CHAPTER 5

Instructional Systems Design and CAI

Confusion often occurs because many new teacher/authors encounter for the first time both a relatively new approach to instructional design and a new delivery vehicle. The result of this combination seems to be a lack of confidence about where to begin and a misunderstanding of which of the two—instructional design or computer systems—should provide the basis for development of (CAI) materials.

(Eisele, 1978, p. 14)

In Chapter 4 we examined principles derived from educational psychology and their implications for CAI design. Instructional design systems, systematic approaches to the development of instruction in any medium, provide sets of procedures that designers can follow in order to produce instruction based on these principles.

We are strong supporters of instructional design systems. However, development of the knowledge and skill required to understand and apply instructional design creatively and to several media is beyond the scope of this book. Attempts to teach instructional design and other aspects of CAI development have resulted in the problems Eisele described. Many colleges and universities offer one or more courses in the design of instruction. We recommend such courses for the professional designer.

The goal of this chapter is to provide a working knowledge of instructional design, applied specifically to the task of creating computerized instruction. A checklist has been provided, which applies instructional design to CAI. A carpenter doesn't need to know how to build the tools of the profession. The master carpenter modifies tools and becomes expert in using those tools to produce the desired results. In much the same way, by learning to use the tools provided in this book and adjusting them to work most effectively, you will be able to produce sound CAI lessons.

OBJECTIVES

Comprehension

After completing this chapter you will be able to

1. Discuss the meaning of the term *instructional design system.*
2. Name and diagram the phases of the CAI Design Model (CDM).

3. Discuss the flow from phase to phase within the CDM.
4. List at least five activities to be completed or questions to be answered in each phase.
5. Discuss when and how to use the CAI Design Checklist (CDC).
6. Discuss the advantages and disadvantages of a team approach to CAI development.

Application

After completing this chapter you will be able to

Complete sections 1.1–1.4 of the CAI Design Checklist based on a CAI project you are likely to undertake in the near future.

Systematic Approaches to the Design of Instruction

Is it possible to use a single plan to develop instruction on topics as diverse as learning chemical formulas, learning to play a musical instrument, learning to drive an automobile, learning to divide fractions by mixed numbers, and understanding the implications of nuclear war? Is it possible to use a single plan that will produce effective instruction for pre-school children as well as for graduate students and senior citizens? Can it be used in schools, homes, and businesses? For print-based, video-based, or computer-based delivery? Experts would answer yes to all of these questions.

Individuals in the field of instructional systems design (ISD or ID) have developed procedures to produce efficient, effective instruction for a variety of outcomes, learners, environments, and media. Several models have been developed to lead the designer through the process (Briggs and Wager, 1981; Davis, Alexander, and Yelon, 1974; Dick and Carey, 1985; Gagné and Briggs, 1979; Smith and Boyce, 1984). These models differ in a number of ways, but their differences are slight when compared to their similarities.

Instructional design *systems*, as the name implies, are composed of a series of interrelated steps based on principles derived from educational research and theory. Table 5.1 lists steps generally found in instructional design systems.

The initial step in solving any problem is to gain a thorough understanding of it. In solving instructional problems, this involves determining goals and objectives. It then becomes possible to break each objective down further, uncovering several levels of tasks. Once an understanding of all of the skills and knowledge required has been gained, the designer must decide which of these will be included in the lesson and which will be assumed as prerequisites before the lesson begins.

Understanding the instructional task is critical, but so is understanding the constraints under which development and production are to occur. Knowledge of the target population for which the lesson is to be developed, as well as awareness of deadlines, budget, equipment, and human resource availability are also required. After gaining an understanding of the instructional task and the constraints within which work will proceed, the designer generates multiple alternative methods for solving each aspect of the problem. Several ways to achieve each objective are identified, after which the designer selects those with the best probability of

Table 5.1
Steps Generally Included in Instructional Design Systems

- Identification of instructional goals and objectives
- In-depth analysis of the tasks and subtasks involved
- Determination of the prerequisite skills and knowledge required for the lesson to be successful (which skills will be taught and which will be assumed before the lesson begins)
- Identification of the constraints (such as time and budget) within which the product must be developed
- Generation of instructional alternatives
- Selection from among the alternatives generated
- Determination of the sequence in which the tasks are to be achieved
- Development of the instruction
- Testing the accuracy of the program execution
- Evaluation of the instruction
- Revision of the instruction

succeeding, given the task and constraints identified. After the sequence in which the instructional events will occur is determined, development begins. Evaluation of the product occurs during as well as after development, and revisions may be made to the plan or the product at any time.

The steps are generally combined into phases. The number of steps and the number of phases into which the steps are grouped differ from model to model. The CAI design model (CDM) presented in Figure 5.1 combines these activities into four phases: needs assessment, design, development and implementation, and evaluation and revision.

The model generally flows from left to right. Notice that the only pathway from phase to phase is through "Evaluation and Revision." When each of the three phases has been completed, progress is evaluated, and the project may move to the next phase, may return to the current phase for additional attention, or may be returned to a previous phase for modification. Notice also that there is no stopping point. During CAI development, a good deal of time is spent cycling between implementation, evaluation, and revision.

PHASE 1: NEEDS ASSESSMENT

The purpose of needs assessment is to define clearly the specifications of a project. During this phase, the designer develops an understanding of the student for whom the program is to be developed, the environment in which the program will

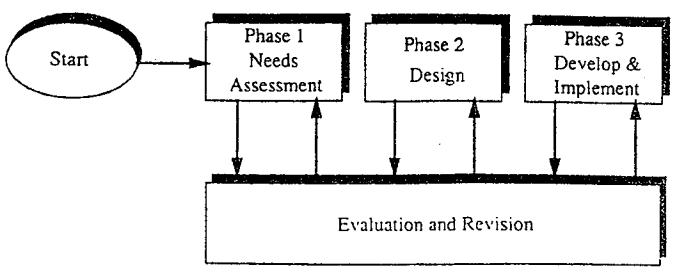


Figure 5.1
The CAI design model (CDM).

be used, the constraints within which the program will be developed, the goals and objectives the program is to achieve, and the assessment items that will be used to determine the extent to which objectives have been met. The designer identifies the skills and knowledge the student is to acquire during the lesson, as well as those the student must possess *prior to beginning* the lesson. Chapters 7 and 8 present more detailed accounts of task analysis and the writing of objectives.

When needs assessment has been completed, attention is shifted to the evaluation and revision phase, where needs assessment findings are scrutinized. If necessary, the project is returned to the needs assessment phase for additional information.

PHASE 2: DESIGN

After completing the needs assessment, the designer understands the problem more completely. Expectations have been clarified, resources and constraints have been identified, and other parameters of the problem have been defined. The purpose of phase 2 is to identify and document the best means of achieving the desired results.

The first step is to determine the sequence in which the objectives will be met. After the sequence has been determined, several potential solutions are considered *for each objective identified during the needs assessment.* After potential solutions have been identified for an objective, the best solution is selected, given the nature of the problem. For example, the options generated for a unit on adding fractions might include illustrating the concept using a number line, illustrating the concept using segments of a pie, or conveying the concept using only text without any illustrations. Once these options have been identified, one may be selected based both on the information gained during needs assessment and on the probability of success.

During this phase, the designer determines the sequence in which the objectives will be met, expands the list of objectives and assessment items to include a description of the activities that will meet each objective, and transfers the entire lesson to storyboards. Storyboards are illustrations depicting each change to the computer screen and conveying other important information to the reviewers and programmers. A sample storyboard is depicted in Figure 5.2.

At the top of the storyboard, the designer indicates the number of the objective that the screen supports, the screen number, and the number of screens that relate to the objective. The screen depicted is the third of seven screens used to achieve objective number four. In this example, the body of the storyboard illustrates how the screen will look *at the end* of the sequence. This screen has a title, "Dividing by Fractions," which is repeated on all of the screens for this objective. Below the title, the rule is stated, after which two examples are displayed. As the comments to the programmer indicate, the examples are displayed using a technique sometimes called progressive disclosure, in which sections of the screen are displayed in sequence. In this example, the title and rule are displayed first. The student reads the title and rule and then presses the spacebar. Initially, only the problem is displayed. After a three-second pause, the solution begins by flashing the multiplication sign and the inverted fraction. After three more seconds, the problem is completed, and the "Press Spacebar to Continue" prompt returns. When the student responds, a second problem is displayed in the same manner. Although the notes to the programmer are quite complete in this example,

Figure 5.2
A sample storyboard.

Objective # 4 Board # 3 of 7

DIVIDING BY FRACTIONS

RULE: DIVIDING BY A FRACTION PRODUCES THE SAME RESULT AS MULTIPLYING BY ITS INVERSE.

FOR EXAMPLE: $\frac{3}{4} \div \frac{1}{8} = \frac{3}{4} \times \frac{8}{1} = \frac{24}{4} = 6$

ANOTHER EXAMPLE: $\frac{2}{3} \div \frac{5}{6} = \frac{2}{3} \times \frac{6}{5} = \frac{12}{15} = \frac{4}{5}$

–PRESS SPACEBAR TO CONTINUE–

Preceding Storyboard(s): 4-2

Following Storyboard: 4-4

Action Required to Advance: SPACEBAR

Comments: Step 1 = Title + Rule only - Student presses spacebar
Step 2 = "For example: 3/4 ÷ 1/8 =" followed by 3 second pause
Step 3 = "3/4 × 8/1" with "× 8/1" flashing - 3 second pause
Step 4 = complete solution to first problem - spacebar -
Step 5-7 = Display second problem in same manner.

abbreviations and other conventions allow the designer and programmer to communicate more efficiently.

After illustrating the content of the lesson, the designer determines the extent to which performance will be monitored and reviews the storyboards to identify

methods to maximize interaction and to increase the program's ability to account for individual differences.

Once phase 2 activities have been completed, the status is reviewed during the evaluation and revision cycle, where the quality and validity of activities from both phases 1 and 2 are assessed. Purcell (1984) recommends a "walk-through," where the design is explained to a team of evaluators who provide feedback before programming begins. The project may be redirected to either needs assessment or design phases if additional effort is deemed necessary, or it may proceed to phase 3 for development and implementation.

PHASE 3: DEVELOPMENT AND IMPLEMENTATION

As phase 3 begins, the lesson exists only on paper. The activities in phase 3 convert the lesson from paper to the point at which it is actually used by students in the target environment. Activities during this phase include flowcharting, authoring and/or programming, testing and debugging, documenting lesson procedures, formative evaluation, summative evaluation, and revision. The computer program is produced during this phase. The final product is a CAI lesson in the form of a computer program or set of programs that accomplishes the prescribed goals and objectives within the constraints identified during needs assessment.

The first step during phase 3 is the development of a flowchart, a diagram indicating the possible paths through the lesson. Although sophisticated flowcharting conventions are often used by experienced programmers, the primary goal is for both programmer and designer to understand the lesson execution "blueprint."

After completing a lesson flowchart, authoring or programming the lesson may begin. This step involves the writing of commands needed to accomplish the conditions and actions prescribed in the flowchart and illustrated in the storyboards. Authoring, an alternative to programming which results in an educational computer program without requiring a general-purpose programming language, is discussed in depth in Chapter 14.

After the program has been developed, it must be tested thoroughly. Testing in this context refers to an evaluation of program execution, not to the ability to produce desired educational outcomes.

Lesson developers must also inform potential users of the objectives of the lesson, the prerequisite skills and knowledge required to be successful, lesson use, the computer system(s) on which the lesson will run, and any additional instructions required to implement the lesson. By developing the clear, concise documentation prior to evaluation, it becomes possible to test the effectiveness of the documentation as well as the lesson.

After verification that the program executes flawlessly and that necessary documentation has been developed, the goal is to evaluate the lesson's ability to meet the prescribed objectives. *Formative evaluation* refers to evaluation that takes place *while the program is being developed*. By allowing the evaluator to watch closely while actual members of the target population use the product, formative evaluation identifies areas of the lesson that are in need of revision, while it is still relatively simple and inexpensive to modify the lesson.

After the lesson has been revised according to the information gained during en-route evaluation, a final, or summative, evaluation validates the fact that the lesson performs as required. Several alternatives used in CAI lesson evaluation are discussed in Chapter 18.

The CAI Design Checklist

There are many steps involved in the successful design and development of a CAI lesson. The CAI Design checklist (CDC), included as Figure 5.3, prescribes the principal steps of the process.

Modify the CDC to meet your needs by adding, deleting, and rearranging steps.

Figure 5.3
The CAI design checklist (CDC.)

Phase 1. Needs Assessment

1.1. Describe the student for whom the lesson will be written.

1.1.1. Age range: 12 to 16

1.1.2. Reading level: 1 2 3 4 5 (6) 7 8 9 10 Adult

1.1.3. Other significant characteristics:

They like game formats and animated graphics.

1.2. Describe the environment in which the lesson is to be used.

1.2.1. Is supervision available to start the lesson? Y (N)

To answer questions? Y (N)

1.2.2. Would music or sound pose a problem for others? Y (N)

1.2.3. Describe the setting(s) in which the lesson will be used:

① Computer labs

② Classrooms

③ Student can take the lesson home

1.2.4. On what computer system(s) must the lesson run?

Apple II Microcomputers

1.3. Describe the constraints within which the lesson must be developed:

1.3.1. Time:

1.3.1.1. Date the project must be ready for distribution: March 15th

Figure 5.3 (*Continued*)

1.3.1.2. Date the project must be ready for summative evaluation: Feb. 1ST

1.3.1.3. Date the project must be ready for testing: Jan. 3RD

1.3.1.4. Date the design must be ready for programming: Nov. 20

1.3.1.5. Date the needs assessment must be complete: Oct. 25

1.3.2. Fiscal:

1.3.2.1. Total cost of the project not to exceed: $1,500.00

1.3.3. Resource:

1.3.3.1. How many machines will be available for development and testing? 1 for development, 10 for testing

1.3.3.2. What people will be involved in the development of the lesson, and in what capacity?

Name:	Role:
Wilma Daniels	Subject Matter Expert
Sara Meagher	Inst. Designer
Doug Archer	Programmer
Carolyn McMain	Test Coordinator
Walt Wright	Evaluator

Figure 5.3 (Continued)

1.4. List the goals of the lesson:

① The student will understand how chords are formed.

② The student will learn to name notes as they are played.

③ The student will be able to understand how scales are formed.

1.5. List the objectives of the lesson and an assessment item for each.

Obj. # Objective (Conditions/Action/Quality)

1 Given the tonic note, the student will provide the third note of the major triad within two attempts.

Assessment item: "You are building a major triad. The tonic note is the note F below middle C. Find the third note of the triad."

Obj. # Objective (Conditions/Action/Quality)

2 Given the note upon which a scale is built, the student will provide the 4th note within 2 attempts.

Assessment item: "What is the 4th note of a scale built on the note $D^{\flat}$ just above middle C?"

Figure 5.3 (*Continued*)

Obj. # Objective (Conditions / Action / Quality)

3 Given the note upon which a 7th chord is built, the student will provide the 7th within 3 attempts.

Assessment item: "Find the 7th in a 7th chord built on the note A^b below middle C."

Objectives / Assessments: Page 1 of 4

1.6. List the skills and knowledge the student must bring to the lesson in order to be successful:

Must understand:

① major triad

② tonic note

③ layout of a piano keyboard

Must read at the 6th grade level or above.

Evaluation and Revision of Phase 1.

✓ Student characteristics are adequately understood.

✓ The environment(s) in which the lesson will be used is/are understood.

✓ The constraints under which the lesson is to be developed are understood and are reasonable.

✓ The objectives of the lesson are clear, measurable, and attainable.

Figure 5.3 (*Continued*)

✓ Assessment items are congruent with the objectives.

✓ List of prerequisite skills required of the learner is complete.

✓ The resources (human, fiscal, and computer) available are appropriate for the size of the task.

Phase 2. Design

2.1. Sequencing the objectives

2.1.1. Examine the objectives listed in section 1.5 and place the objective numbers below to indicate the sequence in which they will be met.

Lesson Sequence	Objective #	Lesson Sequence	Objective #	Lesson Sequence	Objective #
1	1	18		35	
2	3	19		36	
3	4	20		37	
4	2	21		38	
5	5	22		39	
6	12	23		40	
7	11	24		41	
8	6	25		42	
9	7	26		43	
10	8	27		44	
11	10	28		45	
12	9	29		46	
13		30		47	
14		31		48	
15		32		49	

Figure 5.3 (*Continued*)

Lesson Sequence	Objective #	Lesson Sequence	Objective #	Lesson Sequence	Objective #
16	_____	33	_____	50	_____
17	_____	34	_____		

2.2. Description of Activities to Meet Objectives

✓ 2.2.1. Examine each objective, generate multiple methods of attaining that objective, and select the best.

✓ 2.2.2. Determine the method by which remedial instruction will be presented.

2.2.3. Record the decisions from steps 2.2.1 and 2.2.2 below:

Objective #: 1 Topic: 3rd note of major triad

Primary means of achieving the objective: Student will control the movement of an animated figure across a keyboard on the screen, and will stop on the correct note. Computer will play the student's triad and the correct answer.

Method in which remediation will be presented: Correct answer will be presented graphically, by highlighting appropriate keys as the notes are played. A brief explanation will follow.

2.3. Storyboarding

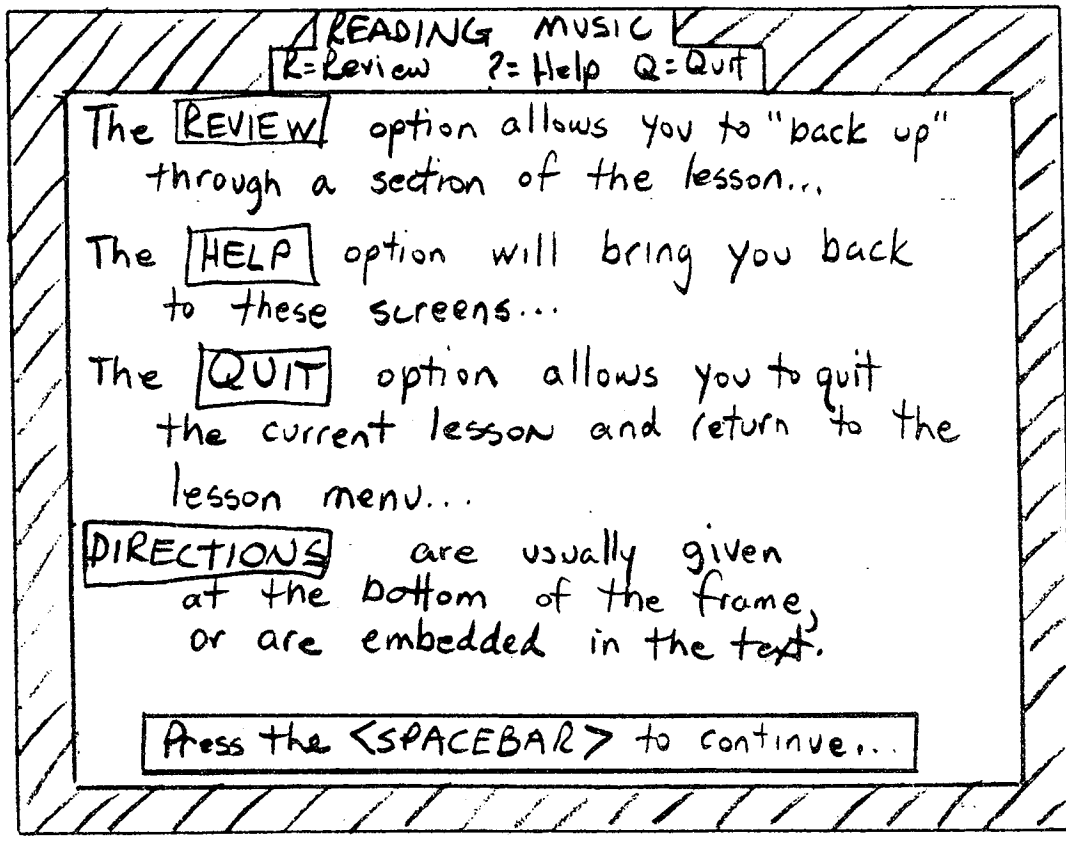
2.3.1. Using the forms provided, develop a set of storyboards for each objective.

2.3.2. Conduct a "walk-through" with team members and designated others.

Figure 5.3 (*Continued*)

Storyboard Form for Presentation Frame

Objective # 2 Board # 3 of 7



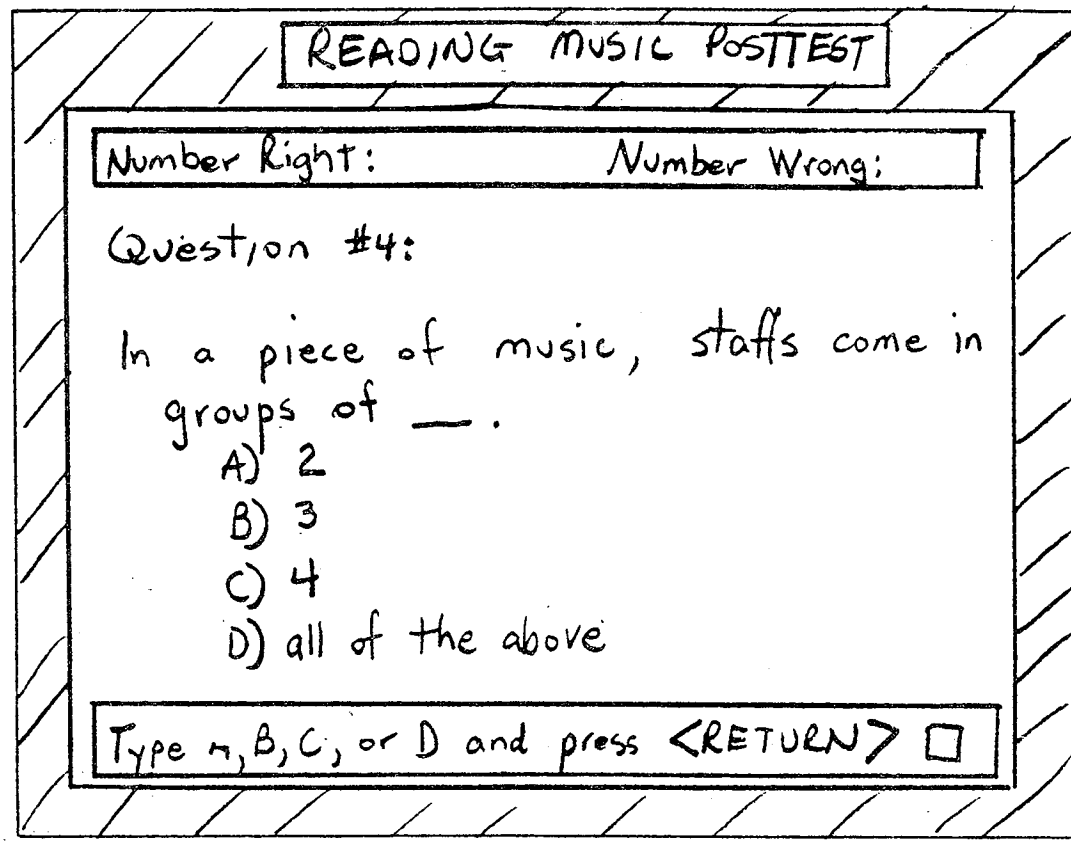
Preceding Storyboard(s): 2-2

Following Storyboard: 2-4

Action Required to Advance: <SPACEBAR> (after each option)

Comments: Display each option statement sequentially until the full screen is complete. Keywords (REVIEW, HELP, QUIT, DIRECTIONS) to be inversed for amplification effect.

Figure 5.3 (*Continued*)
Storyboard Form for Menu/Question Frame
Objective # 8 Frame # 4 of 10



Preceding storyboard(s): 8-3

Anticipated answers:	*On this answer branch to:*
A	PART CORRECT → B/C SUBS
B	PART CORRECT → A/C SUBS
C	PART CORRECT → A/B SUBS
D	CORRECT SUB → 8-5

Figure 5.3 (*Continued*)

On unanticipated answer, branch to: ILLEGAL PROMPT SUB

Record answers? Correctness: (Y) N Actual response: (Y) N

2.4. Describe performance reports to be provided:
2.4.1. Student Performance:

2.4.1.1. To the student: Immediate knowledge of correctness of response. If 90% or better, play song. No written report to the student.

2.4.1.2. To the Teacher or Supervisor: NONE

2.4.2. Lesson Performance:
2.4.2.1. To the Teacher or Supervisor: NONE

Figure 5.3 (*Continued*)

2.4.2.2. To the lesson's Designer(s) Track the number of responses each student makes for each item.

✓ 2.5. Make another pass through the storyboards modifying them to increase interactivity and individualization.

Evaluation and Revision of Phase 2.

✓ Sequence of objectives is appropriate (progresses from easy to more difficult, no undefined terms or necessary concepts covered later than needed, etc.).

✓ Activities listed for primary instruction and for remediation for each objective seem adequate to meet instructional needs.

✓ Storyboards communicate the designer's intent unambiguously so that the programmer will be able to faithfully execute the designer's plan.

✓ Requirements for performance tracking are conveyed clearly.

✓ Lesson involves the learner by requiring meaningful interactions as opposed to a series of automatic or contrived responses.

✓ The lesson adapts to the needs of individual students.

— The student is made aware of the lesson's objectives. (needs work)

✓ The student is made aware of prerequisite skills required.

✓ The lesson provides adequate remediation and re-teaching.

✓ The lesson provides an adequate amount of learner control.

✓ The lesson will maintain the student's interest.

✓ The instructions are clear.

✓ The actions required of the student are unambiguous and easy to understand.

✓ The student is made aware of the correctness of each response.

Figure 5.3 (*Continued*)

✓ Positive feedback is not overdone.

✓ Negative feedback is informative but not punitive.

✓ Feedback is varied.

✓ Feedback motivates the student to respond correctly.

✓ The lesson does not require an inordinate amount of Teacher/Supervisor involvement.

✓ Test items match instructional objectives.

✓ Questions are unambiguous and do not use unfamiliar terminology.

✓ Answer judging is thorough. Correct alternatives to the most likely answer are considered correct, anticipated incorrect answers are considered incorrect, and the student is asked to redo unanticipated answers.

✓ Responses required do not interfere with the student's ability to answer. For example, inability to type is not misinterpreted as inability to generate the correct answer.

✓ The lesson uses graphics appropriately to eliminate overuse of text.

✓ Highlighting is used to direct the student's attention

✓ The lesson provides adequate opportunity to practice.

Additional Comments on Phase 2 Work:

Nice design! Good collaboration between subject matter expert and instructional designer.

Phase 3. Development and Implementation

✓ 3.1. Flowcharting

✓ 3.1.1. Develop the flowchart.

✓ 3.1.2. Programmer and Designer review the flowchart.

✓ 3.2. Determine the programming language or authoring system or language which will be used to produce the lesson. (Remember that the computer system on which the lesson is to run and cost constraints should be major considerations in this decision.)

Figure 5.3 (*Continued*)

✓ 3.2.1. Which programming language or authoring system or language will be used? Super PILOT

3.3. Set a date for preliminary review of first module of the lesson, to take place on the computer system for which the lesson is being developed. (12/15)

✓ 3.4. Produce first module.

✓ 3.5. Review first module to verify that designer's intentions are being implemented by the programmer, and that any questions are answered before the entire lesson is done incorrectly.

✓ 3.6. Make indicated revisions to the first module. (done 12/20)

✓ 3.7. Produce remaining modules.

✓ 3.8. Test program to assure flawless program logic.

✓ 3.9. Designer takes entire lesson to validate that the plan was implemented as intended.

✓ 3.10. One to one formative evaluation (See Chapter 17 for more detail on evaluating CAI)

✓ 3.10.1. Evaluate

✓ 3.10.2. Report results

✓ 3.10.3. Identify problems

✓ 3.11. Revise as indicated

✓ 3.12. Small group evaluation

✓ 3.12.1. Evaluate

✓ 3.12.2. Report results

✓ 3.12.3. Identify problems

✓ 3.13. Revise as indicated

✓ 3.14. Field test

✓ 3.14.1. Evaluate

✓ 3.14.2. Report results

✓ 3.14.3. Identify problems

Figure 5.3 (*Continued*)

✓ 3.15. Revise as indicated

✓ 3.16. Revise the flowchart to reflect any changes made since it was first developed

✓ 3.17. Revise the storyboards to reflect any changes

✓ 3.18. Develop appropriate documentation to accompany the lesson.

- ✓ 3.18.1. Teacher/Supervisor level documentation
- ✓ 3.18.2. Student level documentation
- ✓ 3.18.3. Programmer level documentation (for maintenance purposes)

Evaluation and Revision of Phase 3.

✓ The lesson design has been implemented faithfully. (as modified)

✓ The lesson has been thoroughly tested and has been proven to run adequately on the desired computer system(s) and to be free from errors of program logic.

✓ The documentation accompanying the lesson provides adequate information for the Teacher/Supervisor to determine appropriate uses for the program.

✓ The instructions allow the student to use the lesson with minimal or no Teacher/Supervisor involvement.

✓ The lesson has been adequately evaluated and meets its objectives.

✓ Students like the lesson.

Individual Development vs. the Development Team

There has been considerable discussion concerning the benefits associated with a team approach to the development of CAI. Teams generally consist of one or more instructional designers, content experts, programmers, and evaluators. Proponents of development teams (Applebaum, 1985; Bork, 1984; Dimas, 1978; Hartman, 1981; Roblyer, 1983) believe that a team approach to CAI design offers the ability to divide responsibility, resulting in shorter development times and reduced individual burdens. The use of experts in each aspect of product development provides a system of checks and balances, which tends to yield better products.

Disadvantages have also been associated with a team approach. Such disadvantages have included higher development costs, a lack of "ownership" felt by team members, divergence of opinion, and quality dragged down by the weakest

member of the team. There may also be communication problems among different team members, as well as scheduling complications.

Since many of the disadvantages may be avoided, the advantages of a team approach often outweigh the disadvantages. Few people are qualified to assume multiple roles, and even fewer have the time to do so. A team approach capitalizes on the strengths of the team members, and the momentum provided by the team can maintain progress on projects that might have faltered due to waning interest or energy.

In instances where local resources preclude the use of a development team, consider involving another person to evaluate decisions made during each phase. Even an evaluator with no knowledge of CAI design but an understanding of the educational process can identify shortcomings and provide helpful suggestions to improve the lesson and reduce time-consuming revisions. If it is impossible to assemble a team or even an additional evaluator, the quality of CAI products can be improved if you, as the sole designer, consciously perform the functions normally assigned to different experts to the best of your ability.

Chapter Summary

According to Roblyer (1981, p. 174), "Even now, when it is widely recognized that programming expertise alone is not enough in courseware development, there still appears to be an emphasis on 'authoring CAI programs' rather than on 'designing instructional courseware.'" If computers are to have significant impact on education, the quality of available courseware must improve. Use of an instructional system design model can help to improve the quality of the CAI produced.

This chapter described instructional design systems as a series of interrelated steps based on principles derived from educational research and theory. By following these steps, the designer can be more confident that the resulting instruction will be appropriate for the students, environments, and computer systems for which it was designed.

The CAI design model (CDM) classified the steps of instructional design systems into four phases: needs assessment, design, development and implementation, and evaluation and revision. Adherence to the CAI design checklist (CDC), a set of activities that proceeds from planning through design and development, can also improve CAI lessons. It is suggested that this tool be modified to meet your needs by adding items, deleting items, and modifying the sequence of items as appropriate.

For some time, CAI authorities have warned that more attention must be paid to instructional design (Hazen, 1985; Kearsley, 1984; Leiblum, 1984; Splittgerber, 1979). CAI will be successful or will fail depending on the lesson design. Devote the effort required to ensure success.

References

Applebaum, W. R. (1985). Course-centered development: A team approach to CBT. *Data Training*, 4(4) 26–27.

Bork, A. (1984). Producing computer-based learning material at the educational technology center. *Journal of Computer-Based Instruction*, 11(3), 78–81.