Research Product 89-10

Selection of a Computer-Based Training Authoring System: Functional Requirements and Evaluation Criteria

February 1989

Automated Instructional Systems Technical Area
Training Research Laboratory

U.S. Army Research Institute for the Behavioral and Social Sciences

U.S. ARMY RESEARCH INSTITUTE FOR THE BEHAVIORAL AND SOCIAL SCIENCES

A Field Operating Agency Under the Jurisdiction of the Deputy Chief of Staff for Personnel

EDGAR M. JOHNSON
Technical Director

JON W. BLADES
COL, IN
Commanding

Technical review by

Judith E. Brooks
Beatrice J. Farr

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

ADA207141

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS --
2a. SECURITY CLASSIFICATION AUTHORITY --	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE --	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) ARI Research Product 89-10	5. MONITORING ORGANIZATION REPORT NUMBER(S) --
6a. NAME OF PERFORMING ORGANIZATION U.S. Army Research Institute for the Behavioral and Social Sciences 6b. OFFICE SYMBOL (If applicable) PERI-II	7a. NAME OF MONITORING ORGANIZATION --
6c. ADDRESS (City, State, and ZIP Code) 5001 Eisenhower Avenue Alexandria, VA 22333-5600	7b. ADDRESS (City, State, and ZIP Code) --
8a. NAME OF FUNDING/SPONSORING ORGANIZATION -- 8b. OFFICE SYMBOL (If applicable) --	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER --
8c. ADDRESS (City, State, and ZIP Code) --	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 62722 / PROJECT NO. A791 / TASK NO. 4.4.3 / WORK UNIT ACCESSION NO. H.2

11. TITLE (Include Security Classification)
Selection of a Computer-Based Training Authoring System: Functional Requirements and Evaluation Criteria

12. PERSONAL AUTHOR(S)
Park, Ok-choon, and Seidel, Robert J.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM 88/01 TO 89/01	1989, February	43

16. SUPPLEMENTARY NOTATION
--

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Computer-based training Computer-based training authoring system

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

A computer-based training (CBT) authoring system is a special kind of software tool kit for training developers to use in CBT materials. Since the late 1970s, many CBT authoring systems have been developed that serve the same purpose but have different theoretical approaches, functional features, and power. Thus, selection of an authoring system requires a systematic evaluation. This paper proposes a set of evaluation criteria for selecting an authoring system on the basis of its functional capabilities.

A CBT authoring system can be evaluated according to the following criteria:

1. its functional capability to provide computer programming aids for a training developer with minimal or no programming skills;

(Continued)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code)	22c. OFFICE SYMBOL
Ok-choon Park	(202) 274-8827	PERI-II

DD Form 1473, JUN 86 *Previous editions are obsolete.*

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

ARI Research Product 89-10

19. ABSTRACT (Continued)

2. the quality of instructional design guidance available for the training developer with limited experience in instructional design and training development;

3. the system's flexibility to expand its functional capability through interface with other software tools and hardware peripherals;

4. the ease with which the system can be learned and used; and

5. the cost, including price, hardware requirements, contract conditions, and training expenses.

KR)

Although the five criteria listed above provide basic information for evaluating an authoring system, other factors unique to the given situation should be considered in the selection of a system. Other important factors include training and experience of the developers; characteristics of the subject domains; goals and objectives of the CBT programs; existing hardware and software; expected period and frequency for using the system. Also it should be noted that the five evaluation criteria are not completely independent of each other. Ultimately, the relative weights of the various criteria are based on the subjective judgment of the training developers and managers. To aid this judgment, we propose a combined evaluation approach of checklist and benchmarking.

Research Product 89-10

Selection of a Computer-Based Training Authoring System: Functional Requirements and Evaluation Criteria

Ok-choon Park and Robert J. Seidel

Automated Instructional Systems Technical Area
Robert J. Seidel, Chief

Training Research Laboratory
Jack H. Hiller, Director

U.S. Army Research Institute for the Behavioral and Social Sciences
5001 Eisenhower Avenue, Alexandria, Virginia 22333-5600

Office, Deputy Chief of Staff for Personnel
Department of the Army

February 1989

Army Project Number
2Q162722A791

Automated Training
Development

FOREWORD

The computer has become a primary vehicle for training in the U.S. Army. The success of computer-based training (CBT) depends on the effectiveness of training materials delivered by the computer and the efficiency of development of these materials. To support the need for the development of effective CBT materials in efficient ways, many CBT authoring systems have been developed.

This paper gives Army training managers and developers comprehensive guidelines for understanding and evaluating the important functional requirements of a CBT authoring system and selecting a system that will satisfy U.S. Army training requirements. This paper is based on Dr. Robert J. Seidel's information briefing on CBT authoring systems as presented to the Deputy Commanding General for Training, TRADOC, Lieutenant General John S. Crosby on November 20, 1987.

EDGAR M. JOHNSON
Technical Director

SELECTION OF A COMPUTER-BASED TRAINING AUTHORING SYSTEM: FUNCTIONAL REQUIREMENTS AND EVALUATION CRITERIA

EXECUTIVE SUMMARY

A computer-based training (CBT) authoring system is a special kind of software tool kit for training developers to use in CBT materials. Since the late 1970s, many CBT authoring systems have been developed that serve the same purpose but have different theoretical approaches, functional features, and power. Thus, selection of an authoring system involves a decision regarding the optimal combination of the system's development power, efficiency, and costs for producing high volumes of training materials. The purpose of this paper is to define the functional requirements of a CBT authoring system and propose a set of evaluation criteria for selecting an authoring system.

The first evaluation criterion of an authoring system is its functional capability to provide computer programming aids for a training developer with minimal or no programming skills. Even if the training developer has programming skills, the specific development aids built into the authoring system are usually more efficient than a programming language.

The second criterion is the quality of instructional design guidance available for the training developer with limited experience in instructional design and training development. The important instructional design guides include: (a) guides for the layout of lesson structure; (b) aids for selecting instructional strategies; (c) provision of templates for developing lesson components; and (d) aids for collecting and managing data for instructional decisions.

The third criterion is the system's flexibility to expand its functional capability through interfaces with other software tools and hardware peripherals. The important software tools and hardware peripherals with which an authoring system needs to interface include: (a) existing development tools (e.g., word processor, graphic editor, etc.); (b) programming languages; (c) built-in utilities in the computer operating system (e.g., calculator, clock, etc.); (d) training management systems (although some CMI features are integral to an authoring system); (e) hardware peripherals (e.g., printer, interactive video, etc.); (f) an interoperating system among independent hardware systems, if any; (g) on-line communication and networking system; and (h) various types of input devices.

The fourth criterion is its ease of learning and use. Important sub-criteria for evaluating the ease of learning and use include: (a) flexibility to allow a variety of instructional design and development approaches; (b) effort required for developing and revising courseware components; (c) provision of built-in development templates; (d) effort required to integrate the developed courseware components into a structure; (e) adaptation to the author's skill level; (f) direct observation of or development from the student's view; (g) 'help' features built into the system; and (h) an adequate user manual.

The last main criterion for evaluating an authoring system is the cost, including price, hardware requirements, contract conditions, and training expenses.

Although the five criteria listed above provide basic information for evaluating an authoring system, other factors unique to the given situation should be considered in the selection of an authoring system. Other important factors include training and experience of the users; general characteristics of the subject domains; goals and objectives of the CBT programs; existing hardware and software; and expected period and frequency for using the system. Also, it should be noted that the five evaluation criteria are not completely independent of each other. Ultimately, the relative weights of the various criteria are based on the subjective judgment of the training developers and managers. To aid this judgment, we propose a combined evaluation approach of checklist and benchmarking.

SELECTION OF A COMPUTER-BASED TRAINING AUTHORING SYSTEM: FUNCTIONAL REQUIREMENTS AND EVALUATION CRITERIA

CONTENTS

LIST OF FIGURES

CONTENTS (Continued)

SELECTION OF A COMPUTER-BASED TRAINING AUTHORING SYSTEM: FUNCTIONAL REQUIREMENTS AND EVALUATION CRITERIA

INTRODUCTION

As computer technology is undergoing rapid development, information-handling requirements are reaching a broader spectrum of potential users across occupations in all facets of society. Most of today's computer users do not need, nor have the time, to develop programming skills. What they need is to select and use the most appropriate software tool for their information applications. These recently developed software tools have taken into account the needs of computer-naive users so that the learning requirements for using the tools is supposed to be minimal. However, the selection of the best software tool for the given purpose requires a systematic evaluation because many software tools have been developed to serve basically the same purpose with different approaches, functional features and power.

A computer-based training (CBT) authoring system is a special purpose tool-kit for training developers with no programming skills and limited instructional design experience to use in developing CBT materials. The intent of this paper is to: (a) define the functional requirements of a CBT authoring system; (b) propose evaluation criteria for selecting an authoring system on the basis of the functional requirements. The selection of a good authoring system is one of the critical factors for the success of CBT development projects. Avner, Smith and Tenczar (1984) concluded from the analysis of CBT materials produced by 143 development groups that the use of an appropriate authoring tool and the developer's experience are the most important factors in predicting the quality of the CBT materials and the efficiency of the development process.

Locatis and Carr (1985) proposed a comprehensive checklist for selecting an authoring system. However, this checklist has three basic problems. First, questions in the checklist are too abstract and ambiguous (e.g., "Does the system provide adequate levels of authoring flexibility?"). It is not clear what "authoring flexibility" means. It could mean a flexibility to develop various formats of CBT materials (e.g., tutorial, games, simulations, etc.), or to adopt various approaches to instructional principles and strategies, or to interface with other software development tools, or to edit and

modify developed materials. Many questions in the checklist have similar problems and are difficult to use as a comparative criterion for selecting a system among many alternatives. Second, the checklist ignores some important factors that should be evaluated (e.g., instructional design guidance). Third, the checklist asks questions, mostly in a dichotomous form, about a single dimension of the authoring feature (i.e., is a given function present or not?). Other dimensions are ignored such as ease of using the specific function, level of the functional sophistication, etc.

Another approach to conducting a comparative evaluation of authoring systems is to develop benchmark lessons using the systems (Hillelsohn, 1984). An advantage of the benchmark approach is the ability to obtain a variety of empirical evaluation measures of the systems (e.g., development time, difficulty of learning and using the system, quality of the developed materials, etc.). However, the benchmark approach also has several problems: (a) it is not feasible to expand the benchmark to include all types of generic instructional forms and to evaluate every important feature of the system because most lessons to be developed using the systems are likely to require only limited system features (for example, most benchmark materials developed for evaluation are segmental portions of a lesson and the function for integrating them into a complete lesson structure is frequently ignored); (b) evaluation results are influenced by the similarity and difference between the evaluator's prior experience (including instructional design orientation) and the instructional development procedures built into the system; (c) the evaluator's experience in one system influences his/her use and evaluation of other systems.

Since each of the evaluation approaches falls short by itself, we think a benchmark approach and a functional feature-comparison approach (i.e., a checklist) should be used together to complement each other. However, a simple combination or simultaneous use of the two methods does not automatically eliminate the problems inherent in the methods. Functional features to be included in the evaluation checklist and benchmark lesson-development should be selected carefully not only to minimize the limitations of each method, but also to complement the shortcomings of the other method as much as possible. The functional requirements and selection criteria proposed in this paper will provide basic information to develop a complete set of evaluation questions for comparing important features among alternative authoring systems

and selecting important lesson characteristics that should be required in the development of benchmark evaluation lessons.

STAGES OF TRAINING DEVELOPMENT AND THE ROLE OF AUTHORING TOOLS

Two primary types of functions that an authoring system should provide are: (a) programming aids for computer-naive training developers, and (b) instructional design aids for developers with limited experience in the field. Figure 1 is a simple example for comparing authoring requirements in the use of an authoring system and a computer programming language; it also shows simple instructional design aids that an authoring system can provide (see "computer guided" column in the Figure). Specific features of an authoring system that are necessary to perform the two basic functions are discussed in the following section: Functional Requirements of An Authoring System.

We think a complete CBT authoring system should provide aids for the training developer through the entire process of a systems approach to training development. Therefore, we briefly review the process of training development before discussing specific functional features of an authoring system. As Figure 2 shows, the first stage of training development is to determine new training needs from the analysis of employees' (potential trainees) job performance, new task requirements imposed by adaptation of new technology, and other situational changes (e.g., changes of missions and/or doctrines in the military). The training needs identified from the analysis may include a choice between the development of new training programs or the modification of existing programs (including delivery systems).

The second stage is to analyze specific requirements for developing new training programs or modifying existing programs. The requirement analysis of training development includes: analysis of job/tasks, determination of job performance criteria, analysis of trainee characteristics, and analysis of situational variables including time, budget, personnel, facility, etc. From this requirement analysis, the most appropriate delivery system will be selected for the given situation. If CBT is selected, a determination should be made whether to use it as the sole delivery system, a primary delivery system with others (e.g., classroom instruction,

Figure 1. Use of An Authoring System: An Example

AUTHORING SYSTEM

Computer Guided	Author-written
What is the name of this lesson?	Firing Position of Missile Launcher of ITV
Question #1	What position must the Turret Launcher be into fire TOW missiles?
Correct Answers?	Erect position
Feedback for correct responses?	That's right. The turret launcher must be in erect position before arming the TOW missile.
Number of trials?	2
Expected wrong answers?	Normal position
Feedback for second wrong answer?	No. It should be in erect position.
Do you have another question? (Y/N)	Y

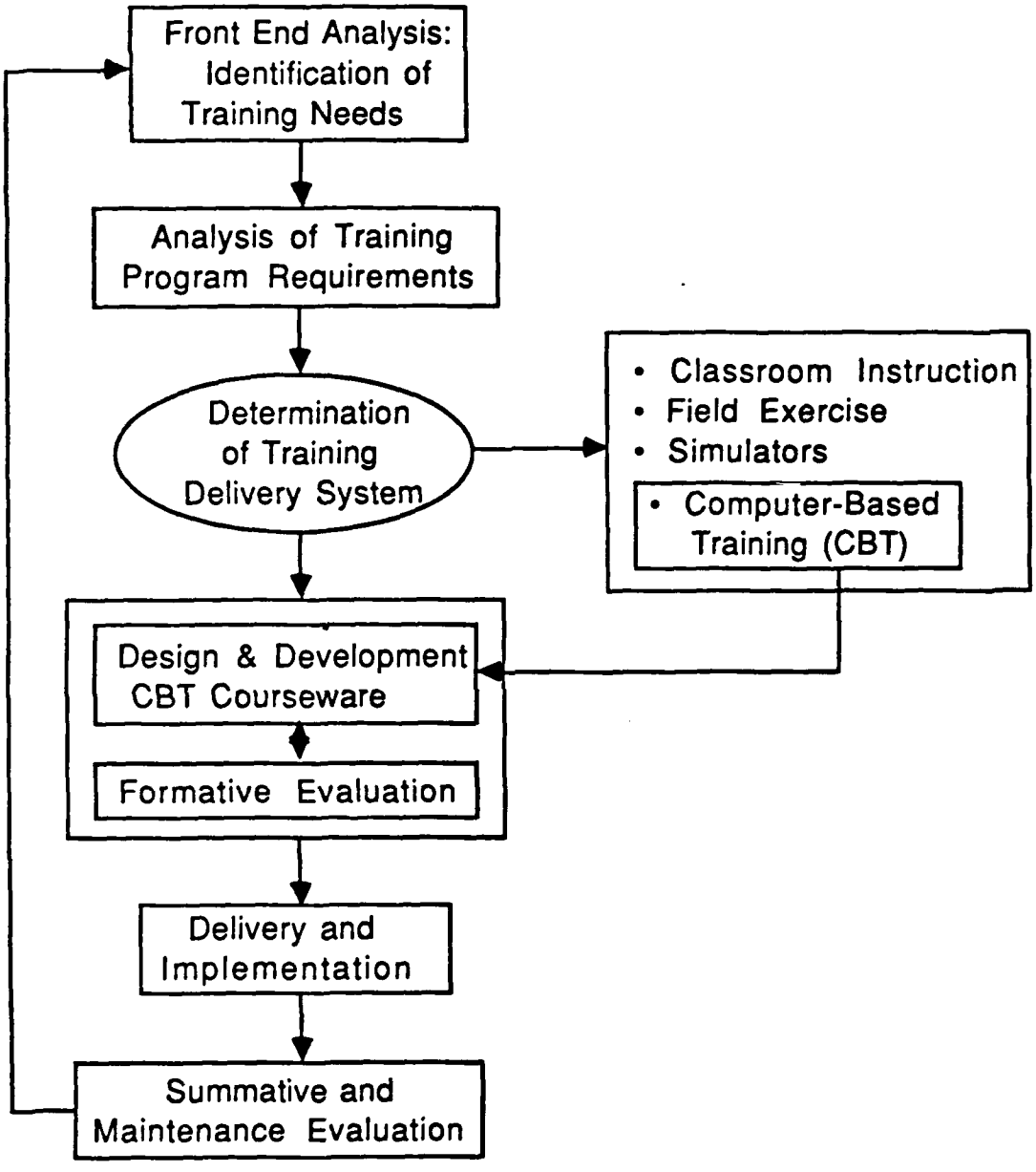
PROGRAMMING LANGUAGE (BASIC)

Author-written
(No Computer Guidance)

```
100 HOME
150 PRINT "Today we'll learn about "Firing
      Position of Missile Launcher of ITV";
200 FOR T=1 TO 1860:NEXT T:REM 2 SECOND
    TIME DELAY
250 CORRECT=CORRECT+1
300 REM FIRST QUESTION
350 HOME
400 PRINT "What position must the Turret
      Launcher be into fire TOW missiles?"
450 INPUT ANSWER$
500 READ WRD$
550 IF WRD$=ANSWER$ THEN GOSUB 2000:
    GOTO 750: REM CORRECT FEEDBACK
600 IF WRD$="0" THEN GOSUB 3000:
    GOTO 750: REM INCORRECT FEEDBACK
650 GOTO 500
700 DATA ERECT POSITION, NORMAL
      POSITION, 0
750 HOME
800 PRINT "Please press RETURN for more
      information about "Firing Position of
      Missile Launcher of ITV.";
850 END
2000 PRINT "That's right. The turret
      launcher must be in erect position
      before arming the TOW missile.";

2100 FOR T=1 TO 1860: NEXT
2200 RETURN
3000 PRINT "No. It should be in erect
      position.";
3100 FOR T=1 TO 3720:NEXT T
3200 RETURN
```

Figure 2. Process of Training Development



field exercises, simulator-based training, embedded training, etc.), or as a supplement for other primary delivery systems. It should be noted that, whatever the intended use of CBT is, the computer-based individualized training must be integrated with existing group-based training by adopting a new training management procedure. Once CBT is selected as a primary or supplemental delivery system, training programs to be delivered by the computer must be developed.

Finally, after a series of formative evaluations for the developed programs (including content validation, technical debugs, and small group try-outs), the programs will be implemented to achieve the intended training objectives and goals.

A complete authoring system should provide various types of aids to help the training developer perform the required tasks in each stage of the training development process reviewed above. However, it is difficult in a single system to build all the aids necessary for the required activities in every stage of the process because quite different types of tasks are required for the different stages. If all aids were built in, the system would become too complex and sophisticated for most training developers to learn and use. Therefore, in our view, the best approach is to develop an independent tool for each of the stages and to allow them to interface with each other in the operating hardware system. Figure 3 shows some computer-based tools, for the different stages of the training development process, which are under development or proposed to be developed at the U. S. Army Research Institute (ARI) and other U.S. Army training agencies.

For the problem and need analysis (i.e., front-end analysis) stage, ARI and the Army Training and Doctrine Command (TRADOC) have just begun to develop a relational data management system to concurrently automate the management (including input, editing, and retrieval) of various types of information in different data bases. This relational data management function is performed by inter-connecting the data bases into a network structure. The output to be produced by this tool will be used for: (a) determining the need for new training programs and (b) generating training materials such as soldier manuals. We expect this tool will be expanded as a system to automate the entire process of the Systems Approach to Training (SAT) development by integrating various software tools developed for the different stages of training development.

Figure 3. Sample Systems for Automating Training Development

TRAINING PROCESS	SAMPLE SYSTEMS:UNDER DEVELOPMENT/PROPOSED
1. FRONT END ANALYSIS	• AUTOMATED SYSTEMS APPROACH TO TRAINING (ASAT) • MISSION FUNCTION TASK DATA BASE
2. ANALYSIS OF TRAINING REQUIREMENTS	• MEDIA SELECTION AID • TRAINING STRATEGY SELECTION AID
3. DESIGN	• INSTRUCTIONAL DESIGN EXPERT SYSTEM
4. DEVELOPMENT	• ELECTRONIC INFORMATION DELIVERY SYSTEM (EIDS) ASSIST • COMPUTER-BASED INSTRUCTIONAL RESEARCH SYSTEM (CBIRS) • OTHER COMMERCIAL SYSTEMS
5. DELIVERY	• EIDS & OTHER HARDWARE SYSTEMS
6. MANAGEMENT	• COMMERCIAL CMI SYSTEMS • ARMY INFORMATION MANAGEMENT SYSTEM (AIMS) • INTEGRATED TRAINING MANAGEMENT SYSTEM (ITMS)

For the stage of training requirement analysis, ARI has initiated development of a decision support system for selecting training delivery media. An instructional design expert system is under development by ARI to provide training developers specific aids for the design of a training program, including the analysis and structure of the given subject content and the selection of specific instructional strategies. A number of authoring tools are already available to support training development activities in the Army's standard CBT hardware system, EIDS (Electronic Information Delivery System). Currently, an effort is underway to establish the standard features of the Army CBT authoring system. The Army at present does not have a standard computer-managed instructional (CMI) system to support EIDS, although other CMI systems (e.g., PLATO Learning Management System, On-Line Authoring System) are used to support CBT facilities at several Army Schools. The U.S. Army has two training data management systems to utilize information collected during various types of training (including CBT) in the process of making training management decisions and policies: (a) the Army Information Management System (AIMS); and (b) Integrated Training Management System (ITMS).

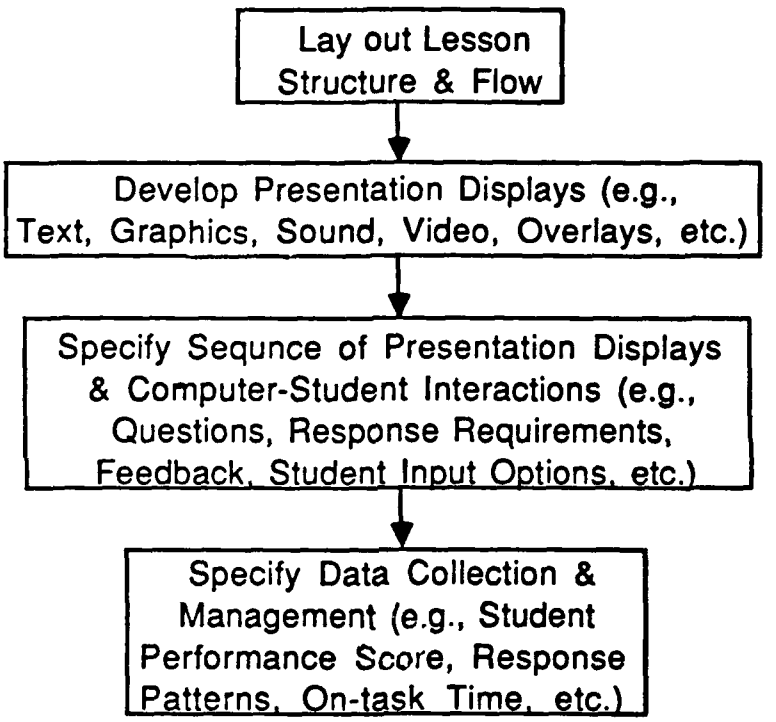
FUNCTIONAL REQUIREMENTS OF AN AUTHORING SYSTEM

As discussed above, CBT design and development requires the entire range of activities involved in the SAT process. Thus, an authoring system in a broad sense includes the various types of automated tools to support activities required in the different stages of the training development process. However, the definition of an authoring system in a narrow sense is usually limited to a specific type of automated tool directly used for the development of CBT materials. In this paper, we discuss the functional requirements of an authoring system in the narrow definition. After discussing the functional requirements, we present a set of criteria for selecting an authoring system.

Figure 4 depicts a simplified process of CBT courseware development. Courseware to be developed is usually specified in detail in a design document. Thus, most authoring systems mainly provide programming aids to develop the courseware as specified in the design document. However, the design document prepared in advance is usually not specific enough to develop the detailed

characteristics of courseware. In addition, the document itself needs frequent revisions during the development process. Therefore, as detailed below, an authoring system needs to provide instructional design aids as well as programming aids. Instructional esign aids include the organization of lesson structure, plus the selection and development of specific instructional strategies.

Figure 4. Development of Computer-Based Instructional Courseware



Lay-out of Lesson Structure

The first stage of courseware development is to lay out the lesson components (e.g., presentation displays such as text and graphics, interaction procedures such as questions and feedback, etc.) into a lesson structure. A typical aid for this stage is to provide a structured framework in which representations (e.g., titles, icons, etc.) of various lesson components (e.g., presentation displays, interactions, etc.) can be arranged. For example, a set of symbolic icons representing different types of lesson components can be provided for the author to arrange in a lesson structure. A

template associated with each of the representations is provided to support specific development activities for the lesson component. The author can request the development template at any time for repeated uses whenever needed in the development process without creating the same template format again. The selection and organization of the representations into a lesson structure should be flexible enough for the developer to do the following activities:

- design single or combined types of instructional formats (e.g., tutorial, drill and practice, games, simulations, modeling, etc.) for any domain content;

- design any size lesson or curriculum by allowing the organization of the representations into unlimited levels of a hierarchical structure;

- reorganize the lesson structure by simple rearrangements of the representations;

- develop or revise a lesson component (i.e., presentation materials and procedures) corresponding to the selected representation at any time.

For example, "Course of Action" developed by Authorware, Inc., has eight icons for representing different lesson components. The lesson structure organized with the icons can be easily edited by using standard Macintosh editing procedures (i.e., selection of simple menu-based options such as 'select', 'insert', 'delete', 'cut', 'copy', 'paste', etc.). If the lesson structure organized with icons is changed, the structural order of lesson components containing presentation content and procedures are automatically reorganized to be consistent with the new lesson structure. The system allows organization of the lesson into a hierarchical structure with an unlimited number of levels.

Selection of Instructional Strategies

Once the lesson structure is laid out, specific contents and instructional strategies for the lesson components (e.g., presentation displays such as text and graphics, interaction procedures such as questions and feedback, etc.) should be selected and developed. As Table 1 shows, instructional strategies range from preinstructional strategy to post-instructional strategy, and

from the selection of presentation stimuli (e.g., text, graphic, sound, etc.) to the determination of interaction conditions. It is important that the authoring system has a built-in capability that allows the author to select specific instructional strategies and to develop the selected strategies by use of preformatted templates. For example, a number of options for selecting a question type (e.g., true false, matching, multiple choice, short answer, etc.) can be presented to the author in the form of a menu. If the author selects a question type, the system provides appropriate templates for developing the selected type of questions. These aids for instructional strategy selection and development are particularly helpful for courseware developers with limited instructional design experience. This is especially true in the military because of the rapid personnel turnover.

Development of Lesson Components: Presentation Displays

The third stage of CBT courseware development is to produce specific lesson components. The lesson components can be divided into two main categories: (a) presentation displays and (b) student-computer interaction algorithms. Because the interaction algorithms involve decisions for sequential presentations of display components, the presentation displays are usually selected and developed first.

The presentation displays consist of various components such as text, graphics (including animation), sound, video, and overlays of the components. Authoring aids are needed to develop these display components without using computer-programming skills; even if programming skills are available, it should be more efficient to develop the courseware materials using authoring aids. The following is a list of the specific aids for developing presentation displays.

- A text editor is needed to write and edit the text content. An authoring system should have various types and sizes of fonts and characters; it should allow the author to highlight (e.g., underline, blinking, etc.) and edit the text. Typical options in a text editor of an authoring system are usually very similar to those of a word-processor (e.g., insert, delete, cut, paste, indent, copy, etc.).

- A graphics editor is required to develop and edit graphical

presentations of instructional information. The graphic editing function of most authoring systems is limited to simple drawings such as line, box and circle. It is difficult to build a full capability for developing various types of sophisticated graphics (including high-resolution graphics and animations) into a single authoring system which should have many other different capabilities and features. Even if such a graphic development capability could be built into an authoring system, the system might be too complex for the computer-naive author to completely understand and use. Thus, a capability to transport into the lesson structure graphics (including animations) developed using other tools is an important feature that an authoring system should have (a complete capability description of graphic tools is beyond the scope of this paper). Another graphic-development function is the ability to digitize, save and edit visual images of pictures and real objects. Because of the unique technical requirements, this function also may be performed through the system's interface with a special tool (e.g., Thunderscan for the Macintosh). The function of combining computer generated displays (including text and graphics), video images, and digitized images into a single presentation screen is usually performed through the interface with interactive videodisc. Color is another important element for graphics, which can also be used for other types of displays (including text).

- A sound generator/editor is needed if the author wants to include sound presentations (e.g., signals, voice, music, etc.) as a part of the instructional components or strategies. The computer's sound editing capability is usually limited to the use of system-provided sounds (i.e., recorded analog with or without video), although some systems allow the user to digitize specific sounds. The sound generation and editing functions are usually performed through the system's interface capability with an interactive video disc (see next section) or a specific software tool designed solely for sound recognition, generation, retrieval and edition. Again, it should be noted that there are technical complexity and practical limitations against including all the capabilities required for the authoring process in a single authoring system.

- A video production aid has become an important feature of an authoring system, especially since interactive video is used as an integral part of many CBT systems (refer to "EIDS-ASSIST" by Computer Science Corporation). Although the production of a video lesson requires a completely different expertise and process from those of a computer-based lesson, a CBT authoring system may provide some aids to facilitate the production of video portions of CBT materials. The video production aid helps in developing storyboards with the system's text and graphic editors and in scheduling video shots with an automatic sorting function. (However, video production management aids are not discussed in detail in this paper because they are not functions directly required for CBT authoring.). An authoring system's capability to allow the author to specify the presentation of video frames during the implementation of CBT courseware is discussed in a later section (see "Interface with Videodisc").

Development of Lesson Components: Specification of Computer-Student Interaction Procedures

The author should easily be able to specify the presentation sequence of the developed displays (and other information, if any). Although the general order of the display presentations is specified in the lay-out stage of the lesson structure (stage 1), the detailed interactions between the computer and student are specified at this stage. The characteristics of interactions are generally selected on the basis of pedagogical decisions for questioning, evaluating student responses, and feedback. Another factor influencing the interaction characteristics is the type of input device used by the student.

- Types of questions and required responses are one of the primary factors for determining the interaction process. The frequently used types of questions in CBT include the following: true or false; multiple choice with single or multiple answers; matching with single or multiple items in single or multiple sets of items; short answers. The student may or may not be required to provide overt responses to the given questions. Research issues related to the selective use of question types, although important, are not a concern of this paper.

- Number and presentation order of questions are specified to determine the interaction algorithms. For example, the developer should specify how many questions are to be presented in what order (e.g., random, sequential, conditional including response-sensitive, etc.); also, the developer may need to specify whether the presented questions should be replaced in the question pool for possible re-presentations.

- Number of trials allowed for the student to answer correctly is another variable for determining the student and computer interactions. For example, the interaction (including feedback) in a case where only one trial is allowed will be different from a case where multiple trials are allowed.

- Feedback is another important instructional variable determining the interaction process. First, depending upon the type of information to be provided in feedback (e.g., simple knowledge of results - correct or incorrect-; analytical explanations; algorithmic step-by-step feedback), the nature of the interaction will be different. Second, time to provide the feedback (e.g., immediate or delayed; if delayed, when) also determines the interaction process. The pedagogical rationale for the selection of specific feedback type and time could be provided to the author via a "HELP" function in the system, when it is necessary. The author should be able to specify conditions for determining the sequence of presentation displays and the interaction procedures on the basis of the values of many different variables (e.g., test score, mastery level, response type, prior performance, aptitude information, etc.).

- Student Input Device should be selected after the interaction processes are specified on the basis of pedagogical decisions described above. Although the keyboard is a main input device for most CBT training, mouse, light-pen, joystick, touch-screen, etc. can be used as well. These tools can be used for the student to point to a specific location in the screen display and to move a part of or the entire screen display to a certain location. The author should specify what input devices the student should use for their input, including responses. It should be noted that these input devices should be available in the authoring process as well.

Specification of Data Collection and Management

The final stage of CBT courseware development is to specify what kinds of data should be collected during the instruction and how the data should be used and managed. The data are collected for two general purposes: (a) instructional diagnosis and (b) training management decisions. Instructional diagnosis can be made on a specific learning problem (e.g., a specific question or a portion of a problem-solving procedure) in a given content and on a curriculum structure level (e.g., instructional objectives and lesson units). Data for a specific learning problem are usually used to select specific content to be presented to the student and to choose a tutorial strategy for effective presentation of the content in the instructional process. Diagnostic information collected on a curriculum ievel may be used to assign the student to the appropriate level in a hierarchical curriculum structure and to prescribe a general level of instructional treatment (i.e., assignment of learning resources like text reading, specific CBI unit, practice problems). The collected data may also be used to evaluate the courseware itself and to make training management decisions. The authoring system should have a capability for the training developer to specify in the authoring process what kind of data should be collected and how and for what purpose the data should be used and managed.

For instructional diagnosis, courseware evaluation, and training management, a variety of data for the trainee group as well as for individuals should be collected. Important data to be collected include: student performance levels, time on the different levels of curriculum structure, and specific learning patterns including interactions between the student and system. If the courseware is installed and implemented into a computer-managed instructional (CMI) system, most data for management decisions and high level instructional prescription assignments may be collected by the CMI system. That is, if every courseware to be developed using a given authoring system is to be implemented in a CMI system, the authoring system's capability to specify some data collection options in the courseware may not be necessary, depending upon the CMI system's data collection capability.

CRITERIA FOR SELECTING AN AUTHORING SYSTEM

Depending upon the intended use and organizational situation, many different considerations should be given to determine what authoring system, if any, should be used. For example, the variety of courseware to be developed, frequency of use, the developer's experience, etc. should be considered. That is, the criteria for selecting an authoring system should be determined by the given situational needs. However, we think there are several primary factors that should be considered in most situations. They are: (a) functional capability of the authoring system; (b) instructional aids built into the system; (c) flexibility and expandability of the system; (d) ease of learning and using the system; (e) cost/ affordability.

Functional Capability

In the above section, we examined five main stages of courseware development and important functional features that an authoring system should have in order to provide specific aids to the computer-naive author in developing courseware. Thus, the first criterion for selecting an authoring system is its functional features and its capability for using the features. Some systems may not have all the functional features discussed earlier. Also, the system's capabilities for using the features may not be sufficient for given needs. For example, the system may have a graphic editor, but the editor's function may be limited to the development of very simple graphics such as straight lines, boxes and circles. Thus, the instruments for selecting an authoring system should begin with: (a) a complete check-list to note the functional features of the given system; (b) a set of rating scales to evaluate the capabilities for using the features in the given need. A benchmark method could be used to validate the functional features and their capabilities assessed with the checklist and rating scales.

Instructional Design Guidance

In the above section, we stated that an authoring system should provide instructional design guides. Important instructional design guides include the following:

Guides for the Lay-out of Lesson Structure. As an example of instructional design guides, we mentioned icon-based aids in the

above section (e.g., Course of Action, PCD3, Course Builder). A lesson structure, as specified in the design document or as the initial design, is laid out with a simple arrangement of a limited number of icons representing different lesson components. Each lesson component (e.g., presentation displays such as text or graphics, interaction process, etc.) is separately developed but the developed components are automatically combined into a lesson structure as specified in the icon-based lesson lay-out. The lesson structure can be reorganized with a simple rearrangement of the icons using a mouse or other input device. This simple editing procedure allows for the icons to be deleted, moved to a different place in the structure, or added at any point in the lesson structure.

Aids for Selecting Instructional Strategies. For situations in which the training developer needs to select specific instructional strategies, the authoring system needs to provide a set of alternative strategies in a menu form for the developer to choose. If the chosen strategy requires further specification, the system should provide the next level of alternatives in a menu form again. For example, if the author chooses to develop feedback to the student's response, the system may ask what kind of feedback the author wants to provide in a menu (or command) form. Also, the system may ask the developer when the feedback should be presented (immediately or delayed); if the author chooses a delayed presentation, the system may request the author to specify how long it should be delayed. To provide additional helps, the system can have a built-in advisor for the selection of instructional strategies. The advisor may contain explanatory information to help the author select the most appropriate strategy for the given situation. These kinds of interactive aids for choosing appropriate instructional strategies not only save authoring time, but also improve the quality of courseware. Many important instructional strategies that should be embedded in an authoring system are discussed by Merrill (1985, 1987) and Muller, Levy, Nelson and Dean (1988).

Templates for Developing Lesson Components. The authoring system can provide templates for developing specific lesson components (e.g., presentation displays, interaction algorithms, etc.) selected by the author. A template is a pre-formatted screen form which the author fills in to develop a specific lesson component. It may be associated with an icon or other representation (e.g., title) of a lesson component used in the lay-out (or design) of the lesson structure, as described above. The template is usually presented not

only with an appropriate format for developing the selected instructional component, but also verbal guidance, when necessary. For example, if the author selects to develop a multiple-choice type of question, the system asks the author to provide the question statement, correct answer(s), incorrect alternatives, feedback for the correct answer and incorrect answers, etc.; these requests may be made with provision of appropriate templates for the development.

Aids for Selecting Data Collection Options. Specifications of data collection and their use in the instructional process is one of the important instructional design decisions that should be made during development of the program. The authoring system may have the capability to allow the author to specify the data to be automatically collected during implementation of the courseware. The system may have a capability to allow the author to specify even how the data should be analyzed and what kind of reports should be generated. Again, the need for this aid will be dependent on the CMI system in which the CBT program is implemented.

Flexibility and Expandability

In the previous section, we noted that it is not desirable to build all the technical functions required in courseware development and delivery into a single authoring system because of the technical difficulty of building such a system and the the complexity involved in using it. An ideal solution to this problem is to expand the system's functional capability by allowing it flexibility to interface with other software and hardware systems. Important software and hardware systems that can be interfaced with an authoring system are described below.

Interface with Existing Development Software Tools. There are many different kinds of off-the-shelf software tools that can be directly used in the process of courseware development. For example, a wordprocessor, graphic and animation development tool (or packages), spreadsheet program, computer game development tool and others can be used in the development of the courseware. These specialized software tools are usually more powerful than the authoring system for handling the special functions because most authoring systems are designed to do many different functions required for the development of various types of courseware. However, lesson materials developed using the specific software

tools need to be transportable into the courseware structure. The interface capability of an authoring system with the specialized tools makes the transportation possible without requiring a complex programming procedure.

Interface with Programming Languages. The generic, multi-functional structure of an authoring system allows users to develop various types of courseware components (e.g., text, graphics, simple interaction, answer judging, etc.). However, its functional power for developing specific, complex courseware components (e.g., special graphics, unusual interaction routines, etc.) is limited. This limitation can be overcome when the system has an interface capability with a high level programming language such as BASIC, PASCAL, C, etc. That is, some specific lesson components which are difficult or impossible to develop with an authoring system may be developed with a programming language that is more powerful and flexible (although it requires programming skills). In order to transport courseware components developed with the programming language into the courseware structure, however, the authoring system must have an interface capability with such a programming language.

Interface with Built-in Utilities in the Computer. Some utilities built into the software operating system of the computer (e.g., calculator, clock, letter fonts, etc.) can be used in the development and delivery process of the courseware. However, the use of built-in utilities requires the authoring system's capability to interface directly with the operating system.

Interface with Training Management Systems. As the computer is used for instructional management as well as for delivery of courseware, various types of training management systems (including CMI) have been developed. Important functions of a training management system may include: integration of various types of courseware and training programs (including non-CBT programs) into a curriculum structure; development of various types of test items without requiring programming skills; administration of tests and collection of data; generation of high level instructional prescriptions based on the test results (e.g., study assignments for unmastered instructional units or courses); generation of reports on different levels of instructional units (e.g., objective, lesson, course, curriculum, etc.) for trainees, trainers, and administrators; limited statistical analysis of the collected data (e.g., test item

difficulty analysis, discrimination analysis, descriptive statistics of student records, etc.). If a CMI system is available in the given training environment, courseware developed using the authoring system may need to be integrated into the management system. The integration can be made when information collected during the courseware implementation can be transferred to the management system through the interface mechanism. The authoring system used to develop the courseware should have a capability for the author to specify the interface procedure between the courseware and the management system.

Interface with Hardware Peripherals. A variety of hardware peripherals including videodisc, VTR, CD, printer, etc. are used as auxiliary delivery devices in CBT environments. The authoring system should have the capability to allow the author to use peripherals as necessary in the development and delivery of CBT materials. This is particularly important for the integration of video materials into the CBT lesson structure since computer-controlled interactive video has become an essential part of CBT. The video interface capability should allow the author not only to specific the presentation of video materials during the courseware implementation, but also to combine the video materials (including sound) and computer-generated courseware materials into a single presentation display.

Interoperability among Independent Hardware Systems. Due to the rapid development of hardware technology, a specific hardware-dependent authoring system not only limits its use (development and delivery) to the specific type of hardware, but also will be quickly obsolete. Thus, the interoperability among independent hardware systems is a very desirable feature for most software including authoring systems. But, efforts to develop hardware-independent authoring systems have had limited success. For example, some authoring systems require specific hardware for development, but courseware developed using the authoring system can be delivered through other types of hardware (e.g., IMSATT, WICAT's WISE, Authorware's Course of Action). In spite of the high desirability of a completely hardware-independent authoring system, it does not seem to be available in the near future because of the technical difficulty for the development.

On-line Communication and Network Capability. Most computers have an electronic network communication capability.

The communication capability can be used in CBT for interactions between the student and instructor and among students themselves. For example, the student may ask questions of the instructor (or another student) or provide comments on the CBT lessons through an electronic communication channel. The instructor may answer the student's questions and provide new assignments through the communication channel. The communication can be made through an electronic mail facility, which allows sending and receiving messages, or an electronic "talk" facility which allows for direct conversations between the computer users by typing on their screens. In order to use this communication facility during the implementation of CBT courseware, with a minimum interruption of the instruction process, the author should specify the use of the communication facility in the process of developing courseware.

Although some main-frame based systems (e.g., PLATO) have the capability to send and receive electronic messages (including real time computer-based talks) without forcing the user (including student) to close out the program in use (e.g., courseware), most network systems do not have the capability to process the communication facility during the active operation of other software programs. A system's capability to implement the communication facility and courseware simultaneously is useful because it provides an on-line tutorial function (i.e., the student can ask questions and the instructor can provide advice) during the learning process. Furthermore, if the system has a capability to allow for one user to directly monitor another user's screen displays, the instructor not only can observe directly the student's learning process, but also can give demonstrations to the student by allowing him/her to observe his/her own screen displays.

Use of Variety of Input Devices. The keyboard is the primary input device used by the student during instruction. Because of the typing requirement, however, other devices should be provided for the student to use in interacting with the computer. Mouse, touch-screen, and light-pen are representative examples. These devices can be used separately or simultaneously with the keyboard. The best input device should be selected based on the student's ability and interest, the system-student interaction requirements, and availability of the devices. The authoring system should allow the author to select the specific input device for the student to use in a given situation. If a specific input device is selected, it should be specified when and how the device should be used during the

instruction. For example, if a mouse (or touch-screen) is selected, the point and time at which the mouse should be used (or screen areas to be touched) must be specified. Thus, an authoring system should allow the author not only to select specific input devices to be used by students, but also to specify how they should be used. These input devices should also be available for the author to use in developing courseware materials (e.g., presentation displays and interaction algorithms, etc.).

Ease of Learning and Use

The primary purpose of an authoring system is to provide training developers with computer programming aids and instructional design aids so that they can develop CBT materials without programming skills and extensive knowledge/experience in training development. Thus, learning and using the authoring system should be as easy as possible. Because an authoring system itself is a kind of software, however, learning and using the system requires significant effort and time. Therefore, "ease of learning and ease of use" is an important criterion for selecting an authoring system. In assessing the ease of learning and use, the following sub-criteria should be considered:

Flexible Design and Development Process. Usually, courseware development is a kind of mechanical process to code the courseware design specifications into computer programs. Thus, any change in previously-coded portions of the courseware requires not only the revision of the design specifications, but also the reprogramming of the courseware structure and interaction procedures. Since, courseware design/ development is a process which combines instructional design principles and the author's intuitive sense of teaching into a lesson structure (which is runnable on the computer), the need for design revisions during the development process frequently arises. This need for design revisions during development requires a flexible process for these activities rather than a rigid sequential design-and-development process. Some authoring systems allow a flexible design and development process. For example, a courseware lesson can be designed completely before the development; or components of the lessons can be developed separately in any order and then sequenced in a lesson structure; or design and development can be processed simultaneously. In this flexible system, any structural change of a courseware lesson can be made with a simple reorganization of the lesson lay-out

(i.e., a sequential flow of lesson components) which consists of representations (e.g., icons, simple titles) of different lesson components. The lesson structure is automatically changed when the design lay-out is reorganized because every content component in the lesson is attached to its representation (e.g., icons, titles) in the lay-out (see "Course of Action" by Authorware, "Course Builder" by TeleRobotics International, and "PCD3" by Control Data Corporation). This flexible process of design and development not only improves the quality of courseware, but also saves a significant amount of revision time.

Easy Development and Revision of Courseware Components. The ease of learning and using of an authoring system should be evaluated in terms of the procedural difficulty for developing new courseware materials and revising already-developed materials. To increase the facility to learn and use, various user friendly features such as icon-driven menus, pull-down and pop-out menu selection, English-like prompts, multiple-windows, etc. are adopted in many authoring systems; the use of programming language-like commands is minimized.

Provision of Built-in Development Templates. For the development of certain types of courseware materials (or components), the system may provide templates for the author to develop the components in required structure and formats. For example, if the author selects to develop multiple-choice questions from a given menu of test types, the system may provide appropriate templates for the author to develop the instruction for the test, test questions, response alternatives for each question, correct answer(s), feedback for the correct and incorrect answers, number of allowed trials, presentation order of the questions, success criterion, etc. The provision of the templates in a logical order and the automatic integration of the components developed in the given templates not only save the author's development effort and time but also increase the quality of the courseware.

Easy Integration of Developed Components into Courseware (Lesson) Structure. As reviewed in a previous section, a CBT lesson consists of various components (e.g., text, graphics, sound, video, interaction algorithms, etc.). Once the components are developed, they should be integrated into a lesson structure. The integration includes not only the organization of the developed lesson components into a lesson structure, but also the combination of

different display segments for the development of a new display or lesson component (e.g., overlays of text, graphics and sound on video frames). An authoring system should have the capability to allow the author to integrate various display materials and procedural algorithms into a new lesson component or a lesson (curriculum) structure without requiring complex development specifications or programming skills.

In an earlier section, we have recommended special-purpose software tools (e.g., word-processors, graphic editors, sound generators, etc.) for the development of specific courseware materials because they are usually easier and more powerful than authoring systems. To utilize the power of various specific-purpose software tools in the courseware authoring process, however, the authoring system should have a capability to interface with the software tools and to integrate materials developed using them into a new program or a lesson structure.

Adaptation to Author Skill Levels. Usually, there is a trade-off between ease-of-use and the functional power of a software tool because of the technical difficulties for developing these two *requirements into a system. Also, an easy system to use for* inexperienced authors may not be efficient for experienced authors. Thus, many systems are developed with a primary emphasis on one requirement depending upon the specific needs of the intended users (i.e., ease-of-use or functional power). One of the solutions to this trade-off dilemma between ease-of-use and functional capability of the system is to develop different levels of system functions that can be selectively used according to the author's skill and experience. For example, the low level of the system function will be selected by inexperienced users because it would be easy for them to use even though its capability may be limited to the development of relatively simple features (e.g., use of simple templates). However, experienced and skillful authors may wish to use the sophisticated level of the system if it is more efficient and powerful. The low level of the system functions may be implemented with simple menu-driven procedures, while use of simple programming commands or codes may be required for the sophisticated level.

Development Using the Student Mode. One of the tricky problems in authoring courseware with a programming language is that the author can not see the exact features of the courseware

from the implementation angle (i.e., the student mode) until the program is actually executed. Thus, the author needs to repeat a cycle of the programming, execution, and reprogramming (i.e., editing) process until it looks and runs as designed. Obviously, this process is tedious and time-consuming. However, some authoring systems allow the author to develop the courseware in the student mode. Although the interaction procedures between the system and student may still require trial executions to see how they actually work, most display components are developed under direct observation from the student mode (i.e., implementation angle).

HELP Features Built into the System. On-line HELP which the author can obtain during the process of designing and developing courseware is an important feature for determining the degree of ease-of-learning and ease-of-use of the system. There are several types of system-provided HELP. The most common type is an on-line "user manual" which is mostly used as a reference. The second type of HELP is a kind of user tour guide for showing the system functions and procedures for using the system. The tour may allow the user to observe a demonstration program and to practice the observed features directly on the demonstration program. This type of HELP is mainly designed for facilitating the beginning user's learning of the system. The third type of HELP is a built-in tutorial lesson. The lesson may include step-by-step instruction, sample demonstrations, and practice. The fourth type is the presentation of specific information based upon user requests or problems encountered during the authoring process. Because of the purposes and program structures, the first three types of HELPS are usually stored in the system as independent programs. So, the HELP features are not available during the authoring process unless the author gets out of the system and calls in the program. However, the last type of HELP is usually integrated in the authoring system as a part of the system program. Thus, the author may be able to request specific help information during the authoring process without interruption of his/her authoring (development) process. Although all four types of HELP can be developed for the system, the last one is particularly important because of its accessibility during the authoring process. The quality of the HELP, as well as its existence, should be evaluated to determine the degree of ease-of-learning and ease-of-use of the system.

An Adequate User Manual. If the system is user-friendly and has built-in HELP, one might think that a printed user manual might

not be necessary. However, a User Manual booklet is a must for most authoring systems because the structural features and functional procedures of many authoring systems are too complex for most users with limited experience in computer software. Also, as discussed in the section above, most authoring systems do not have built-in-HELP features in the system for the author to refer to during the authoring process. Thus, to become familiar with the system the user needs frequent access to the manual during the learning and authoring process. Clear and concise step-by-step explanations, examples, indexes, etc. are important criteria for judging the quality of the user manual.

Cost: Hardware Requirements, Price, and Contract Terms

Cost, including hardware requirements, the system price, and contract terms, is of course an important factor for the selection of an authoring system. The cost factor should be carefully examined in terms of the affordability, cost-benefits of the system, and the contract restrictions.

Hardware. Although we have discussed the hardware-independence (i.e., interoperability among different hardware systems) of an authoring system as an important feature for evaluating its flexibility, most authoring systems are hardware dependent. Some systems have the capability to develop courseware that can be delivered with more than one type of hardware. For development, however, every system we have examined requires a specific hardware system. Therefore, the necessary hardware for using a given authoring system is one of the main expenses in determining the total cost. In most cases, the available hardware system is usually an a priori condition for the selection of an authoring system. However, the specific configurations required for authoring and delivering courseware with the given hardware system should be thoroughly analyzed. For example, it is necessary to check whether the existing hardware system is already equipped with the required software boards, number and type of disk drives, memory capacity, etc. If they are not available, the cost to equip the configurations should be estimated. If peripheral equipment (e.g., video players, printer, etc.) is needed, the brands and models that can be interfaced with the main hardware system through the authoring system should be examined in terms of the cost, functionality, and quality.

Price. The complete price of the authoring system should be examined in terms of affordability and cost-effectiveness. Sometimes, the system price is different depending upon the components included in the system. For example, some components (e.g., peripheral interface program, instructional management program, advance graphic editor, etc.) may be added to the basic system package with additional cost. The present and future use of the different components, as well as their prices, should be examined.

Contract Terms. Detailed contract terms should be considered when selecting an authoring system. Important contract terms include contract type (ownership or leasing), duration of lease (years versus in perpetuity), site license versus workstation license, distribution restrictions, provisions of upgrades, debugging and maintenance, provisions for technical help to users, etc. Expenses for different contract specifications, if additional costs are involved, should be included in the price of the system. Depending upon the contract specifications, the total expense may be significantly different; nevertheless, the overall costing evaluation should take into account the complete life-cycle costs.

Training Expenses. Sometimes user training is included in the contract terms. If it is not, training expenses should be estimated. The training length, tuition, and the expected skill level to be achieved from the training are main factors for estimating the training expenses. A user friendly system with a good manual and system built-in HELPS may not require formal training. Depending upon the user experience in computer software (including authoring systems) and system sophistication, however, most authoring systems require significant amounts of time for training and/or system familiarization. Employee time to learn the system should be taken into account when the total expense is estimated.

SUMMARY

In this paper, we propose five dimensions of CBT authoring systems as criteria for evaluating an authoring system. The five dimensions are: (a) functional capability for developing courseware materials; (b) instructional design aids; (c) flexibility and expandability; (d) ease of learning and ease of use; (e) cost and

affordability (see Figure 5). For each dimension, a number of specific characteristics are discussed.

However, specific evaluation criteria should be selected not only on the basis of the five dimensions and their characteristics, but also on the basis of other factors unique to the given situation.

First, the five dimensions and their characteristics discussed in this paper are by no means exhaustive. Many other variables unique to the given training development situation should be considered in selecting the evaluation criteria. The important variables may include training and experience of the developers, characteristics of the subject domains, goals and objectives of the CBT, existing hardware and software, expected period and frequency for using the system, etc. Depending on the given situation, some of the characteristics discussed in this paper may not even need to be evaluated.

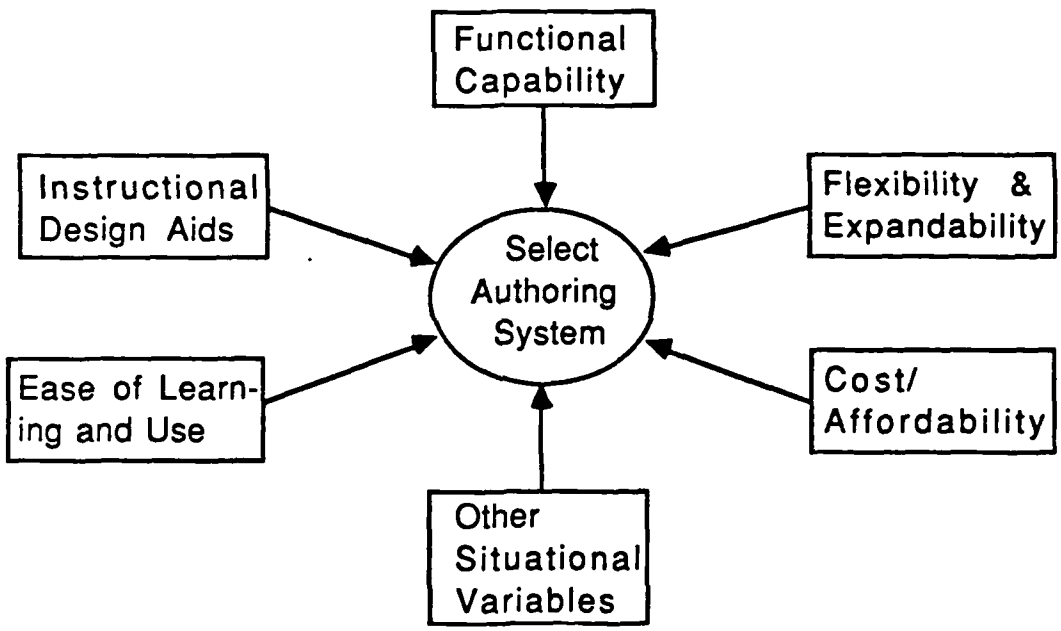
Second, the functional dimensions and their characteristics to be evaluated should be weighted in terms of their degree of importance in the given situation. For example, if every developer in the organization has knowledge and experience in CBT courseware design and development, instructional design aids may not be very important; if every developer has experience in computer programming, the ease-of-learning and -use criterion should not be a major concern.

Third, dimensional characteristics discussed in this paper are not completely independent of each other. For example, the 'flexibility and expandability' of the system are directly related to its 'functional capability'. For example, the system's interface capability with other software tools can enhance its functional capability; instructional design aids have a direct impact on the ease-of-learning and -use. However, we believe that separate and independent examination of the characteristics described in each dimension provides information for analytically assessing the various functions of an authoring system.

Finally, as we discussed in the introductory section, evaluation of an authoring system needs more than a checklist or development of a few simple benchmark lessons for assessing general characteristics of the system. We recommend using both the checklist and benchmark methods. However, selection of evaluation

items to be included in the checklist and of lesson characteristics to be developed in the benchmark evaluation requires a systematic analysis of the functional dimensions and their characteristics. Although situational variables mentioned earlier should be considered in the design of the evaluation plan and criteria, we believe the five dimensions and their characteristics discussed in this paper provide guideline information for preparing the evaluation checklist and identifying the lesson characteristics to be developed for the benchmark evaluation.

Figure 5. Criteria for Selecting Authoring System



REFERENCES

Avner, A., Smith, S., & Tenczar, P. (1984). CBI authoring tools: Effects on productivity and quality. Journal of Computer-Based Instruction, 11, 85-89.

Locatis,C., & Carr, V. (1985). Selecting authoring systems. Journal of Computer-Based Instruction, 12, 28-31.

Merrill, M. D. (1985). Where is the authoring in authoring systems? Journal of Computer-Based Instruction, 12, 90-96.

Merrill, M. D. (1987). Prescriptions for an authoring system. Journal of Computer-Based Instruction, 14, 1-10.

Muller, D. G., Levy, L., Nelson, H., & Dean, D. (1988). The role of authoring systems in the development of interactive videodisc courseware. Journal of Interactive Instruction: Development, 1, 16-24.

Park, O. (1983). Instructional strategies: A hypothetical taxonomy. Control Data Corporation Technical Report TR-1983-3. Minneapolis, MN: Control Data Corporation.

Hillelsohn, M. J. (1984). Benchmarking authoring systems. Journal of Computer-Based Instruction, 11, 95-97.

Appendix A

TYPES AND FORMS OF INSTRUCTIONAL STRATEGIES*

1. Pre-Instructional Strategies

    1) Instructional Objective
    - Terminal Objectives and Enabling Objectives
    - Cognitive Objectives versus Behavioral Objectives
    - Performance Criterion and Condition Specifications

    2) Advance Organizer
    - Expository Organizer versus Comparative Organizer
    - Verbal Organizer versus Pictorial Organizer

    3) Overview
    - Narrative Overview
    - Topic Listing
    - Orienting Questions

    4) Pretest
    - Types of Test (e.g., Objective: true-false, multiple choice, matching, etc. versus Subjective: short answer,essay, etc.)
    - Order of Test Item Presentation (e.g., random, sequence, response-sensitive, etc.)
    - Item Replacement (e.g., with or without replacement of presented items)
    - Timing (e.g., limited versus unlimited)
    - Reference (e.g., criterion-reference versus norm-reference)

2. Knowledge Presentation Strategies

    1) Types of Knowledge Presentation
    - Generality (e.g., definition, rules, principles, etc.)
    - Instance (e.g., examples and nonexamples)
    - Generality Help (e.g., analytical explanation of generality)
    - Instance Help (e.g., analytical explanation of instance)

2) Formats of Knowledge Presentation
- Enactive, concrete physical representation
- Iconic, pictorial/graphic representation
- Symbolic, abstract verbal or notational representation

3) Forms of Knowledge Presentation
- Expository, statement form
- Interrogatory, question form

4) Techniques of Knowledge Presentation
- Mnemonic
- Metaphors and Analogies
- Attribute isolations (e.g., coloring, underlining, etc.)

3. Interaction Strategies

1) Questions
- Level of Questions (e.g., understanding/idea versus factual information)
- Time of Questioning (e.g., before or after instruction)
- Response Mode Required (e.g., selective versus constructive; overt versus covert)

2) Hints and Prompts
- Formal, Thematic, Algorithmic, etc.

3) Feedback
- Amount of Information (e.g., knowledge of results versus explanatory feedback)
- Time of Feedback (e.g., immediate versus delayed feedback)
- Type of Feedback (e.g., cognitive/informative feedback versus psychological reinforcing)

4. Post-Instructional Strategies

1) Summary
- Narrative Review
- Topic-Listing
- Review Questions

2) Post Organizer
- Conceptual mapping
- Synthesizing

3) Post Test
- Types of Test (e.g., Objective: true-false, multiple choice, matching, etc. versus Subjective: short answer, essay, etc.)
- Order of Test Item Presentation (e.g., random, sequential, response-sensitive, etc.)
- Item Replacement (e.g., with or without replacement of presented items)
- Timing (e.g., limited versus unlimited)
- Reference (e.g., criterion-reference versus norm-reference)

5. Instructional Control Strategies

1) Sequence
- Linear
- Branching
- Response-sensitive
- Response-sensitive plus aptitude-matched

2) Control Options
- Program control
- learner control
- learner control with advice
- condition-dependent mixed control

* Types and forms of instructional strategies listed in this table are not complete. The classifications are arbitrarily made by the authors.