problem solving both alluring and perplexing. No goal in education seems more important today than making students good problem solvers, yet no area is as ill defined and difficult to understand. Even scientists have difficulty defining problem solving. Funkhouser and Dennis (1992) quoted an earlier author as saying that "Problem solving [means] the behaviors that researchers who say they are studying problem solving, study" (p. 338). Sherman (1987–1988) was somewhat more specific, claiming that all problem solving involves three components: recognition of a goal (an opportunity for solving a problem), a process (a sequence of physical activities or operations), and mental activity (cognitive operations to pursue a solution). Sherman said that problem solving is a relatively sophisticated mental ability that is difficult to learn and that it is highly idiosyncratic. That is, problem-solving ability depends on "knowledge, prior experience, and motivation, and many other attributes" (p. 8).

This definition of problem solving covers a wide variety of desired component behaviors. The literature mentions such varied subskills for problem solving as metacognition, observing, recalling information, sequencing, analyzing, finding and organizing information, inferring, predicting outcomes, making analogies, and formulating ideas. Since even the definition of problem solving inspires ongoing controversy in education, it is not surprising that opinions differ dramatically about the proper role of courseware and other technology products in helping to foster this important capability. The positions lean toward two general ways in which teachers can view problem solving. Which of these views a teacher uses will determine the strategy for teaching problem solving and the application of related technology resources.

Two views on fostering problem solving. Some teachers view problem solving as a high-level skill that can be taught directly, at least in part, by specific instruction and practice in its component strategies and subskills. Others suggest placing students in problem-solving environments and, with some coaching and guidance, letting them develop their own heuristics for attacking and solving problems. Although the purposes of the two views overlap somewhat, one is directed more toward supplying prerequisite skills for specific kinds of problem solving. The other view aims more toward motivating students to attack problems and to recognize solving problems as an integral part of everyday life. Blosser (1988) confirms this dichotomy, saying that "Problem solving includes . . . an attitude or predisposition toward inquiry as well as the actual processes by which individuals . . . gain knowledge." Students need to combine these two elements; teachers must make ongoing adjustments to the amount of time they spend on each kind of approach in each of several content areas.

Two types of problem-solving courseware for directed instruction. Two distinct types of courseware purport to teach problem-solving skills. One is specific to teaching content area skills, primarily in mathematics. (For example, *The Geometric Supposer* by Sunburst encourages students to learn strategies for solving geometry problems by drawing and manipulating geometric figures.) The other type of problem-solving software focuses on general, content-free skills such as recalling facts, breaking a problem into a sequence of steps, or predicting outcomes. For example, Sunburst's *Memory Castle* is designed to help students remember instructions and follow directions.

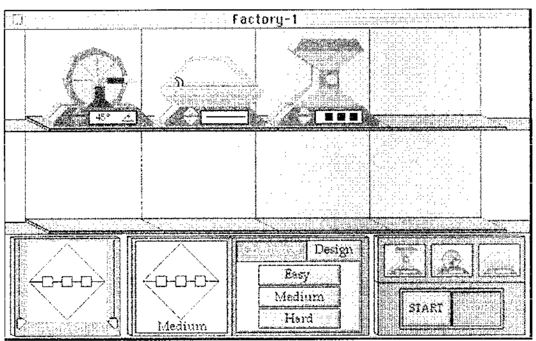
Most courseware is specifically designed to focus on one of these two approaches; however, some authors point out that programs can help teach problem solving without being specifically designed to do so (Gore, 1987–1988). Courseware implements numerous approaches to teach each of these kinds of skills. Some use challenge strategies (*The King's Rule* by Sunburst); others use puzzle games (*Safari Search* by Sunburst), adventure games (*Carmen Sandiego* by The Learning Company; *My Make Believe Castle* by Logo Computer Systems), or simulation approaches (*The Factory* by Sunburst). Still others are what might be called problem-solving "environments." These more complex, multifaceted packages offer a variety of tools to allow students to create solutions to problems presented by video scenario (*Alien Rescue*, University of Texas). See Figures 4.8 through 4.11 for examples.

Issues Related to Problem-Solving Courseware

Names versus skills. As mentioned earlier, courseware packages use many terms to describe problem solving and

Figure 4.8 Problem Solving of Sequence Skills from Sunburst Communications' *The Factory*

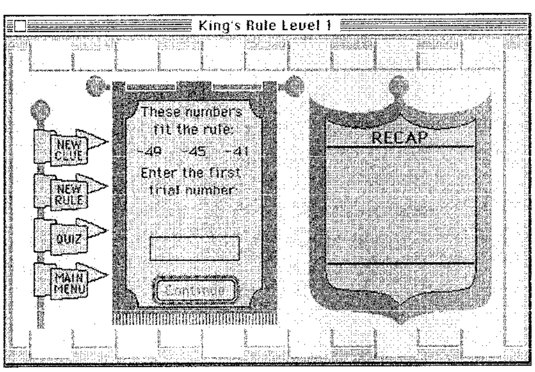
By selecting a sequence of machines to create a given product, students learn that problems must be analyzed and solved in a certain order to achieve desired results.



Source: Used by permission of Sunburst Communications.

Figure 4.9 Problem Solving of Confirmation Bias from Sunburst Communications' *The King's Rule*

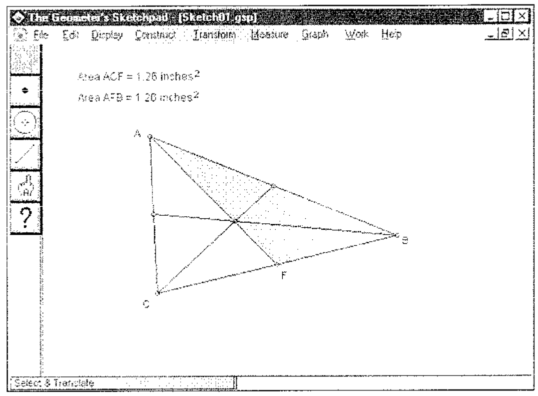
Students are given a number pattern (e.g., 5, 10, 15, 20) and must determine the rule that results in the pattern by entering other number sequences that also follow the rule. This program helps address the problem of "confirmation bias" that results when students fail to gather sufficient evidence before giving an answer.



Source: Used by permission of Sunburst Communications.

Figure 4.10 Problem Solving with Geometry from *The Geometer's Sketchpad*

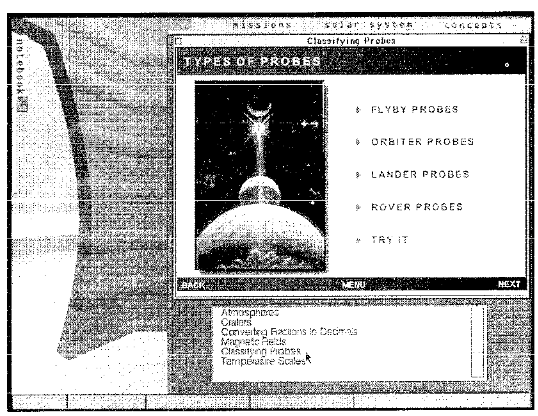
This program is a dynamic construction and exploration tool. Students construct an object and explore its mathematical properties by dragging the object with the mouse. They first visualize and analyze a problem, and then make conjectures before attempting a proof.



Source: *The Geometer's Sketchpad*®, Key Curriculum Press, 1150 65th Street, Emeryville, CA 94608, 1-800-995-MATH.

Figure 4.11 Problem-Solving Environment from *Alien Rescue*

This program shows a video challenging students to find homes for each of several displaced aliens circling Earth in a spaceship. The software offers a variety of tools to let students research planets and moons in our galaxy.



Source: Used by permission of The Alien Rescue Team 2001–2002.

their exact meanings are not always clear. Terms that appear in courseware catalogs as synonyms for problem solving include thinking skills, critical thinking, higher level thinking, higher order cognitive outcomes, reasoning, use of logic, and decision making. In light of this diversity of language, teachers can identify the skills that a courseware package addresses by looking at its activities. For example, a courseware package may claim to teach inference skills. One would have to see how it defines *inference* by examining the tasks it presents, which may range from determining the next number in a sequence to using visual clues to predict a pattern.

Courseware claims versus effectiveness. It would be difficult to find a courseware catalog that did not claim that its products foster problem solving. However, few publishers of courseware packages that purport to teach specific problem-solving skills have data to support their claims. When students play a game that requires skills related to problem solving, they do not necessarily learn these skills. They may enjoy the game thoroughly and even be successful at it without learning any of the intended skills. Teachers may have to do their own field testing to confirm that courseware is achieving the results they want.

Possible harmful effects of directed instruction. Some researchers believe that direct attempts to teach problem-solving strategies actually can be counterproductive for some students. Mayes (1992) reports on studies that found "teaching-sequenced planning to solve problems to high ability learners could interfere with their own effective processing" (p. 243). In a review of research on problem solving in science, Blosser (1988) also found indications that problem-solving instruction may not have the desired results if the instructional strategy does not suit certain kinds of students. For example, students with high math anxiety and low visual preference or proportional reasoning abilities will profit from instruction in problem solving only if it employs visual approaches.

The problem of transfer. Although some educators feel that general problem-solving skills such as inference and pattern recognition will transfer to content-area skills, scant evidence supports this view. In the 1970s and 1980s, for example, many schools taught programming in mathematics classes under the hypothesis that the planning and sequencing skills required for programming would transfer to problem-solving skills in math. Research results never supported this hypothesis. In general, research tends to show that skill in one kind of problem solving will transfer primarily to similar kinds of problems that use the same solution strategies. Researchers have identified nothing like "general thinking skills," except in relation to intelligence (IQ) variables.

How to Use Problem-Solving Courseware in Teaching

Benefits, applications, and guidelines for using directed strategies with problem-solving courseware. Integration of courseware into direct teaching of problem-solving skills places even more responsibility than usual on teachers. Usually, teachers want to teach clearly defined skills. To teach problem solving, they must decide which particular kind of problem-solving ability students need to acquire and how best to foster it. For example, Stokes (1999) recommends that students use a teacher-designed reflection sheet and keep a log of problem-solving strategies and outcomes. With clearly identified skills and a definite teaching strategy, problem-solving courseware has unique abilities to help focus students' attention on required activities. This kind of courseware can get students to apply and practice desired behaviors specific to a content area or more general abilities in problem solving. These six steps can help teachers to integrate courseware for these purposes:

1. Identify problem-solving skills or general capabilities to build or foster skills in:
   a. solving one or more kinds of content-area problems (building algebra equations);
   b. using a scientific approach to problem solving (identifying the problem, posing hypotheses, planning a systematic approach); and
   c. components of problem solving such as following a sequence of steps or recalling facts.
2. Decide on an activity or a series of activities that would help teach the desired skills (see Technology Integration Idea 4.7).
3. Examine courseware to locate materials that closely match the desired abilities, remembering to not judge capabilities on the basis of vendor claims alone.
4. Determine where the courseware fits into the teaching sequence (for example, to introduce the skill and gain attention or as a practice activity after demonstrating problem solving or both).
5. Demonstrate the courseware and the steps to follow in solving problems.
6. Build in transfer activities and make students aware of the skills they are using in the courseware.

Benefits, applications, and guidelines for using constructivist strategies with problem-solving courseware. Like many technology resources, some software with problem-solving functions can be employed in directed ways, but are designed for implementation using more constructivist models. These models give students no direct training in or introduction to solving problems; rather they place students in highly motivational problem-solving environ-

TECHNOLOGY INTEGRATION IDEA 4.7

Problem-Solving Courseware Activity: Addressing Confirmation Bias When Determining Number Rules in Mathematics

TITLE: Are You Sure Your Answer Is Correct?

CONTENT AREA/TOPIC: Mathematics—testing hypotheses in problem-solving strategies

GRADE LEVEL: Middle school

NETS FOR STUDENTS: Standards 3, 6

DESCRIPTION: This activity draws students' attention to the problem of confirmation bias in problem solving. Students play the Hidden Rule Game, in which the program generates three numbers and students have to give their own sets of three numbers that follow the same rule. The program tells them whether or not their sets follow the rule. As students consider the numbers, the teacher must ask questions to help them generate evidence to support their positions. For example, suppose the numbers are 16, 18, 20. Students must be encouraged to gather confirming evidence and submit their rules carefully. They may jump to the conclusion that the rule is "Jumps of two," when it may be "all even numbers" or "ascending numbers."

Source: Johnson, J. (1987). Do you think you might be wrong? Confirmation bias in problem solving. *Arithmetic Teacher, 34*(9), 13–16.

ments and encourage them to work in groups to solve problems. Bearden and Martin (1998) describe such a strategy using a problem-solving software combined with a listserv for students to share their results. (Also see Martin and Bearden, 1998.)

Constructivists believe this kind of experience helps students in three ways. First, they expect that students will be more likely to acquire and practice content-area, research, and study skills for problems they find interesting and motivating. For example, to succeed in the *Carmen Sandiego* software series, students must acquire both some geography knowledge and some ability to use reference materials that accompany the package. Also, they must combine this learning with deductive skills to attack and solve detective-type problems (Robinson & Schonborn, 1991).

Second, constructivists claim that this kind of activity helps keep knowledge and skills from becoming inert because it gives students opportunities to see how information applies to actual problems. They learn the knowledge and its application at the same time. Finally, students gain opportunities to discover concepts themselves, which they frequently find more motivating than being told or, as constructivists might say, *programmed* with the information (McCoy, 1990).

Guidelines for using problem-solving software. Seven steps help teachers integrate problem-solving courseware according to constructivist models:

1. Allow students sufficient time to explore and interact with the software, but provide some structure in the form of directions, goals, a work schedule, and organized times for sharing and discussing results.
2. Vary amount of direction and assistance, depending on each student's needs.
3. Promote a reflective learning environment; let students talk about their work and the methods they use.
4. Stress thinking processes rather than correct answers.
5. Point out the relationship of courseware skills and activities to other kinds of problem solving.
6. Let students work together in pairs or small groups.
7. For assessments, use alternatives to traditional paper-and-pencil tests.

Problem-solving and simulation activities work so similarly in constructivist models that it usually is difficult to differentiate between them. Integration strategies for either usually are the same.

Integrated Learning Systems

Integrated Learning Systems: Definition and Characteristics

Integrated learning systems (ILSs) are the most powerful—and the most expensive—of available courseware products, primarily because they are more than just courseware and because they require more than one computer to run them. From the time they were introduced in the early 1970s until recently, ILSs were computer networks: a combination of instruction and management systems that ran on terminals or microcomputers connected to a larger computer. However, a new way of providing ILS-type capability is offering the curriculum online via the Internet, rather than through a local network.

Regardless of the delivery system, an ILS is characterized by a "one-stop shopping" approach to providing courseware, what Brush (1998) refers to as a "turnkey implementation process for integrating computer-based education into a curriculum" (p. 7). Each ILS offers a variety of instructional techniques in one place, usually as a package complete with technical maintenance and teacher training. They present strengths like prepared curricula and ease of use so that school personnel need not know a great deal about technology to use them. Consequently, they usually simplify integration decisions by defining schoolwide curriculum rather than individual lessons.

In addition to providing a combination of drill and practice, tutorial, simulation, problem-solving, and tool courseware, an ILS is capable of maintaining detailed records on individual student assignments and performance data and supplying printouts of this information to teachers. Bailey and Lumley (1991, p. 21) include the following as characteristics of an ILS:

- Instructional objectives specified, with each lesson tied to those objectives
- Lessons integrated into the standard curriculum
- Courseware that spans several grade levels in comprehensive fashion
- Management systems that collect and record results of student performance.

ILS courseware and management software are housed on a central computer or *server,* which students may access via a local network or the Internet. As each student signs onto a microcomputer connected to the server or Internet, the file server sends (or downloads) student assignments and courseware to the station and proceeds to keep records on what the students do during time spent on the system. The teacher makes initial assignments for work on the system, monitors student progress by reviewing ILS printouts, and provides additional instruction or support where needed.

The first ILSs on the market in the 1970s were primarily drill and practice delivery systems designed to improve student performance on the isolated skills measured by standardized tests. These self-contained, mainframe-based systems predated the microcomputer era, and they did not run any software besides their own. Usually housed in labs, they were designed for use in pull-out programs to supplement teachers' classroom activities, that is, students were pulled out of classrooms daily or weekly and sent to ILS labs for remedial or reinforcement work. However, these systems have evolved into multipurpose products that can run software and courseware other than their own; they can now provide a variety of instructional support from enrichment to complete curriculum. As with other media such as videodiscs, school districts view ILSs as alternatives to traditional classroom materials such as textbooks. Brush reported that as of 1998, estimates were that between 11% and 25% of all U.S. schools owned ILSs.

The courseware component of an ILS. Instructional activities available on an ILS range from simple drill and practice to extensive tutorials. Many ILSs are moving toward complete tutorial systems intended to replace teachers in delivering entire instructional sequences. An ILS usually includes instruction on the entire scope and sequence of skills in a given content area; for example, it may cover all discrete mathematics skills typically presented in grades 1 through 6.

The management system component of an ILS. The capability that differentiates ILSs from other networked systems is the emphasis on individualized instruction tied to records of student progress. A typical ILS gives teachers progress reports across groups of students as well as the following kinds of information on individual performance:

- Lessons and tests completed
- Questions missed on each lesson by numbers and percentages
- Numbers of correct and incorrect tries
- Time spent on each lesson and test
- Pretest and post-test data.

Issues Related to ILSs

The costs of ILSs. The primary criticism of ILSs centers on their expense compared to their impact on improving learning. Bentley (1991) warned that "The search for less expensive alternatives to the ILS is only logical considering how difficult it would be to find options that are *more* expensive" (p. 25). ILS proponents, on the other hand, feel that the students who experience the most success with ILSs are those whose needs are typically most difficult to meet (Bender, 1991; Bracy, 1992; Shore & Johnson, 1992). ILS proponents say there is value in any system that can help potential dropouts stay in school or help remedy the deficiencies of students with learning disabilities. They point to studies and personal testimony from teachers over the years that attest to the motivational qualities of allowing students to work at their own pace and experience success each time they work on the system.

Research on ILS impact. When Becker (1992) reported his summary of some 30 studies of ILS effectiveness, he found widely varied results with various implementation methods and systems. Students generally tend to do somewhat better with ILSs than with other methods, and results were sometimes substantially superior to non-ILS methods. But Becker found no predictable pattern for successful and unsuccessful ILSs. He concluded that data were not sufficient either to support or oppose the purchase of an ILS in a given school or district.

A subsequent large-scale study of ILS use in Indiana reported by Estep, McInerney, and Vockell (1999–2000) found no differences on statewide achievement test scores between schools who did and did not use an ILS. An extensive study of ILS use in New York City schools reported varying results (Miller, 1997). However, Brush, Armstrong, and Barbrow (1999) found that two different resources offered in the same ILS had different impacts on achievement. Individualized software designed to provide foundations instruction had less impact than software that could be selected by teachers to supplement their own instruction. In summary, it seems to be as Van Dusen and Worthen (1995) observed: The impact of ILS varies greatly with implementation methods.

Concerns about the role of ILSs. In a follow-up to his literature review on ILS uses, Becker (1994) criticized uses of ILSs that encourage "mindless adherence to the principle of individualized instruction" (p. 78). Brush (1998) agreed with Becker, finding that ". . . lack of teacher involvement (in ILS use) has led to improper coordination between classroom-based and computer-based instructional activities . . . and lack of teacher understanding regarding effective strategies and procedures for using ILSs" (p. 7).

An early concern expressed by many educators (White, 1992) that the cost of ILSs combined with the comprehensive nature of their curricula might cause schools to view them as replacements for teachers has not yet proven to be a real problem. However, the fear expressed by Maddux and Willis (1993) remains: that ILSs can have the effect of shaping or driving a school's curriculum rather than responding to it.

Despite the amount of curriculum they cover and the number of activities they include, the success of ILSs hinges primarily on how they are viewed and implemented. When used only as a teacher replacement to provide individual student instruction, they seem less effective. When viewed as a supplement to other teacher methods and carefully integrated into a total teaching program, they seem more likely to have the desired impact on raising achievement.

One way to ensure appropriate and cost-effective uses of ILS products may be through a careful, well-planned purchasing process that involves both teachers and administrators. One such process was developed by the California Department of Education (Armstrong, 1999). This five-stage process (planning, pre-evaluation, evaluation, selection, and implementation/post-evaluation) is designed to "establish selection procedures that ensure that . . . curricular goals remain at the heart of the selection process" (p. 3). Guidelines to potential ILS purchasers based on those offered by Smith and Sclafani (1989), Chrisman (1992), and Vaille and Hall (1998) are summarized here:

- Clearly identify the problem the ILS is supposed to solve and understand the instructional theory on which the system is based.
- Determine whether the ILS is a closed system (one that provides 80% or more of the instruction for a given course) or an open system (one linked to the school's resources).
- Find out if the system's scope and sequence are matched to that of the school.
- Determine the target population for which the system was designed and whether or not it closely matches the characteristics of students with whom the ILS will be used.
- Consider the adequacy of the reporting and management system for the school's needs.
- Consider how much of its resources the school must spend on hardware and software.
- Project the educational benefits to the school from the system and compare them with the costs.
- Request that vendors inform the school on ILS updates.
- Carefully evaluate the grade-level courseware, management system, customization, and online tools and be sure they match the school's expectations.
- Set up reasonable terms of procurement and calculate the personnel and fiscal impact of the ILS.

How to Use ILSs in Teaching

Since an ILS creates a combination of the materials already described previously in this chapter, its potential benefits are similar. The highly interactive, self-pacing features of an ILS can help to motivate students who need highly structured environments; these activities free up the teacher's time for students who need personal assistance. Also, teachers can personalize instructional activities for each student by reviewing the extensive information on student and class progress provided by the ILS management system.

Successful uses of ILSs have been reported for two different kinds of teaching approaches: directed and constructivist.

Directed applications for ILSs. In a directed teaching approach, an ILS system can be used for remediation and as a mainstream delivery system.

- **For remediation.** Although ILSs are expensive alternatives to other kinds of delivery systems, White (1992) observes that "they will probably play an increasing role in the large urban systems that have faced achievement test scores that seem intractable to the usual classroom solutions" (p. 36). However, schools still must determine how ILS functions coordinate and complement those of the classroom teacher. Most ILS uses serve target populations that have typically presented the most difficult problems for traditional classroom activities: Chapter I groups, ESOL students, special education students, and at-risk students. Schools have tried and usually failed to reach these students with other methods.

■ **As a mainstream delivery system.** Rather than using an ILS only as a backup system to address educational problems, a school may let an ILS do the initial job of teaching whole courses for all students in a grade level. In light of the expense of ILSs, these uses are more rare. However, some alternative projects, like the Edison Project (Walsh, 1999), predict that the costs of using technology in this way will amount to substantially less over time than teacher salaries. Using ILSs to increase student-to-teacher ratios has stimulated ongoing debate and study.

In either of these uses, teachers still have important roles to play. As Blickman (1992) puts it, "ILSs allow teachers a comfortable transition from the role of deliverer of instruction to manager of instruction . . . [T]eachers are still actively engaged in the teaching process but as 'guides' or facilitators as opposed to distributors of information" (p. 46). American educators generally assume that ILSs should not be seen as "teacher proof" but rather "teacher enhancing." Teachers must still assign initial levels of work, follow up on student activities on the system, and give additional personal instruction when needed.

Constructivist applications for ILSs. Just as an integrated learning system combines several kinds of courseware to create a skill-based, directed learning environment, a network can also combine several kinds of technology resources to support the goals of constructivist learning approaches. When networks provide technology resources of constructivist design and use, the resulting products are sometimes labeled with terms other than ILS to differentiate them from what some educators consider more traditional uses of technology. For example, they may be called *integrated technology systems (ITSs), integrated learning environments, multimedia learning systems,* or *open learning systems* (Armstrong, 1999; Hill, 1993, p. 29).

ILS products useful for constructivist purposes provide varieties of unstructured tools on the same networked system as directed ones. Typically, there will be some kind of information bank (electronic encyclopedias), symbol pads (word processing and/or desktop publishing software), construction kits (Logo or other graphic languages or tools), and phenomenaria (computer simulations and/or problem-solving resources). They also usually have data-collection systems to track student usage of the system (Mageau, 1990). Thus, this kind of networked product can provide what Perkins (1991) called a "rich environment."

Evaluating and Selecting Instructional Software

In the 1980s, microcomputer courseware began to flood the educational market from such diverse sources as state projects, major publishing houses, and even cottage industries. This torrent made educators increasingly aware that simply putting instructional routines on the computer did not ensure that they would take advantage of its potential power as an instructional tool. Indeed, some of the products were so bad that they could be worse than no instruction at all.

It was during this era that courseware quality became a major issue in education and courseware evaluation evolved into a popular and highly publicized practice. Many professional magazines created sections to report the results of product evaluations; indeed, whole magazines like *Courseware Review* were developed to publish such evaluations. The Northwest Regional Lab's Microsoft Project and the Educational Products Information Exchange (EPIE) were just two of the many organizations that sprang up for the sole purpose of reviewing and recommending good instructional courseware.

As the field of educational technology matured and educators refined their attitudes toward computer use, the mystique of courseware faded and assumed more of the mundane aspects of purchasing any good instructional material. During the 1980s, teachers primarily evaluated and selected their own courseware. Now, state- and school district-level personnel increasingly control these purchases. Thus, the evaluation procedures and criteria have changed considerably from the early days of microcomputers. Regardless of who chooses the products, teachers should recognize that just because courseware addresses certain topics or skills, it does not mean that it will meet their needs.

The Need for Evaluation

Courseware quality is less troublesome now than it was in the early days of microcomputers when technical soundness frequently caused problems. For example, courseware programming did not anticipate all possible answers a student might give and did not account for all possible paths through a sequence of instruction. Consequently, programs frequently would "break" or stop when these unusual situations occurred. Early courseware also strongly emphasized entertainment value, giving less attention to educational value.

Courseware producers have obviously learned much from their early errors and problems, and overall quality has improved considerably. But educators still have good reasons for spending some time reviewing and/or evaluating courseware before selecting it for classroom use. Computerized instruction is not necessarily effective instruction, and eye-catching screen displays should not be the primary criteria for selecting materials.

Teachers should review courseware even after prescreening by committees or experts. Very often, state- or district-level committees are responsible only for selecting courseware that does not have gross problems and reaches the desired general level in a general content or topic area. Each teacher must then determine which specific curriculum needs and specific grade levels the package addresses and whether or not courseware functions fit with planned teaching strategies. It cannot be emphasized enough that courseware must match clearly identified instructional needs. It should *not* be used simply because it is available at a discount or supplied free by the state or district.

Courseware Evaluation Procedures: A Recommended Sequence

Evaluation procedures and criteria vary dramatically depending on whether a teacher is selecting courseware for a single classroom or is part of a district-level committee screening materials for use by many schools. One major difference is that committees generally must justify decisions to purchase one package over another by using weighted criteria checklists and assigning total point scores to individual packages. Small groups or individual teachers use much less formal procedures and criteria.

This section is designed primarily for individual teachers or small organizations like individual schools that (1) do not have large organizations purchasing courseware for them, (2) wish to supplement resources purchased for them by others, or (3) want to review preselected courseware to determine its usefulness for their immediate needs. These procedures are intended to help teachers anticipate and deal with problems related to courseware quality and to assist them in matching courseware to their classroom needs. The following sequence is recommended when selecting courseware for classroom use:

1. Begin with an identified need. Know what topics and skills you want to address and how you think you will use technology. This will require some knowledge of what kinds of instructional support technology has to offer.

2. Locate titles. As mentioned earlier in this section, teachers should probably not base their courseware purchasing decisions on descriptive reviews. Recommendations from colleagues and professional magazines and journals should serve primarily as leads. Some good sources for leads are:

- The Association for Supervision and Curriculum Development's (ASCD) summary of "Only the Best" software produced from their review of 25 annual software evaluations from organizations in the United States and Canada (ASCD, 1999)
- ISTE's *2001 Educational Software Preview Guide* (Johnson, 2001) with hundreds of reviewed and recommended titles
- The California Instructional Technology Clearinghouse, a searchable database with hundreds of software reviews, is available at http://clearinghouse.k12.ca.us
- The Educational Software Selector (TESS), a searchable database produced by the Consumers Union and containing more than 19,000 software product descriptions, is available at http://www.epie.com.

Once teachers discover a package they find interesting, they should use one or both of the next two general procedures to determine its usefulness.

3. Complete hands-on reviews. There is no substitute for running the courseware. Teachers should avoid reviewing demo packages (abbreviated versions of actual courseware), which can be misleading substitutes for the real thing. A typical hands-on review consists of two or three passes through a program: once to assess the package's capabilities and what it covers, and again to make incorrect responses and press keys that aren't supposed to be pressed in order to determine the program's ability to handle typical student use. Depending on its capabilities, the teacher may choose to go through the program one more time to review the usefulness and/or quality of particular demonstrations or presentations. Evaluation checklists such as the one given later in this chapter (see Figure 4.12) are usually helpful to guide teachers in a hands-on review. Tergan (1998) says that, given the number of packages teachers have to choose from and the time it takes to review each one, checklists can be a useful tool if the evaluator knows what to look for. He advocates that teachers either develop their own checklist or adopt one that seems thorough and well designed. Though many checklists are based on rating software and assigning points to various aspects, teachers usually are much less concerned with a total score than with making sure they have looked at all relevant characteristics.

4. Collect student reviews. Experienced teachers usually can tell from their own hands-on reviews when instructional materials are appropriate for their students. Even so, they are sometimes surprised at student reactions to courseware. Students sometimes encounter unexpected problems, or they may not seem to get out of the activity what the teacher expected they would. If at all possible, it is beneficial to field test courseware by observing students using it, getting their reactions, and, if possible, collecting data on their achievement. Gill, Dick, Reiser, and Zahner (1992) describe a detailed method for evaluating software that involves collecting data on student use.

Courseware Evaluation Procedures: Recommended Criteria

The set of recommended evaluation criteria in Figure 4.12 represents a synthesis from many sources (Comer & Geissler, 1998; Hoffman & Lyons, 1997; Roblyer, 1983). Teachers may find it helpful to use the essential criteria checklist shown in Figure 4.13 for the first pass, then use the more comprehensive list for a second pass. In addition to these essential criteria, teachers also may want to review the optional criteria described in Figure 4.14. These criteria may make the difference when teachers locate two or more packages that meet the essential criteria.

Figure 4.12 Explanation of Essential Criteria for Evaluating Instructional Courseware

Many sets of courseware criteria are available, but they tend to vary widely depending on the educational philosophy of the evaluator and the courseware functions being reviewed. Courseware criteria may be divided into two types: those that confirm *essential characteristics* are present and criteria that review *optional or situational characteristics* and are sometimes applicable and sometimes not, depending on the user's needs. The following is a comprehensive list of *essential criteria* based on Roblyer (1983), Hoffman and Lyons (1997), Lockard and Abrams (2001), and Vaille and Hall (1998). (See the essential criteria checklist in Figure 4.13. In addition to the following essential characteristics, this checklist contains a recommended format for such checklists, as well as essential criteria specific to several types of course functions.)

I. Essential Instructional Design and Pedagogy Characteristics: Does It Teach?

- **Appropriate teaching strategy, based on best known methods.** This covers a wide range of needs related to teaching methodology, e.g., providing enough examples for concept development, presenting ideas in a logical order, and including various components required for learning. For example, most educators would consider a mathematics package to be pedagogically flawed if it is intended for very young children and has no graphics. Learners at early stages of development are known to need concrete examples rather than text only.
- **Presentation on screen contains nothing that misleads or confuses students.** One particularly blatant error of this type was in a courseware package intended to teach young children about how the human body works. It depicted the human heart as a square box. Another, a math program, displayed a number of objects based on what the student answered, but never bothered to change the number of objects if it was a wrong answer. Thus, the student could be seeing the corrected numeral but the wrong number of objects.
- **Comments to students not abusive or insulting.** Programs must be sensitive to student's feelings, even if comments are intended humorously. One program based on a well-known cartoon cat with an acerbic personality belittled the student's name, saying "What kind of name is that for a worthy opponent?" It also commented on the student's "lack of mental ability" when a wrong answer was supplied. Although this was in keeping with the cat's persona, it was still inappropriate.
- **Readability at an appropriate level for students who will use it.** Although this may apply to any use of language in any program, it is particularly applicable to tutorials, which may require many explanations. For example, one tutorial for math skills at about the second-grade level had a great many explanations to read at about a fourth-grade level. This would probably not be an appropriate expectation for students who were having trouble with this level of math.
- **Graphics fulfill important purpose and are not distracting to learners.** Pictures and animation are considered motivational to students, but this is not always true. For example, animated feedback may be charming the first 10 times the students see it, but may achieve just the opposite effect after that. Also, some courseware attracts students' attention by flashing text or objects on the screen. This can be distracting when one is trying to focus on other screen text. Early courseware used a device called "scrolling" which had text moving up the screen as the student tried to read it, but this was quickly identified as a distracting mechanism and is rarely seen now.

II. Essential Content Characteristics: Is Content Accurate, Current, and Appropriate?

- **No grammar, spelling, or punctuation errors on the screen.** Even though a program may be on a nonlanguage topic, it should reflect correct spelling, grammar, and word uses, since students learn more than just the *intended* skills from instructional materials. One early courseware release on punctuation skills misspelled the word "punctuation" three different ways in the program!
- **All content accurate and up to date.** Many people are surprised to find accuracy errors with courseware material; they seem to trust content presented on a computer, as if the computer would correct the text itself if it becomes out of date! Content inaccuracies have been observed in a number of packages. For example, one program referred to blood as a "red substance," which, of course, is not always true. Instructional materials in social studies should be carefully screened for inaccurate reflections of country names, which are changing rapidly. Examples should be free of slang or other content that dates material and makes it less than useful to current students.
- **No racial or gender stereotypes; not geared toward only one sex or to certain races.** Look for diversity in the names and examples used. Are they all for "Dick and Jane" and are they always in the suburbs? Also review examples for gender stereotypes. Are all doctors men? Are all homemakers women? One famous simulation package required that students sign on only as males!
- **Social characteristics.** Does courseware exhibit a sensitive treatment of moral and/or social issues? For example, do games and simulations avoid unnecessary violence?
- **Match to instructional needs.** Does courseware match district or state curriculum objectives teachers are required to teach?

Figure 4.12 *(continued)*

III. Essential User Interface Characteristics: Is It "User Friendly" and Easy to Navigate?

- **User has appropriate control of movement within the program.** Depending on the purpose of the program, students normally should be able to go from screen to screen and read each screen at their own desired rate. They should also have obvious exit options available at any time.
- **User can turn off sound, if desired.** Since courseware may be used in classrooms, the teacher should have the ability to make the courseware quiet so it will not disturb others.
- **Interface is intuitive.** Screens and usage are consistent, allowing students to be able to use the software without extensive assistance.

IV. Essential Technical Soundness Characteristics: Does It Work Correctly?

- **Program loads consistently, without error.** A common problem in early courseware, problems of this kind now are seen rarely.
- **Program does not break, no matter what the student enters.** Again, this was a more common problem in early courseware. Programs should be designed to expect any possible answer, not just the correct or most obvious ones. When unexpected answers are entered, they should give an appropriate response to get the student back on track.
- **Program works on desired platform.** If one needs a Macintosh OS-based program, a program written exclusively for Windows is of little use (and vice versa). Also, the program should work on the version of the operating system one needs (e.g., Windows NT vs. Windows 2000).
- **Program does what the screen says it should do.** If the screen indicates the student should be able to exit or go to another part of the program, this capability should be allowed as stated.
- **Online links work correctly.** A feature in many new courseware packages is the capability of linking students to Internet web sites for additional demonstrations or interactions with others. If links are given, they should work as designated. If links are to nonworking sites, the user should have the option to eliminate or correct them.
- **Videos and animations work correctly.** If a screen is to display a moving graphic object (e.g., video or animation), it should work as indicated. Sometimes, a graphic requires so much memory it works too slowly to be practical for the classroom.

Selecting Software for Constructivist Versus Directed Uses

Although descriptions of instructional software in the literature are changing, many references to courseware evaluation criteria and evaluation methods focus on products to be used with directed instruction. While many criteria are appropriate for software designed for both kinds of uses, additional details often are lacking on what to look for in software that will be used with constructivist methods. Constructivist activities tend to emphasize multimedia and distance learning products rather than drill or tutorial software. For example, Litchfield (1992) lists criteria for "inquiry-based science software and interactive multimedia programs." Checklists by Hoffman and Lyons (1995) and Vaille and Hall (1998) are among those that include criteria for more open-ended products. Further criteria and methods for evaluating multimedia and online multimedia products will be discussed in Chapters 7 and 8.

Figure 4.13 Essential Criteria Checklist for Evaluating Instructional Courseware

The following is an example checklist based on essential qualities that can be used to discriminate between acceptable and unacceptable courseware material. If courseware does not meet these criteria, it probably should not be considered for purchase. For each item, indicate Y for yes if it meets the criterion, or N for no if it does not.

Title ______________________ Publisher ______________________

Content Area ______________________ Hardware Required: ______________________

Courseware functions:

______ Drill and practice ______ Instructional game

______ Tutorial ______ Problem solving

______ Simulation ______ Other:

I. Instructional Design and Pedagogical Soundness

______ Teaching strategy is matched to student needs/levels and is based on accepted methods

______ Presentation on screen contains nothing that misleads or confuses students

______ Readability and difficulty are at an appropriate level for students who will use it

______ Comments to students are not abusive or insulting

______ Graphics fulfill important purpose (motivation, information) and are not distracting to learners

Criteria specific to drill-and-practice functions:

______ High degree of control over presentation rate (unless the method is timed review)

______ Appropriate feedback for correct answers (none, if timed; not elaborate or time consuming)

______ Feedback is more reinforcing for correct than for incorrect responses

Criteria specific to tutorial functions:

______ High degree of interactivity (not just reading information)

______ High degree of user control (forward and backward movement, branching upon request)

______ Comprehensive teaching sequence so instruction is self-contained and stand-alone

______ Adequate answer-judging capabilities for student-constructed answers to questions

Criteria specific to simulation functions:

______ Appropriate degree of fidelity (accurate depiction of system being modeled)

______ Good documentation available on how program works

Criteria specific to instructional game functions:

______ Low quotient of violence or combat-type activities

______ Amount of physical dexterity required appropriate to students who will use it

II. Content

______ No grammar, spelling, or punctuation errors on the screen

______ All content accurate and up to date

______ No racial or gender stereotypes; not geared toward only one sex or to certain races

______ Exhibits a sensitive treatment of moral and/or social issues (e.g., perspectives on war or capital punishment)

______ Content matches required curriculum objectives

III. User Flexibility

______ User normally has some control of movement within the program (e.g., can go from screen to screen at desired rate; can read text at desired rate; can exit program when desired)

______ Can turn off sound, if desired

______ Interface is easy to use (e.g., similar format from screen to screen for forward and back movement in program)

IV. Technical Soundness

______ Program loads consistently, without error

______ Program does not break, no matter what the student enters

______ Program does what the screen says it should do

______ Program works on desired platform

______ If included, online links work as indicated

______ If included, animations and videos work as indicated

Decision:

______ Is recommended for purchase and use

______ Is not recommended

Figure 4.14 Optional Criteria for Evaluating Instructional Courseware

Teachers reviewing courseware may consider a great many other criteria depending on their needs, the program's purpose, and the intended audience. These are detailed in Roblyer (1983), Hoffman and Lyons (1997), Lockard and Abrams (2001), and Vaille and Hall (1998). Many of these criteria, which are listed below, are subjective in nature or dependent on teacher needs; it is up to the teacher to decide whether or not the courseware meets them and/or whether or not they are important enough to affect selection decisions.

Optional Instructional Design Criteria

- **Stated objectives.** Does the courseware state its objectives and are stated objectives likely to be attained through courseware activities?
- **Prerequisite skills.** Are skills specified that students will need in order to use the courseware activities and are students likely to be able to acquire the skills?
- **Interest quotient.** Are examples and strategies likely to interest students at the targeted level?
- **Presentation logic.** Do instructional units follow a logical sequence based on skill hierarchies?
- **Tests.** Do tests match stated skills and are they good measures of the skills?
- **Significance.** Are stated skills "educationally significant" (e.g., in the curriculum)?
- **Use of medium.** Does courseware make good use of computer capabilities?
- **Field testing.** Is there evidence the courseware has been used with students and revised based on this feedback before its release?

Optional Interface/Navigation Criteria

- **Student ease of use.** Is the program easy to use for the intended students? Does it require physical dexterity to answer items the students may not have even though they know the correct answers? Is a lot of typing required?
- **Required keys.** Are the keys required to input answers easy to remember (e.g., pressing back arrow for going back)?
- **Input devices.** Are alternate input devices allowed to make courseware more usable for special populations?
- **Directions.** Are there on-screen directions on how to use it?
- **Shortcuts.** Lengthy introductory screens may be bypassed, if desired.
- **Support materials.** Are there print support materials to support on-screen activities?
- **Optional assistance.** Is a "HELP" feature available if the student runs into difficulty?
- **Optional directions.** Can students skip directions, if they desire, and go straight to the activities?
- **Creativity.** Do materials foster creativity rather than just rote learning?
- **Summary feedback.** Are students given an on-screen summary of performance when they finish working?

Optional Teacher Use Criteria

- **Teacher ease of use.** Can teachers figure out, with minimum effort, how to work the program?
- **Management.** Does courseware contain adequate record-keeping and management capabilities?
- **Teacher manuals.** Are clear, nontechnical teacher manuals available with the courseware? Are manuals well produced and do they include a good table of contents and index?
- **Ease of integration.** Are courseware materials designed to integrate easily into other activities the teacher is doing?
- **Teacher assistance.** Does courseware improve the teacher's ability to teach the subject?
- **Adaptability.** Can teachers modify and adapt the courseware for their needs by changing content (e.g., spelling words) or format (e.g., animated versus written feedback)?

Optional Presentation Criteria

- **Graphics features.** Are graphics, animation, and color used for instructional purposes rather than flashiness?
- **Colors.** Are colors required or is software still useful on noncolor monitors?
- **Screen layout.** Are screens so "busy" or cluttered that they interfere with reading?
- **Audio and speech capabilities.** Is audio and speech of adequate quality so students can understand it easily?
- **Video and animation.** Do moving graphics display clearly, quickly, and without jerkiness? Are they in a high enough resolution to be useful?
- **Required peripherals.** Does the program require peripherals the schools are likely to have (e.g., light pens, speech synthesizers)?
- **Screen printing.** Can key screens from the courseware (e.g., summary performances) be printed?

Figure 4.14 ***(continued)***

Optional Technical Criteria

- **Response judging.** Does the response judging allow for ALL possible correct answers and disallow ALL possible incorrect ones?
- **Timing.** Does the program present itself quickly so displays and responses are accomplished without noticeable delays?
- **Portability.** Can teachers transfer the courseware from one machine to another?
- **Compatibility.** Does courseware run on more than one platform?
- **Components.** Are all required drivers and plug-ins identified and either provided or easily downloadable online?
- **Technical manuals.** Do teacher or user manuals contain technical documentation on program operation and any technical features or options? Does the manual tell how to install and uninstall the program?

Optional Publisher Support

- **Cost effectiveness.** Is the price of the package appropriate in light of what it accomplishes?
- **Available versions.** Is the program available in desired versions (e.g., network or site license)? Does the company provide for free or discounted upgrades later?
- **Preview allowed.** Will the company allow free previews? Will they refund the purchase price or supply a replacement if user is not satisfied, or if software is lost, stolen, or damaged?
- **Backup.** Is a backup disk provided or can user make one?
- **Training.** For more complicated course packages, is on-site or web-based training provided to buyers, and is there a newsletter or other way to communicate applications and updates?
- **Packaging.** Is it made to stand up to normal school and classroom wear? Are disks labeled clearly as to program part and platform?
- **Ongoing support.** Does the company answers questions and provide help for problems via local representatives, a toll-free telephone line, or a web site?

Exercises

Companion Website

Record and apply what you have learned.

Chapter 4 Self-Test

To review terms and concepts in this chapter, take the Chapter 4 self-test. Select Chapter 4 from the front page of the Companion Website (located at http://www.prenhall.com/roblyer), then choose the *Multiple Choice* module.

Portfolio Activities

ISTE The following activities address ISTE National Educational Technology Standards for Teachers (NETS-T) and will help you add to your professional portfolio. To complete these activities online and save or submit the materials electronically, select Chapter 4 from the front page of the Companion Website (http://www.prenhall.com/roblyer), then choose the *Portfolio* module.

1. ***Instructional Software Examples (NETS-T Standards: I-B, II-C)*** From instructional software packages, select at least one that represents each function described in this chapter. Using word processing or multimedia software, prepare a description of the software that focuses on which function(s) it fulfills.
2. ***Instructional Software in a Content Area (NETS-T Standards: I-B, II-C, V-B)*** On the Internet, do a search for software examples in your content area or grade level. Prepare a list of the sites with good examples of each type of software function.

Questions for Thought and Discussion

These questions may be used for small-group or class discussion or may be subjects for individual or group activities. To take part in these discussions online, select Chapter 4 from the front page of the Companion Website (http://www.prenhall.com/roblyer), then choose the *Message Board* module.

1. The tendency to refer to drill and practice software by the derogatory term "drill and kill" is growing. Is this because the number of situations is diminishing in which drill and practice software would be the strategy of choice or because people fail to recognize appropriate situations for using it?
2. Some schools, like those with a college preparatory focus, do not allow the use of instructional games of any kind. Is there a compelling case to be made for allowing the use of instructional game software to achieve specific educational goals? That is, can games do something in an instructional situation that no other strategy is able to do? If so, what?

Collaborative Activities

ISTE

The following activities address ISTE National Educational Technology Standards for Teachers (NETS-T) and can be done in small groups. Each group should present the findings to the class in a format they know how to use (word-processed report, presentation software, multimedia product). Completed group products can be copied and shared with the entire class and/or included in each person's personal portfolio.

1. ***Courseware Evaluation (NETS-T Standards: I-B, II-C)*** Class members obtain one or more example instructional software packages, either from the instructor or from their own schools. Working in small groups, each selects one of the categories of software criteria (e.g., instructional design, content, user interface) from the courseware criteria checklists in Figures 4.17 and 4.18. Using the criteria under the category, the groups evaluate the courseware package. Each group also identifies the functions they believe are represented in the package. Each group prepares a description and demonstration to illustrate the software characteristics they observe. They present these to the class.
2. ***Matching Curriculum Needs with Instructional Software (NETS-T Standards: II-A, B, C)*** Instructor and students agree on a set of specific state or national (e.g., NCTE, NCTM, NSTA) curriculum standards and/or skills on which each small group will work. Each group identifies one or more instructional software packages that could help teachers address each skill. They present their findings to the class.
3. ***Lesson Integration Strategies for Instructional Software (NETS-T Standards: II-A through E; II-C, D; III-B; IV-A, B)*** Using the five-step integration sequence described in the lesson in Chapter 2, Figure 2.6, example, small groups prepare an integration sequence for an instructional software package to be used in a real or fictional classroom. (Students should pay special attention to the "justification" description in Step 1: What's the Relative Advantage—Why Use Instructional Software?) They present their findings to the class.

Integrating Technology Across the Curriculum Activities

The *Integrating Technology Across the Curriculum* CD-ROM is a set of technology integration ideas and links to online lessons, arranged as a searchable database. The CD comes packaged with this textbook. Complete the following exercise using this CD:

Simulations are considered some of the most powerful and versatile of the instructional software functions. Most simulations are used in science and social studies content areas, but the ways they are integrated differ greatly between the two content areas. Locate three to five integration ideas for science and a like number for the social studies area; compare and contrast the integration strategies used in the two lessons. Prepare a two-paragraph description summarizing the differences in integration strategies between science simulations and social studies simulations, and identify what you can conclude about the instructional needs simulations can meet in each content area.

References

Adams, C. (1998). Teaching and learning with SimCity 2000. *Journal of Geography, 97*(2), 47–55.

Alessi, S. (1988). Fidelity in the design of computer simulations. *Journal of Computer-Based Instruction, 15*(2), 40–47.

Alessi, S., & Trollip, S. (2001). *Multimedia for learning: Methods and development.* Needham Heights, MA: Allyn and Bacon.

Allen, D. (1993). Exploring the earth through software. Teaching with technology. *Teaching PreK–8, 24*(2), 22–26.

Andaloro, G. (1991). Modeling in physics teaching: The role of computer simulation. *International Journal of Science Education, 13*(3), 243–254.

Armstrong, S. (1999). *A framework for evaluating integrated learning systems/multimedia instructional systems.* Modesto, CA: California Instructional Technology Clearinghouse.

Arnett, P. (2000). Mastering reading and writing with technology. *Media & Methods, 37*(1), 12–14.

Association for Supervision and Curriculum Development (ASCD). (1999). *Only the best: The annual guide to the highest rated educational software and multimedia.* Alexandria, VA: Author (ERIC Reproduction No. ED 437 897).

Baek, Y., & Layne, B. (1988). Color, graphics, and animation in a computer-assisted learning tutorial lesson. *Journal of Computer-Based Instruction, 15*(4), 31–35.

Bailey, G., & Lumley, D. (1991). Supervising teachers who use integrated learning systems. *Educational Technology, 31*(7), 21–24.

Bearden, D., & Martin, K. (1998). My make believe castle: An epic adventure in problem solving. *Learning and Leading with Technology, 25*(5), 21–25.

Becker, H. (1992). Computer-based integrated learning systems in the elementary and middle grades: A critical review and synthesis of evaluation reports. *Journal of Educational Computing Research, 8*(1), 1–41.

Becker, H. (1994). Mindless or mindful use of integrated learning systems. *International Journal of Educational Research, 21*(1), 65–79.

Bender, P. (1991). The effectiveness of integrated computer learning systems in the elementary school. *Contemporary Education, 63*(1), 19–23.

Bentley, E. (1991). Integrated learning systems: The problems with the solution. *Contemporary Education, 63*(1), 24–27.

Blickman, D. (1992). The teacher's role in integrated learning systems. *Educational Technology, 32*(9), 46–48.

Bloom, B. (1986). Automaticity. *Educational Leadership, 43*(5), 70–77.

Blosser, P. (1988). Teaching problem solving—Secondary school science (ERIC Document Reproduction No. ED 309 049).

Bracy, G. (1992). The bright future of integrated learning systems. *Educational Technology, 32*(9), 60–62.

Brush, T. (1998). Embedding cooperative learning into the design of integrated learning systems: Rationale and guidelines. *Educational Technology Research and Development, 46*(3), 5–18.

Brush, T., Armstrong, J., & Barbrow, D. (1999). Design and delivery of integrated learning systems: Their impact on student achievement and attitudes. *Journal of Educational Computing Research, 21*(4), 475–486.

CAI in music. (1994). *Teaching Music, 1*(6), 34–35.

Cann, A., & Seale, J. (1999). Using computer tutorials to encourage reflection. *Journal of Biological Education, 33*(3), 130–132.

Chrisman, G. (1992). Seven steps to ILS procurement. *Media and Methods, 28*(4), 14–15.

Clinton, J. (1991). Decisions, decisions. *The Florida Technology in Education Quarterly, 3*(2), 93–96.

Comer, R., & Geissler, C. (1998). *A methodology for software evaluation.* Paper presented at the 1998 meeting of the Society for Information Technology and Teacher Education, Washington, DC, March 10–14 (ERIC Document Reproduction No. ED 421 140).

Eiser, L. (1988). What makes a good tutorial? *Classroom Computer Learning, 8*(4), 44–47.

Estep, S., McInerney, W., & Vockell, E. (1999–2000). An investigation of the relationship between integrated learning systems and academic achievement. *Journal of Educational Technology Systems, 28*(1), 5–19.

Estes, C. (1994). The real-world connection. *Simulation and Gaming, 25*(4), 456–463.

Flowers, R. (1993). New teaching tools for new teaching practices. *Instructor, 102*(5), 42–45.

Funkhouser, C., & Dennis, J. (1992). The effects of problem-solving software on problem-solving ability. *Journal of Research on Computing in Education, 24*(3), 338–347.

Gagné, R. (1982). Developments in learning psychology: Implications for instructional design. *Educational Technology, 22*(6), 11–15.

Gagné, R., & Merrill, M. D. (1990). Integrative goals for instructional design. *Educational Technology Research and Development, 38*(1), 23–30.

Gagné, R., Wager, W., & Rojas, A. (1981). Planning and authoring computer-assisted instruction lessons. *Educational Technology, 21*(9), 17–26.

Galas, P. (1998). From presentation to programming. *Learning and Leading with Technology, 25*(4), 18–21.

Gill, B., Dick, W., Reiser, R., & Zahner, J. (1992). A new model for evaluating instructional software. *Educational Technology, 32*(3), 39–48.

Gonsalves, D., & Lopez, J. (1998). Catch your students with microworlds games. *Learning and Leading with Technology, 26*(3), 19–21.

Gore, K. (1987–1988). Problem solving software to implement curriculum goals. *Computers in the Schools, 4*(3–4), 7–16.

Graham, R. (1994). A computer tutorial for psychology of learning courses. *Teaching of Psychology, 21*(2), 116–166.

Graham, R. (1998). A computer tutorial on the principles of stimulus generalization. *Teaching of Psychology, 25*(2), 149–151.

Hasselbring, T. (1988). Developing math automaticity in learning handicapped children. *Focus on Exceptional Children, 20*(6), 1–7.

Hasselbring, T., & Goin, L. (1993). Integrating technology and media. In E. Polloway & J. Patton (Eds.), *Strategies for teaching learners with special needs* (5th ed.). New York: Merrill.

Henderson, L., Klemes, J., & Eshet, Y. (2000). Just playing a game? Educational simulation software and cognitive outcomes. *Journal of Educational Computing Research, 22*(1), 105–129.

Higgins, K., & Boone, R. (1993). Technology as a tutor, tools, and agent for reading. *Journal of Special Education Technology, 12*(1), 28–37.

Hill, M. (1993). Chapter I revisited: Technology's second chance. *Electronic Learning, 13*(1), 27–32.

Hoffman, J. L., & Lyons, D. L. (1997). Evaluating instructional software. *Learning and Leading with Technology, 25*(2), 52–56.

Johnson, J. (1987). Do you think you might be wrong? Confirmation bias in problems solving. *Arithmetic Teacher, 34*(9), 13–16.

Johnson, J. M. (2001). *2001 educational software preview guide.* Eugene, OR: ISTE.

Kahn, J. (1998–1999). The same but different: The computer as an alternate medium. *Learning and Leading with Technology, 26*(4), 15–18.

Kraemer, K. (1990). SEEN: Tutorials for critical reading. *Writing Notebook, 7*(3), 31–32.

Litchfield, B. (1992). Science: Evaluation of inquiry-based science software and interactive multimedia programs. *The Computing Teacher, 19*(6), 41–43.

Lockard, J., & Abrams, P., (2001). *Computers for twenty-first century educators* (5th ed.). New York: Longman.

Maddux, C., & Willis, J. (1993, Spring/Summer). Integrated learning systems: What decision-makers need to know. *ED TECH Review,* 3–11.

Mageau, T. (1990). ILS: Its new role in schools. *Electronic Learning, 10*(1), 22–24.

Malone, T. (1980). *What makes things fun to learn? A study of intrinsically motivating computer games.* Palo Alto, CA: Xerox Palo Alto Research Center.

Martin, K., & Bearden, D. (1998). Listserv learning. *Learning and Leading with Technology, 26*(3), 39–41.

Mayes, R. (1992). The effects of using software tools on mathematics problem solving in secondary school. *School Science and Mathematics, 92*(5), 243–248.

McCoy, L. (1990). Does the Supposer improve problem solving in geometry? (ERIC Document Reproduction No. ED 320 775).

McGinley, R. (1991). Start them off with games! *The Computing Teacher, 19*(3), 49.

Merrill, D., & Salisbury, D. (1984). Research on drill and practice strategies. *Journal of Computer-Based Instruction, 11*(1), 19–21.

Miller, H. L. (1997). The New York City Public Schools Integrated Learning Systems Project. *International Journal of Educational Research, 27*(2), 91–183.

Mintz, R. (1993). Computerized simulation as an inquiry tool. *School Science and Mathematics, 93*(2), 76–80.

Muckerheide, P., Mogill, A., & Mogill, H. (1999). In search of a fair game. *Mathematics and Computer Education, 33*(2), 142–150.

Murray, T., et al. (1988). An analogy-based computer tutorial for remediating physics misconceptions (ERIC Document Reproduction No. ED 299 172).

Norris, W. (1977). Via technology to a new era in education. *Phi Delta Kappan, 58*(6), 451–459.

Okolo, C. (1992). The effect of computer-assisted instruction format and initial attitude on the arithmetic facts proficiency and continuing motivation of students with learning disabilities. *Exceptionality: A Research Journal, 3*(4), 195–211.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas.* New York: Basic Books.

Perkins, D. (1991). Technology meets constructivism: Do they make a marriage? *Educational Technology, 31*(5), 18–23.

Ploger, D., & Vedova, T. (1999). Programming dynamic charts in the elementary classroom. *Learning and Leading with Technology, 26*(5), 38–41.

Randel, J., Morris, B., Wetzel, C., & Whitehill, B. (1992). The effectiveness of games for educational purposes: A review of recent research. *Simulation and Gaming, 23*(3), 261–276.

Reigeluth, C., & Schwartz, E. (1989). An instructional theory for the design of computer-based simulations. *Journal of Computer-Based Instruction, 16*(1), 1–10.

Richards, J. (1992). Computer simulations in the science classroom. *Journal of Science Education and Technology, 1*(1), 67–80.

Rieber, L., Smith, L., & Noah, D. (1998). The value of serious play. *Educational Technology, 38*(6), 29–37.

Robinson, M., & Schonborn, A. (1991) Three instructional approaches to *Carmen Sandiego* software series. *Social Education, 55*(6), 353–354.

Roblyer, M. (1983). How to evaluate software reviews. *Executive Educator, 5*(9), 34–39.

Ronen, M. (1992). Integrating computer simulations into high school physics teaching. *Journal of Computers in Mathematics and Science Teaching, 11*(3–4), 319–329.

Salisbury, D. (1990). Cognitive psychology and its implications for designing drill and practice programs for computers. *Journal of Computer-Based Instruction, 17*(1), 23–30.

Sherman, T. (1987–1988). A brief review of developments in problem solving. *Computers in the Schools, 4*(3–4), 171–178.

Shore, A., & Johnson, M. (1992). Integrated learning systems: A vision for the future. *Educational Technology, 32*(9), 36–39.

Simmons, P., & Lunetta, V. (1993). Problem-solving behaviors during a genetics computer simulation. *Journal of Research in Science Teaching, 30*(2), 153–173.

Smith, K. (1992). Earthquake! *The Florida Technology in Education Quarterly, 4*(2), 68–70.

Smith, R. A., & Sclafani, S. (1989). Integrated teaching systems: Guidelines for evaluation. *The Computing Teacher, 17*(3), 36–38.

Steinburg, R., & Oberem, G. (2000). Research-based instructional software in modern physics. *Journal of Computers in Mathematics and Science Teaching, 19*(2), 115–136.

Stokes, J. (1999). Problem solving software, equity, and allocation of roles. *Learning and Leading with Technology, 26*(5), 6–9, 30.

Taylor, R. (1980). *The computer in the school: Tutor, tool, and tutee.* New York: Teachers College Press.

Tergan, S. (1998). Checklists for the evaluation of education software: Critical review and prospects. *Innovations in Education and Training, 35*(1), 9–20.

Trotter, A. (1991). In the school game, your options abound. *Executive Educator, 13*(6), 23.

Vaille, J., & Hall, J. (1998). Guidelines for the evaluation of instructional technology resources. Eugene, OR: ISTE.

Van Dusen, L., & Worthen, B. (1995). Can integrated instructional technology transform the classroom? *Educational Leadership, 53*(2), 28–33.

Walsh, J. (1999). Edison Project, now Edison Schools Inc., plans to go public. *Education Week, 19*(1), 6.

Weinstein, P. (1999). Computer programming revisited. *Technology & Learning, 19*(8), 38–42.

Weizenbaum, J. (1976). *Computer power and human reason.* San Francisco: W. H. Freeman & Co.

White, M. (1992). Are ILSs good for education? *Educational Technology, 32*(9), 49–50.