K/DSRD--444

DE91 006831

NALDA CAI: Recommendations for an Advanced Instructional Model

B. H. Handler
P. A. France**
S. C. Frey**
N. F. Gaubas**
W. H. Hunnum§
K. J. Hyland**
A. M. Lindsey**
D. O. Manley**
D. L. Smith¶

**Oak Ridge Associated Universities (ORAU)
§University of North Carolina at Chapel Hill
¶Memphis State University

July 1990

Research Sponsored by the
Naval Aviation Maintenance Office
under Interagency Agreement No. 1363-1363-A1
with the U.S. Department of Energy

Prepared by
Data Systems Research and Development
Applied Technology Organization

Located at the
K-25 Site
Oak Ridge, Tennessee 37831-7171
Operated by
Martin Marietta Energy Systems, Inc.
for the
U.S. Department of Energy
under Contract No. DE-AC05-84OR21400

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

The authors would like to acknowledge the valuable assistance provided by many individuals throughout the Naval Aviation Logistics Data Analysis (NALDA) computer aided instruction (CAI) project. Special mention is given to Pat Halsey, Al Huntley, and Cathy Snyder who gave their professional time to read and evaluate the manuscript. Alice Twitty and Lynn Duncan contributed many useful editorial suggestions. Velma Reese, Reba Skeens, and Sharon Parsons, of Data Systems Engineering Organization (DSEO), and Barbara Young and Doris Griffin, of the Training and Management Systems Division (TMSD) of Oak Ridge Associated Universities (ORAU), provided support services beyond the call of duty. Paul Tarrant and Heidi Spurling answered an infinite number of editorial and word processing questions. To each of the above, we give a special expression of our gratitude.

LIST OF ACRONYMS

3M	Maintenance and Materials Management System
AI	Artificial Intelligence
AV-3M	Aviation Maintenance and Materials Management System
CAI	Computer Aided Instruction
DSEO	Data Systems Engineering Organization
DSRD	Data Systems Research and Development
EPS	Electronic Publishing System (by Intellisance)
ET	Embedded Training
HT/HM	Hypertext/Hypermedia
ICAI	Intelligent Computer Aided Instruction
ISD	Instructional Systems Development
ITS	Intelligent Tutoring System
NALDA	Naval Aviation Logistics Data Analysis
NAMO	Naval Aviation Maintenance Office
PC	Personal Computer
XPL	Expert Programming Language

EXECUTIVE SUMMARY

The training of personnel to effectively utilize the Naval Aviation Logistics Data Analysis (NALDA) system presents several challenges because users are geographically separated, represent various disciplines such as engineering, safety, supply and maintenance, and have diverse knowledge and experience levels. NALDA personnel can benefit from training which incorporates instructional design techniques based on advanced computer aided instruction (CAI) tools and technologies, such as hypertext/hypermedia, artificial intelligence, and embedded training.

A review of the literature on CAI indicates a consensus among researchers that CAI is an effective instructional medium when properly designed to maximize the use of computer technology in the instructional process. Researchers point out the need for increased focus on such instructional design concerns as high quality interaction and feedback during the development stage as a way to improve the instructional effectiveness of CAI courseware. Current CAI development trends include the use of hypertext/hypermedia tools, artificial intelligence techniques, embedded training concepts, and cognitively-based instructional design principles, which can be effectively incorporated into CAI designs to enhance instructional effectiveness.

Hypertext/hypermedia is the latest generation of software for database structuring. In an instructional system that has been designed using principles of learning based in the cognitive sciences, hypertext/hypermedia is an effective method for capturing and transferring information to the learner. Hypertext facilitates learning by allowing access to information following a path of the user's own choosing. NALDA personnel need two types of information: procedural (skills-oriented) and declarative (knowledge-oriented). Declarative information supports procedural information. Hypertext is the ideal method for teaching declarative information because it allows NALDA personnel to learn how to use NALDA databases while exploring relationships among pieces of information. NALDA personnel must be able to access several different databases, extract information and utilize it according to their need.

Artificial intelligence, as exemplified by intelligent tutoring systems (ITS), will have a far-reaching impact on the instructional design of future training programs. An ITS uses artificial intelligence techniques to facilitate learning. Ideally, individualized learning is provided through intelligent computer aided instruction (ICAI) that uses the same instructional techniques as a competent human instructor. Currently, costs for ITS development are too prohibitive for mainstream military use. However, as technology is refined and standards for development and evaluation are established, artificial intelligence components in the design and development of instructional systems will increase.

Embedded training is instruction that is delivered through the workstation as work is being done. The user can move easily (and transparently) between actual job performance and instruction for task completion. Embedded training increases worker productivity while decreasing training costs. Most of the embedded training systems to date are unique in their developmental approach so that there is little guidance for developers of new embedded training systems. As authoring tools are developed and further research in artificial intelligence is completed, embedded training should become widespread. The Department of Defense has several embedded training systems focusing primarily on skill sustainment.

During the past twenty years, the basic Instructional Systems Development (ISD) model has been the tool of choice for design and development of instruction in military, business, and industrial settings. Today, new advances in computer hardware and software, newly emerging training technologies, and the latest research on how learning occurs require ISD-oriented instructional designers to assess the traditional ISD approach. The use of hypertext/hypermedia, artificial intelligence, and/or embedded training may require the development of enhanced ISD models that incorporate cognitive learning theories and are responsive to newly emerging training technologies for developing the training of the future.

The advanced CAI instruction model is an enhancement to the NALDA CAI prototype developed by Martin Marietta Energy Systems, Inc., Data Systems Engineering Organization (DSEO). The advanced instruction model incorporates features of new technologies, including answer analysis routines and limited adaptive control, to distinguish it from more traditional CAI. The self-modifying features of some artificial intelligence applications are still too experimental for use in the proposed CAI model. The enhanced CAI model uses hypertext for conveying declarative knowledge in support of job performance and simulations with embedded help to teach procedural knowledge.

This document is meant to serve as an introduction to advanced training technologies for Naval Aviation Maintenance Office (NAMO) personnel who have responsibilities for training and as a source of information for any individuals interested in areas related to CAI.

ABSTRACT

Data Systems Engineering Organization (DSEO) personnel developed a prototype computer aided instruction CAI system for the Naval Aviation Logistics Data Analysis (NALDA) system. The objective of this project was to provide a CAI prototype that could be used as an enhancement to existing NALDA training.

The CAI prototype project was performed in phases. The task undertaken in Phase I was to analyze the problem and the alternative solutions and to develop a set of recommendations on how best to proceed. The findings from Phase I are documented in *Recommended CAI Approach for the NALDA System* (Duncan et al., 1987). In Phase II, a structured design and specifications were developed, and a prototype CAI system was created. A report, *NALDA CAI Prototype: Phase II Final Report* (Handler et al., 1989), was written to record the findings and results of Phase II.

NALDA CAI: Recommendations for an Advanced Instructional Model, is comprised of related papers encompassing research on computer aided instruction CAI, newly developing training technologies, instructional systems development, and an Advanced Instructional Model. These topics were selected because of their relevancy to the CAI needs of NALDA. The papers provide general background information on various aspects of CAI and give a broad overview of new technologies and their impact on the future design and development of training programs.

Section 1

REVIEW OF THE RESEARCH IN COMPUTER AIDED INSTRUCTION

TABLE OF CONTENTS

LIST OF FIGURES

REVIEW OF THE RESEARCH IN COMPUTER AIDED INSTRUCTION

1. BACKGROUND

Computers are widely used and accepted today as a means for providing instruction in a variety of situations. However, only thirty years ago a researcher at IBM wondered if digital computers could be used to deliver training, and proceeded to develop the first instructional program for computers. The concept of computer aided instruction (CAI) has been repeatedly demonstrated as being feasible. Perhaps success has brought with it an unrealistic faith in CAI as the ultimate training method. The appropriate use of CAI lies somewhere between total rejection of CAI as dehumanizing the training process and total reliance on CAI for all training needs. CAI is not a static entity for all applications; there is more than one approach to CAI. Some approaches to developing and delivering CAI have proven to be more effective than others.

1.1 PURPOSE

The purpose of this report is to examine the research on computer aided instruction to ascertain what makes effective CAI. Emerging trends cited in the literature suggest future directions in and refinement of the concept of CAI. Finally, considerations for the design of an advanced CAI model are suggested.

1.2 DEFINITIONS

CAI will be used throughout to describe the use of a computer to present instruction directly to a learner. The learner sits at a computer display, interacting with a computer program designed to teach. Some other terms used to describe the same process are computer based training (CBT), computer assisted instruction (CAI), computer aided learning (CAL). The commonality among these terms is the use of a computer to present instruction. Therefore, the terms can be viewed as interchangeable.

The distinction between CAI, or equivalent terms, and computer managed instruction (CMI) should be noted. In computer managed instruction, the learner may receive instructional content directly from a computer or learning may occur in a more traditional fashion such as by reading a book, listening to a lecture, or viewing a film or video. The computer is then used to track a learner's performance, test the learner's knowledge, suggest remedial or enrichment activities, report on the progress of a class, or perform other activities related to the management of instruction. CMI usually does not "teach" and is used for management purposes only.

2. COMPUTER AIDED INSTRUCTION STUDIES

There are literally hundreds of studies investigating CAI. All of the studies fall into one of three categories, depending on the research strategy employed, as illustrated in Fig. 1.1.

RESEARCH STRATEGIES

1. Proof of Concept Studies
2. Comparative Media Studies - Summary Techniques
   A. Narrative Review
   B. Box Score
   C. Meta-Analysis
3. Optimization Studies

Fig. 1.1. CAI research strategies.

2.1 PROOF OF CONCEPT STUDIES

Earliest CAI studies can be termed "proof of concept" studies, because they sought to demonstrate the effectiveness of instructional use of computers. Proof of concept studies were important during the early years of CAI as no historical data existed. The studies addressed such questions as: Can instructional materials be delivered by computers? Will users accept this method of instruction? Will they learn from the programs? What subjects can computers teach? What kinds of users profit from this form of instruction? The typical approach was to test the instructional program. For example, studies were conducted to see whether factoring mathematical equations could be taught electronically or if CAI could be used to teach rules of grammar. The researcher would develop the computer programs, test them by having users complete the lessons and evaluate the results. Depending on the results, the researcher would report that CAI was or was not effective in that particular circumstance.

In general, proof of concept studies demonstrated that computers could be used successfully for providing instruction. Users reported that they learned the CAI course content, and test scores corroborated the users' impressions (Roblyer, Castine, and King, 1988). Proof of concept studies were repeated many times with different content, types of users, and approaches to CAI (Kulik, Bangert, and Williams, 1983; Kulik, Kulik, and Cohen, 1980). The results indicate that CAI works. Users reported that CAI was acceptable to them; indeed, the learners often reported a preference for CAI over other forms of instruction. Drawbacks seemed to be in two areas: development time and costs (Hannum, 1986). Developing CAI takes many hours; 120-150 hours of development time often generates only one hour of CBT. Costs for delivering computerized training were also troublesome. In the days before the advent of microcomputers, CAI was delivered via mainframe computers. The actual costs of delivering CAI varied, depending on costs for computer time and communications costs for telephone lines which linked the terminals to the mainframe. The actual costs of delivering CAI can exceed the cost of traditional training by tenfold (Hart, 1987). Contrary to the proclamations of some early proponents, CAI was a very expensive means of providing instruction. Undoubtedly the cost served to limit the early acceptance and expansion of CAI. It was hard to fault CAI in terms of educational value but easy to criticize it on economic grounds. The invention of the microcomputer in the mid-1970s radically altered the economics of delivering CAI. No longer was it necessary to have an expensive mainframe computer to deliver CAI; a much lower cost microcomputer could handle the task without communications costs. The costs of CAI were altered forever, making CAI much more affordable and accelerating the CAI revolution that had been much discussed but little realized prior to the use of microcomputers.

As computer hardware changed, the question became, "Can microcomputers be used to deliver CAI?" The answer, beginning with the Apple II was "Yes." Even small, relatively inexpensive microcomputers were capable of delivering CAI. There were limitations, such as disk storage, size of memory, and computational speed, but microcomputers could deliver CAI. Input and output devices were subjects of proof of concept studies. CAI was shown to be effective with a variety of input devices, including keyboards, touch screens, and mice (Hannum and Gleason, 1986; Sawyer, 1985). Output devices examined included teletypes, low-resolution displays, higher-resolution displays, televisions, and non-computer-generated displays such as slide projectors and video disks operating under computer control.

As a result of numerous proof of concept studies, researchers concluded that computers make excellent instructional delivery devices (Jonassen, 1988). It became necessary to conduct additional proof of concept studies as a result of developments in computing equipment and learning theory. Once CAI was demonstrated to be feasible by proof of concept studies, the research strategy shifted. It was no longer necessary to demonstrate that CAI was viable. The question became, "How does CAI compare with other forms of instruction?" The answer to that question precipitated a change in CAI research strategy.

2.2 COMPARATIVE MEDIA

A second research strategy found in reviewing the CAI literature is the comparative media study in which CAI is compared to other instructional approaches. The typical study compared the performance of one group of users who completed a CAI lesson to another similar group who completed an identical or similar lesson using traditional instruction (i.e., a teacher and textbook). User performance would be compared along several dimensions: scores on tests of lesson content, attitudes toward both the way the lesson was taught and the content of the lesson, and time required to complete the lesson (Roblyer, Castine, and King, 1988). CAI was compared with programmed instruction and with instructional television, although traditional instruction was the favored benchmark for assessing the comparative effects of CAI (Kulik, Kulik, and Cohen, 1980; Kulik, Bangert, and Williams, 1983).

The reasoning behind the comparative media studies seems sound. CAI has been demonstrated to be a viable means for providing instruction. The next step is to compare the effectiveness of CAI with other forms of instruction. The question shifts from "Does CAI work?" to "Does CAI work better than other forms of instruction?" This shift in question necessitated a shift in research strategy, and comparative media studies were born.

In terms of volume alone, comparative media studies constitute the bulk of CAI research to date (Roblyer, Castine, and King, 1988). Hundreds of comparisons of CAI to other instructional methods have been reported, making a detailed reporting of each individual study too extensive to be practical for this report. Fortunately, the findings from individualized studies have been summarized.

2.2.1 Narrative Review

One summary technique is a *narrative* in which the reviewer reports research findings in a narrative fashion, using prose to express the research results. Commonly, the reviewer consolidates trends in the literature and makes several conclusions based on these trends. The reviewer uses several individual studies to illustrate important points or conclusions.

Narrative reviews of comparative CAI media studies usually examined several studies that compared the performance of a group of CAI users with a group of users learning under a different instructional method. Reviewers typically reported that the CAI group did as well as, if not better than, the other group on tests of subject matter mastery. In an early narrative review, Jamison, Suppes, and Wells (1974) included CAI research along with research on other instructional media. Based on a small number of CAI studies, the authors concluded that computerized training is an effective supplement to regular classroom instruction and, under some circumstances, can be an effective replacement for regular classroom instruction in secondary schools and colleges. The study decried the lack of empirical data and controlled conditions and referred to many of the studies as "journalistic accounts" of research.

In another review, Thomas (1979) examined the use of CAI in secondary schools and concluded that CAI was effective in promoting achievement gains equal to or greater than gains made through traditional instruction. Thomas (1979) and others found that using CAI resulted in time savings for the users.

Reviewers reported additional information from the studies, such as users' comments. Reviewers concluded that CAI was a more acceptable means of instruction than the means with which it was compared and supported their conclusion with users' comments such as: "I really liked the CAI lesson;" "I like learning this way better;" and "I wish all my classes used CAI."

While the narrative method provides a way to see trends in a series of research studies, it appears to have limitations. The narrative method lacks a way to quantify findings and depends on the reviewer to extract trends from the research. Reviewers with a particular bias, either for or against CAI, might see more support for their position in the studies than is actually there. Because of these limitations, some reviewers depend on other techniques that rely on quantification of research results.

2.2.2 Box Score

A second summary technique is the *box score* method. Using this technique, reviewers recorded each study as a "+" if the CAI group outperformed the comparison group, an "=" if there were no differences between the two groups and a "—" if the CAI group did not perform as well as the similar group. By tallying the results, the reviewer can compute a box score indicating the comparative effectiveness of CAI. The box score method is similar to reporting the win - lose record of an athletic team in which each game played is recorded as a victory, tie, or loss. For example, the overall record can be reported as eight wins, one tie, and two losses. The summary reports the number of studies finding CAI better than, equal to, or inferior to the comparison group. In this manner, quantitative data about the findings are reported. The difference between the narrative method of review and the box study method is quantitative recording.

Edwards, et al., (1975) used a box score method to review 30 CAI research studies. The reviews were organized by educational level, subject area, and whether the CAI was supplemental or replacement instruction. The findings echoed other reviews (Thomas, 1979; Hickey, 1968) indicating that CAI is as effective in promoting achievement as other instructional methods, including programmed instruction and tutoring. Additionally, Edwards, et al., reported savings in instructional time when using CAI. A reduction in learning time with roughly equal achievement is also reported by Orlansky (1983) in a review of CAI used in the military. Orlansky and other researchers reported a median reduction in users' time of 30 percent. The approximately one-third time reduction seems to be the best established "fact" from research on CAI.

While the box score method includes a quantitative approach to summarizing research findings, the conclusions are not noticeably different from the narrative review summaries: CAI results in achievement which is as good as or slightly better than other forms of instruction and requires less learning time. However, box score reviews do not provide a measure of the differences among instructional methods. Knowing that one method was better than another provides partial information; knowing *how much* better supplies important additional information. Knowing the magnitude of any differences between CAI and other forms of instruction has important implications not only for research but for practice. If the differences in achievement between CAI and programmed instruction significantly favor CAI, one would be more willing to make an investment in CAI. If the differences are in favor of CAI, but insignificant, it might not be feasible to scrap an existing programmed instruction course for a course using CAI. In summary, the limitation of the box score method is that it does not measure or quantify differences between alternate methods.

2.2.3 Meta-Analysis

The third technique for summarizing research is *meta-analysis*, pioneered by Glass (1976, 1977). In a meta-analysis, the reviewer estimates the size of the difference between treatment groups. This estimate, called effect size (ES), is computed by subtracting the mean score of the comparison (C), or non-treatment group, from the mean score of the treatment group (X) and then dividing the difference by the pooled standard deviation (SD_p) of the groups. Effect size is expressed in standard deviation units. The general formula is:

$$ES = \frac{\bar{X}-C}{SD_p}$$

In meta-analysis the reviewer locates studies that meet criteria for acceptable research methodology, computes the effect size for each study, and then averages the effect size across studies to estimate the difference between treatments. Cohen (1977) gives the following parameters for interpreting effect sizes:

$$ES < 0.2 = \text{small effect}$$
$$0.5 \leq ES \leq 0.6 = \text{medium effect}$$
$$ES > 0.8 = \text{large effect}$$

Several meta-analysis studies report an effect size of 0.42 with adults, 0.25 at the college level (Kulik, Kulik, and Cohen, 1980); 0.32 and 0.40 at the secondary level (Bangert-Drowns, Kulik, and Kulik, 1985; Kulik, Bangert, and Williams, 1983), and 0.47 at the elementary level (Kulik, Kulik, and Bangert-Drowns, 1985). According to Cohen, these reflect a low to medium effect from using CAI. Meta-analysis studies support the contention that CAI is effective as an instructional method. Furthermore, these studies categorize the magnitude of the effectiveness of CAI; this was not possible in reviews using either the narrative or box score methods.

Meta-analysis methodology is not without critics. Slavin (1986; 1987) has argued that people using meta-analysis tend to lump studies that are fundamentally different into the same category. Individual studies use different CAI treatments and may measure users' outcomes differently. Slavin questions the value of combining studies to estimate the effects due to CAI or any other instructional methods for which meta-analysis is used. The results from such analyses are not credible according to Slavin. Others see more value in the careful analysis of a single well-done study than in averaging many studies, some of which may be less well done. Supporters of meta-analysis recommend using screening criteria to discard "weak" studies (Glass, 1976, 1977).

Clark (1983; 1985) criticizes media comparisons and the meta-analysis of CAI, suggesting that it is not CAI as a delivery device that accounts for improved instruction but rather the instructional methods built into the software. The meta-analyses incorrectly attribute the positive effects found in these studies to the computer as a delivery device rather than the instructional methods used. For example, suppose learning improves when the user practices answering questions about content. Because CAI provides the user with many opportunities to practice answering questions, the method should be effective. However, the cause of this effectiveness is not CAI itself but rather the effect of the practice built into the instructional software. Any instructional delivery system that provides practice should be equally effective. By ignoring the instructional methods, media comparisons inappropriately attribute learning gains to the instructional medium. Others, such as Hagler and Knowlton (1987) and Avner, Moore, and Smith (1980), point out the limitations of media comparisons. Hagler and Knowlton express doubt that instructional media such as CAI represent meaningful experimental variables apart from instructional content and instructional strategy. Hagler and Knowlton urge researchers to move away from comparative media studies because of flaws in the internal validity of such studies and encourage researchers to turn away from examining the media and to examine the users and the instructional methods used.

Despite the differences among comparative studies that use narrative, box score, and meta-analyses as summary methods, the conclusions are remarkably similar. In an elaborate review of the impact of CAI based on comparative media studies, Roblyer, Castine, and King (1988) report several consistent findings.

Computer Aided Instruction:

- offers some benefits over other instructional methods,
- requires less instructional time,
- results in improved attitudes toward computers,

- results in improved attitudes toward the subject matter,
- is successful in a variety of content areas,
- in the form of drill and practice may be better at lower levels,
- in the form of tutorials may be better at higher levels, and
- has a greater effect for lower ability learners.

Roblyer, Castine, and King's (1988) findings represent a summary across many studies and are conservative claims for the effectiveness of CAI supported by empirical data.

2.3 OPTIMIZATION

The third and final strategy for CAI research is conducting optimization studies to identify instructional methods or approaches within CAI that can improve the effectiveness of lessons. Because results from a number of studies (Roblyer, Castine, King, 1988; Chambers and Sprecher, 1980; Hofstetter, 1985) indicated that CAI is an effective instructional medium, the optimization study approach does not seek to compare it with other instructional media. Instead, the research question becomes "How can we improve upon the use of CAI?"

A typical optimization study would use two CAI groups. One group would receive some control treatment, such as a simple "yes" or "no" as feedback. The second group would receive an experimental treatment, such as a more elaborate graphic presentation of feedback. The results from post-tests of the two groups would be compared to determine the effect of the experimental treatment. The most basic type of optimization study is a two-group study of a single variable. Many optimization studies have more elaborate designs and test several variables. The intent remains the same: to determine how to best use CAI.

A summary of the optimization research is difficult because so many variables have been used. It is a more complex matter than compiling all the research in which a CAI group was compared with another group as in the comparative media studies. A useable summary of the optimization research would have to consist of many separate summaries, one for each variable. For example, summaries of all the studies that used graphic feedback, users control of examples, and animated text for explanations would be necessary. There are some summaries of research on particular variables, but there are no comprehensive summaries of the optimization studies. This report partially addresses the need to pull together the optimization studies to guide the continued development of CAI.

In summary, the optimization studies are crucial if CAI is to be used to its full potential as an instructional approach (Hagler and Knowlton, 1987). The proof-of-concept studies demonstrate that CAI is viable (Hickey, 1968). The media comparison studies document some of the advantages of CAI when contrasted with other instructional media (Roblyer, Castine, and King, 1988). However, neither the proof of concept studies nor the media comparison studies indicate how to use CAI more effectively; that task falls to the optimization studies. Fortunately, there is a rich history of optimization studies dating back to the late 1960s and maturing in the decade of the 1980s.

The advent of the microcomputer caused an increased use of CAI. Similarly, CAI research was no longer isolated in a few universities and the military services. Rather, the means for conducting research on CAI were widely available, and the number of research studies grew. Currently, a large body of research exists that can guide in the development and refinement of CAI. Gains in the effectiveness of CAI will result more from the application of research findings than from improvements in computer hardware. Instructional software can be improved. The CAI models, including drill and practice, tutorials, and simulations, can be improved as a result of research, particularly optimization studies. Simply put, the technology of CAI has changed as a result of research. CAI for the 1990s should build on this research to optimize instructional power.

3. CAI MODEL EVOLUTION

CAI operationalizes some models of the learning process (Reigeluth, 1983). The initial basis for CAI is found in the behavioral psychology model of operant conditioning. Early CAI programs consist of simple models of learning that emphasize active practice, repetition, frequent feedback, and linear sequences from a simple to a more complex nature. There are two basic models: tutorial, and drill and practice. These models differ as the tutorial CAI provides initial information before practice, whereas drill and practice CAI consists solely of practice. Simulation, a third form of CAI in which the software reproduced some physical phenomenon, procedure, situation, or process, developed later (Alessi and Trollip, 1985). Tutorial, drill and practice, and simulation remain a common classification system for CAI (Lillie, Hannum, and Stuck, 1989).

3.1 DRILL AND PRACTICE

In drill and practice CAI, the users are presented with a question to be answered. The student supplies an answer, which is compared with the correct answer, and feedback informing the user that the answer is right or wrong is supplied (Alessi and Trollip, 1985; Lillie, Hannum, and Stuck, 1989). Simple drill and practice CAI automates flash cards. While some critics berate the drill and practice CAI, evidence indicates that it promotes learning (Merrill and Salisbury, 1984). Fuller (1986) suggests going beyond drill and practice strategies; however, drill and practice remains the most common form of CAI used in schools today (Salisbury, 1988).

More sophisticated drill and practice strategies have been suggested by several researchers (Merrill and Salisbury, 1984; Salisbury, 1988; and Salisbury, Richards, and Klein, 1985). Specifically, Merrill and Salisbury (1984) suggest that drill and practice CAI could be improved through the use of five techniques:

1. Have users work on a subset of all items in the drill during each practice session to minimize interference with short-term memory.

2. Replace practice items users master with "fresh" items not yet mastered. This would maintain an appropriate level of difficulty.

3. Present practice items (from pool) in random order to eliminate sequential learning effect.

4. Review items from prior drills to maintain a high-level of retention.

5. Use diagnostic pretests to match practice items to the skill level of the users.

The use of drill and practice CAI is supported by research indicating one of the reasons users fail to acquire higher order cognitive skills is that they lack the ability to automatically perform the underlying basic skills (Lesgold and Resnick, 1982; Resnick and Ford, 1981). Evidence indicates that automaticity can be a by-product of drill and practice CAI (Salisbury, Richards, and Klein, 1985). These authors identify several CAI design strategies for drill and practice which are consistent with theoretical approaches to learning.

Different practice techniques are suggested by Salisbury, Richards, and Klein (1985) for various types of instructional outcomes and for distinct stages in the learning cycle. Figure 1.2, taken from Salisbury, Richards, and Klein (1985), summarizes their recommendations. Note the detailed guidance given to designers of CAI regarding practice. Differentiating the type of practice represents an improvement over the simple drill and practice strategy.

Some of the recommendations of Merrill and Salisbury (1984) are echoed by Salisbury, Richards, and Klein (1985). Recommendations, such as working on a subset of items, adapting the difficulty level of the practice to the individual user, and spacing reviews of practice items over time, appear in both studies.

Salisbury (1988) describes six sophisticated drill and practice strategies for CAI that go beyond the simple automated flash card concept: one pool drill, two pool drill, three pool drill, three pool drill with increasing ratio, progressive stage drill, and variable item interval drill.

In the one pool drill structure, the user works all items in the drill. The items may be ordered sequentially or randomly. In the two pool drill structure, items answered correctly are removed and placed in a second pool for later review. When the items in the first pool are completed, the user moves to the second pool for review. In the three pool drill structure, there are three pools of items: a pool of items the user is currently working on (usually 5 - 7 items), a pool of items that has been mastered (the review pool), and a pool of items that has not been presented (the replacement pool). In the three pool drill with increasing ratio structure, there are the same three item pools, but new items and review items are introduced into the drill with an increasing ratio of review to new items as the user progresses through the lesson. There are several different ways to determine when to introduce a review item or a new item. Each requires sophisticated instructional software going far beyond simple drills. In the progressive stage drill structure, each item may be presented in various drill modes. An item may be presented in a pretest mode, a rehearsal mode with an accompanying cue or hint, a drill mode without cues or hints, or one of three review modes: daily, weekly, or monthly. In the variable item interval drill structure, each item answered incorrectly will be presented again in a varied position, not in a fixed sequence.

Salisbury (1988) reports on research supporting sophisticated drill and practice strategies, which suggest a more appropriate use of computers in the instructional process than the earlier, simple drill and practice programs. The effectiveness of practice on learning has been established by research and can be explained from different theoretical viewpoints. Any designs of effective drill and practice models should incorporate recent research findings that support productive practice through computer use.

Types of Learning	Initial Practice	Intermediate Practice	Final Practice
Learning Facts	•Tell students the facts they are to learn. •Provide cues to highlight the facts when a number of facts are to be learned at one time (e.g., a table of basic multiplication facts highlighting pairs of numbers and their products).	•Have students apply the facts as well as state them (e.g., using basic multiplication facts). •Fade cues used in initial practice. •Increase amount of practices on facts.	•Have students apply all facts when possible. No longer require students to state the facts. •Eliminate all cues. •Have students practice the total criterion behavior (e.g., all multiplication facts are practiced in a random order).
Defining Concepts	•Define concept and ask students to state it. •Use cues to differentiate between instances and non-instances. •Have students classify instances. •Use concrete objects in early practice when possible.	•Continue to give instances and non-instances but make more difficult to differentiate. •Have students generate some instances. •Continue to ask students to state definition. •Gradually make instances more abstract.	•Require students to state definition and classify or generate instances without cues. •Make practice abstract, verbal, or symbolic.
Giving Explanations	•Provide students with the explanation and require them to practice stating it. •Provide specialized cues (like flowcharts and demonstrations) to help students relate multiple concepts. •Give examples and nonexamples of the explanations.	•Continue giving examples and nonexamples but gradually make them more difficult. •Require students to produce the explanations in their own words in addition to giving the model explanation. •Gradually fade cues.	•Have students state the explanation in their own terms without the use of cues. •Require students to use the concept being explained instead of just giving the explanation (e.g., using the principle of reinforcement, not just stating it).

Fig. 1.2. Practice treatments for five types of learning. Source: Salisbury, Richards, and Klein, 1985.

Types of Learning	Initial Practice	Intermediate Practice	Final Practice
Following Procedural Rules	•Provide students with steps for following the procedure or give a demonstration of the procedure. •Require students to practice steps in the procedure. •Break down long or difficult procedures into smaller units and practice the individual units (e.g., learning to drive may start with learning rules of the road before getting behind the wheel).	•Continue practice of individual steps in the procedure. •Add additional steps to long or difficult procedural chains.	•Require students to practice the procedure in full with no breaks in the chain.
Solving Problems	•Give students the rules needed to solve the problem. •Require students to distinguish correct answers from incorrect answers. •Break up difficult or long problems into smaller units and practice individual units (e.g., solving long division problems is practiced step by step).	•Give students a problem with a wrong answer and ask them to correct it. (Practice moves from the recognition phase to the editing phase.) •Require students to generate their own rules for solving a problem.	•Require students to solve the problem with no help. (Practice moves into the production phase.) •Ask students to generate their own solutions when possible (i.e., long division has a fixed procedure, and unique solutions may not be found). Other problem solving can be unique.

Fig. 1.2. (continued)

3.2 TUTORIAL

Tutorial CAI differs from drill and practice CAI in that tutorial CAI presents initial instruction before providing drill and practice. Drill and practice CAI does not include any specific instruction, introduction, or explanation of the content; it just includes drills or practice. Thus drill and practice CAI is appropriate for use after some initial instruction has been completed. Drill and practice CAI serves to reinforce the initial instruction as well as to make the responding more automatic for the users.

The most common tutorial model presents information to the users in the form of a statement of facts, a description of concepts, or a statement of rules, and follows this material with practice in recalling the information (Alessi and Trollip, 1985; Lillie, Hannum, and Stuck, 1989). These tutorial models have their theoretical basis in behavioral psychology. In fact, many early tutorials took the form of programmed instruction. While tutorials that rely totally on programmed instruction principles have become less widespread, the influence of programmed instruction remains. Programmed instruction workbooks are the metaphor for tutorial CAI.

In recent years, the use of tutorial CAI has been questioned. Derogatory terms such as "page turner" and "electronic workbook" have been applied to much tutorial software and the tutorials are not viewed as utilizing the potential of computers to provide tutorial instruction (Leiblum, 1982). The tutorial CAI has sustained limited success.

Several variations on the basic tutorial model have been suggested. In the RULEG model, the CAI would first present a rule statement, possibly followed by an explanation of the rule. For example, a RULEG model could present a rule such as "an adjective must agree with the noun in number and gender." An example would then be presented showing how this rule is applied. Finally, the users would be asked to complete an example that requires use of the rule. At this point the tutorial looks like drill and practice CAI.

Less crafted tutorials consist of a few screens of rambling prose followed by drills. Research evidence demonstrates that tutorials even using a restricted model, are effective in promoting learning (Robyler and King, 1983; Samson, et al., 1985). In these studies, tutorials are found to be superior to drill and practice. Exceptions to these findings are found in some studies comparing tutorials with drill and practice at the elementary level (Robyler, Castine, and King, 1988). Perhaps, as these authors noted, the effectiveness of different types of CAI varies with the level of the user. Nevertheless, there is research to support the contention that the "basic" tutorial model is effective.

Recent advances in cognitive psychology have fostered the development of tutorials that vary considerably from the behavioral-based models. Gagne, Wager and Rojas (1981) describe a tutorial model built around Gagne's information processing model of instruction. In this model, the computer utilizes nine instructional events that Gagne (1985) identified as leading to successful learning:

1. Gain the user's attention.

2. Inform the user of the objective.

3. Stimulate the recall of prerequisites.

4. Introduce the new stimulus material.

5. Provide learning guidance.

6. Have the user respond to a question.

7. Provide feedback on the user's response.

8. Evaluate the user's success.

9. Provide for transfer and retention.

There are other cognitive based tutorial models that depart from the early behavioral-based tutorial models (Jonassen, 1988). A component common to many of the cognitive-based models is user control over certain aspects o the lesson. User control over lessons has been urged for some time (Grubb, 1969). Most tutorial models allow the user to control the pace of the lesson, but behavioral-based tutorials stop there. Some cognitive models allow the user to control other lesson aspects, such as ability to access a lesson overview, definitions of key terms, examples of concepts or rules, exercises allowing tests of knowledge or skill, and more elaborate explanations of content and criterion tests.

Research on user control remains mixed, failing to demonstrate a clear-cut advantage to giving users free reign over all aspects of the instruction (Merrill, 1980; Steinberg, 1977). Subsequent studies have refined the concept of user control from total control to control over certain aspects of lessons and have reported success with this approach (Rubincam and Olivier, 1985). Goetzfried and Hannafin (1985) extend research on user control by including the factors of prior achievement and advisement in a study of control options in a lesson. They find that instructional time and learning efficiency are affected by the control options. The linear, non-linear control option is the most efficient control strategy, requiring less time than either an adaptive or advisement strategy. There are no differences in overall achievement due to any one of the three treatments. Gay (1986) reports an interaction of prior understanding of users and success in a study of user control when using interactive video. Users with little prior knowledge of the instructional content are not as capable of making decisions in the user control option as users with higher levels of prior knowledge.

Johansen and Tennyson (1983) report that the poor outcomes from research on user control of instruction relate to the cognitive complexity required by the learning of more elaborate tasks. User control systems place a burden on the user's cognition, particularly the meta-cognition or the awareness and control of one's own cognition. In a series of studies, Tennyson and his colleagues demonstrated the efficacy of providing advisement to users in user controlled CAI. Johansen and Tennyson (1983), Tennyson (1981), and Tennyson and Buttrey (1980) found support for the position that users profit from some degree of advisement when given control.

These research studies do not support blanket statements about the superiority of user control CAI over other forms; however, these studies do indicate that user control is effective in certain instances.

There are many other research studies with cognitive bases that can provide guidance for developing CAI lessons. The research has implications for design, such as screen design, use of graphics, and user-computer interactions, apart from the particular model used. Screen design and graphics will be reviewed in Section 5 and interaction in Section 7.

3.3 SIMULATION

While simulation is the most complex form of CAI, the underlying model is very simple and consistent (Lillie, Hannum, and Stuck, 1989). The user is presented with a scenario, such as: (1) an emergency room physician who is to treat a patient who has just been brought in, (2) an executive of a manufacturing firm who must make decisions about running a business, or (3) a counselor doing a diagnostic interview with a client. The user is given initial information about the situation and is asked to make decisions. When a decision is made, the software computes changes in the status of the situation (e.g., the patient's condition or business profits) and displays the results to the user. The users repeat the cycle of status, decision, status, through many iterations.

Several authors indicate that simulations are particularly appropriate for teaching higher order skills like problem solving. Kinzer, Sherwood, and Bransford (1986) report that simulations have an advantage over other forms of CAI because the user becomes an active decision maker. The authors view simulations as realistic and motivating to users and indicate the advantages of simulations in teaching problem solving.

Although computer-aided simulations are widely used in education and training, there is a lack of research substantiating the efficacy of these simulations (Alessi, 1988). Authors report that simulations are effective but often base their statement on anecdotal rather than empirical evidence. Because the model for simulations is so simple, generic, and adaptable to a variety of circumstances, few variations exist. The only enhancement of note is the embedding of tutorial segments in the simulation. If a user goes far astray, making bizarre responses, the simulation can activate a tutorial to explain some of the basic concepts contained in the program. Proponents of simulations claim that one advantage of simulations is that the user is given the opportunity to experiment with various approaches. Including a tutorial within the simulation alters the basic concept of a simulation as a discovery learning or problem solving exercise and changes it into a "get the right answer" exercise. Purists oppose such modifications to the basic simulation model.

4. SCREEN DESIGN RESEARCH

There have been numerous studies related to the design of instructional displays in general (Fleming and Levie, 1978) and the design of CAI displays in particular (Lillie, Hannum, and Stuck, 1989). Considerable research supports the contention that the screen design affects subsequent learning (Fleming and Levie, 1978; Grabinger, 1985; Morrison, Ross, and O'Dell, 1988). A summary of this research and its implications for the design of enhanced CAI courseware is presented in this section.

4.1 LAYOUT

One factor that can influence the impact of CAI is the layout of the screens (Jonassen and Hannum, 1987; Kerr, 1986). Users' attention is highly selective and should be focused on salient features of the instructional stimulus (Fleming, 1987; Gagne, 1985). One way to direct attention is with a liberal use of *white space* or unused screen. Kerr (1986) indicates that the amount of blank space on a screen has a major influence on how users perceive the material and its ease of use. Heines (1984) suggests using only a portion of the computer screen, not every line and every column. A sparsely filled screen helps users attend to the information displayed and prevents sensory overload. Text density also influences the quality of CAI screens (Tullis, 1983). Various rules of thumb appear in the literature regarding how much white space should be included in a CAI screen. At least one third of the screen should be free from text or graphic images, thereby allowing the user to focus on relevant information and not on screen clutter.

Research on screen layout indicates that a *consistent layout* or recurring pattern enhances the readability of screens and improves learning (Jonassen and Hannum, 1987; Kerr, 1986). The use of a similar format for each screen or for each screen type reduces the user's burden of having to decode each screen. Faiola and DeBloois (1988) suggest the use of a grid system to lay out screens in a consistent fashion. Organized displays have been shown to enhance learning (Winn, 1981). Other studies have shown that less mental processing on the part of the user is required when displays are familiar (Haber and Hershenson, 1973). Given the constraints of short-term memory, it is important to organize screens in a consistent fashion. It is generally recognized that individuals can hold up to seven items in short-term memory at any point in time. By relating like items, individuals can functionally expand the capacity of their short-term memory (Gagne, 1985). Thus, ordering or grouping stimuli can facilitate operations in short-term memory. Fleming (1980) states that users attend to order in displays; they seek to find it. Streibel (1984) indicates that the use of principles from research in the layout of text for a computer screen is even more important than layout for print or other visual displays due to the lower resolution and smaller size of computer displays. By ordering information on computer screens in a consistent manner, designers of CAI lessons can influence the amount of learning that will occur from their lessons (Faiola and DeBloois, 1988).

Another important aspect of screen design is the use of *functional areas* of the screen. Heines (1984) suggested that different screen components should be located at consistent places on the screen. Screen components include orientation information, directions to the user, user responses, feedback, error messages, and user options (Heines, 1984; Isaacs, 1987).

Designers can apply these suggestions in several ways. Functional areas can be defined by using a split screen technique, which creates a reserved section of the screen, fixed in size and location. The reserved space is only used for a specific purpose (e.g., if the functional area is dedicated to feedback, then only feedback should be displayed there, and if a particular lesson screen does not have any feedback, the functional area should remain blank).

A variation on the split screen technique was suggested by Jenkins (1982). In this variation, the functional areas or windows are fixed for an individual screen but may vary when screens change (Isaacs, 1987). Any time a particular functional area appears on a screen, it would be located in a standard position. For example, assume that an error message is located on line 21, indented 5 spaces, and framed in a box with a black background the first time it appears; it should be in the same location on any subsequent appearance.

Another technique for establishing functional areas is the ad hoc method in which windows are created when needed on an ad hoc basis and removed when not needed. When an ad hoc area is not in use as a window, it is free for other uses. The advantage of the ad hoc method is that it allows more use of the screen. The disadvantage of this method is that when the window area is required for a special function, information in that location must be removed, causing the disruption of the flow and disturbing the remaining information on the screen (Isaacs, 1987).

Still another technique for establishing functional areas is the use of pop-up or pull-down windows. The convention for establishing windows has become more widespread following the introduction of the Macintosh computer by Apple. A pop-up or pull-down window appears on the screen at a fixed location when the user requests it. An advantage of this type of display is that the screen is relatively clean and totally available for use when the window is not activated. The size of the pop-up or pull-down window can vary from very small to the size of the whole screen. Information in a pop-up or pull-down window is more likely to be attended to because the user is in control of its access. Isaacs (1987) cites two disadvantages to pop-up or pull-down windows: they are difficult to program and are not appropriate for information that needs to be displayed constantly during a CAI lesson.

Screen layout can also be improved by the use of borders or boxes to frame text. Setting text off in boxes focuses the user's attention on that text and facilitates reading. Research has indicated that placing text in highlighted blocks aids users (Simpson, 1982). Consistent use of the boxing technique facilitates the user's grouping of like ideas or content. Murch (1973) found that items close together in displays are grouped in perception and memory.

4.2 GRAPHICS

The use of graphics in lessons can enhance both the learning and retention of subject matter knowledge (Kearsley and Hillelsohn, 1982). To be effective, the graphic image must be related to the instructional content (Lillie, Hannum, and Stuck, 1989). Pictures seem to be retained longer than textual descriptions (Baily, 1982; Fleming and Levie, 1978; Gagne and Rohwer, 1969). Phillips (1986) reports that graphics can serve as "memory and thinking aids." Graphics must have some instructional purpose, including them just for the sake of having graphics in the lesson is not recommended. Research has shown that the simple addition of graphics to a textual presentation does not facilitate learning (King, 1976; Moore, Nawrocki, and Simutis, 1979).

Various types of graphic images can be effective in promoting learning, although elaborate graphic displays have not been shown to be superior to simpler graphic displays. Wileman (1980) classifies graphic images along a continuum from simple to complex. The simple end of the Wileman scale includes images composed of words. Realistic images are at the upper end of the scale and are essentially photographic representations of objects. Concept-related graphics lie between the extremes and are abstract images that relate directly to the concept illustrated. It is not essential to have a high degree of realism in graphic images (Winn, 1982). Simple line drawings can be effective stimuli for instruction (Dwyer, 1971, 1978). Thus, the more appropriate graphic images may be those on the mid-range of Wileman's continuum. Dwyer's finding is especially important for CAI since many computer displays are not capable of portraying high resolution images.

Graphics organize instructional content and exhibit relationships among items, thus facilitating both the initial acquisition and retention of instructional content (Fleming, 1987; Hawk, McLeod, and Jeane, 1983; Moore and Readance, 1983; Resnick, 1981; Winn, 1981). Drawings can be used to show relationships among items of content (Bork, 1977). Graphics can be used to provide cues to aid the user in processing new information. Jay (1983) suggests that graphics can be used effectively to emphasize important information and to provide cues to direct users' attention. Research on the use of cues as learning aids has provided support for the efficacy of cues (Bovy, 1981). Other research has shown that pictures without cues or prompts are more likely to be examined in a superficial manner (Levie and Lentz, 1982).

In a study contrasting well designed and poorly designed displays, Duin (1988) found that users using well designed displays with graphics had less difficulty using CAI programs, required less assistance, had more positive attitudes, and greater learning gains. Graphic images can also be used to guide users through a lesson. Both Benson (1985) and Feinburg (1984) advocate the use of visual images as a road map or guide through instructional material.

Finally, graphic images may be used for feedback and motivation. Malone (1981) includes the use of imagery among several conditions that contribute to user motivation. Jensen (1985) reports that motivation can be enhanced through the use of graphics, and that graphics can be effective in gaining the user's attention. Keller and Suzuki (1988) indicate that graphics can contribute to motivation through arousal. Research has demonstrated that users will use computer displays that incorporate graphics more readily than nongraphic displays and prefer such displays (Carey and Siegeltuch, 1982; Irving, Elton, and Siegeltuch, 1982). Tullis (1981) reports that users in CAI lessons ask for, and more readily use, displays that contained graphics. Contrary evidence about the effectiveness of graphic feedback is reported by Surber and Leeder (1988), who fail to find differential effects for text versus graphic feedback for maintaining users' interest. Feedback that is novel or surprising has been shown to increase users' motivation (Malone, 1981). Carefully planned, "unexpected" graphics can be used to contribute to user motivation.

A final point about the use of graphics involves the use of color. Color can be used to provide strong visual cues for linking related items, differentiating data, highlighting errors, and separating prompts, commands, input fields, and the like (Rambally and Rambally, 1987). The use of color is essential in lessons where the user must make discriminations based on the color of objects. Color is helpful when used selectively to direct attention or to group objects as on a map (Fleming, 1987). Others have found color not to be essential for instruction (Lamberski, 1980). The effects of color on learning seem to be overstated by some designers. Color does not appear to be essential to learning but, when used in an appropriate manner as suggested by Fleming (1987), color can contribute to the effectiveness of graphic displays.

5. INSTRUCTIONAL TEXT DESIGN RESEARCH

The design of instructional text in general and CAI text in particular has been the subject of considerable research and has direct implications for CAI design (Duffy and Waller, 1985; Gillingham, 1988; Jonassen, 1985). While there is a lack of total agreement regarding the design of instructional text, there is an agreement regarding features of instructional text that facilitate learning (Anderson and Armbruster, 1985). Suggestions for the design of text have been summarized by Gillingham (1988), Hannum (1986), Hartley (1981), and Wright (1981).

5.1 TEXT

Effective instructional text is well organized and structured (Goetz and Armbruster, 1980; Mayer, 1979; Van Dijk, 1979). The organization is made explicit to the users through information maps (Horne, 1985), graphic organizers (Hannafin, 1987; Hawk, McLeod and Jeane, 1983; Moore and Readance, 1983), typographical cues (Glynn, Britton, and Tillman, 1985) and the use of headings (Hartley and Jonassen, 1985). Anderson and Armbruster (1985) describe several methods for emphasizing the organization of a lesson, such as providing explicit statements about structure of the text, including introductory and summary statements, and providing pointer words or textual cues.

Carter (1985) suggests that instructional text should use examples, analogies, metaphors, illustrations, and flowcharts to simplify and solidify the content. Gagne and Rohwer (1969) point to the need for making instructional displays concrete and Hannafin and Peck (1988) recommend the use of text for similar purposes.

Jonassen (1985) refers to the use of structuring techniques in instructional text as mathemagenic activities, those activities that give rise to learning. In contrast with explicit activities built into the instructional text, Jonassen also describes generative activities. Generative activities encourage the user to interact with the instructional text in a variety of ways, such as requiring students to generate a mental image of the instructional content (Paivio, 1971; Pressley, 1977) or to develop mnemonics as a way to recall the instructional content (Bellezza, 1981; Higebee, 1979). Other generative activities include having the user develop content elaborations (Bobrow and Bower, 1969), summarize the content (Jonassen, 1985), and generate questions about the content (Andre and Anderson, 1978). Embedding such generative strategies in the instructional text can facilitate learning.

Specific features of text design (Rambally and Rambally, 1987) and the typographic design (Faiola and DeBloois, 1988) influence the effectiveness of CAI. Research suggests that the use of upper case and lower case letters is superior to all upper case letters (Foster and Champness, 1982; Tinker, 1963). Streibel (1984) reports that short sentences and centered, well spaced text contribute to effective screens. Isaacs (1987) cautions against too much reliance on features like reverse video, underlining, flashing, italics, and color. These should be used sparsely for emphasis in CAI lessons.

5.2 COMMUNICATING INSTRUCTIONAL CONTENT

A variety of ways exist for communicating instructional content to users. Research has shown that the use of examples facilitates learning (Fleming and Levie, 1978; Gagne, Briggs, and Wager, 1988). Instruction can be facilitated through the inclusion of analogies or metaphors (Ausubel, Novak, and Hanesian, 1978), which allow understanding of new relationships by building on the user's existing knowledge. As previously stated, illustrations can aid the user in understanding new content (Carter, 1985). A flowchart can be very helpful in a lesson teaching knowledge of procedures. Statements of the concept, or rule to be learned, can facilitate acquisition of intellectual skills (Gagne, 1985). Reigeluth (1983) suggests carefully constructed content elaborations as a means to convey instructional content. Finally, there is support from a variety of theoretical viewpoints for providing feedback to exercises which users complete (Gagne, 1985; Skinner, 1986) and there is support for many different approaches to communicating instructional content. Several texts on the development of CAI have recommended approaches to communicating instructional content that include examples, analogies and metaphors, illustrations, flowcharts, rule statements, content elaborations, and exercises with feedback (Alessi and Trollip, 1985; Hannafin and Peck, 1988; Lillie, Hannum, and Stuck, 1989).

6. USER-COURSEWARE INTERACTIONS RESEARCH

One area of CAI that is fundamental to the effectiveness of CAI lessons is user-courseware interaction (Weller, 1988). There is considerable research that addresses user-courseware interactions (Hannum and Gleason, 1986). A primary attribute of computers as an instructional medium is their ability to provide for interaction (Hannafin and Peck, 1988). The quality of user-courseware interactions has a direct influence on the overall quality of the CAI lessons (Lillie, Hannum, and Stuck, 1989). Recommendations for improving user-courseware interactions appear frequently in the literature on CAI (Gold, 1984; Hazen, 1985; Jonassen and Hannum, 1987; Kearsley, 1985; Larsen, 1985). Research directed at supporting the design of effective user-courseware interactions is presented in this section.

6.1 STIMULUS

Instructional stimuli affects the quality of user-courseware interactions. Certain types of instructional stimuli promote improved interactions. The type and placement of questions in CAI lessons may influence the user-courseware interactions (Jonassen and Hannum, 1987). Questions that require low-level processing promote learning at that level but inhibit higher-level processing of instructional material (Mayer, 1975; Fleming and Levie, 1978). Studies have demonstrated that questions requiring application are preferable to questions that ask for recall of specific information (Andre and Anderson, 1978). Other studies have failed to find differential effects from higher order questions (Schloss et al., 1986). Questions presented prior to new instructional content have a facilitating effect on learning content that is specifically related to the questions and a negative effect on learning other material contained in the lesson (Klauer, 1984). Pre-learning questions that are more general and ask about relations in the content are more likely to have a positive effect on learning (Klauer, 1984; Mayer, 1975). Questions inserted following the new instructional content facilitate learning of related instructional material without depressing the learning of incidental instructional material (Richards, 1979).

The issue of whether users should be allowed to review text when answering post-test questions has not been resolved by research. Duchastel and Nungester (1984) report a facilitating effect of reviewing text before answering questions when compared to reviewing text without the questions. Others report that allowing users to go back over instructional material looking for answers limits users' mental processing and, therefore, their achievement (Schumacher, Moses, and Young, 1983). Regardless, instructional text accompanied by questions seems to be superior to text without questions (Schloss, Schloss, and Cartwright, 1984; Schloss, et al., 1986).

The use of organizers, particularly advance organizers, can assist users in relating new content to previously acquired content, thus promoting both the acquisition and retention of the new content (Ausubel, Novak, and Hanesian, 1978). Numerous research studies have found facilitating effects from the presence of advance organizers (Mayer, 1979), yet others question their effect (Barnes and

Clawson, 1975). In addition, other types of organizers can be used to facilitate learning (Jonassen, 1985). The use of outlines, cognitive maps, structured overviews, and graphic organizers can be incorporated into CAI lessons to facilitate learning in a manner similar to that of advance organizers (Jonassen and Hannum, 1987; Lillie, Hannum, and Stuck, 1989).

The quality of user-courseware interactions can be improved by the use of cues to focus the user's attention on critical aspects of the lesson (Jonassen and Hannum, 1987). Another aspect of cuing that has research support is the use of processing cues to aid users who may be unable to use learning strategies effectively (Bovy, 1981). These cues can increase the meta-cognitive activity of users and engage effective learning strategies, thus promoting more learning.

The difficulty level of the instructional material can affect the degree of motivation and the amount of effort put forth in a lesson (Hartley and Lovell, 1984; Malone, 1981). Lessons that are either too easy or too difficult will result in decreased levels of user motivation. When users are presented with moderate levels of difficulty, they complete the lessons in less time and with more accuracy than users presented with high difficulty levels (Hartley and Lovell, 1984). Users are more successful when lessons present a realistic challenge, and this success promotes further success by enhancing the user's self-efficacy (Keller and Suzuki, 1988; Martin and Briggs, 1986). Thus, the level of difficulty influences motivation and self-efficacy, which in turn contribute to user-courseware interactions.

The perceived degree of control over the lesson impacts the quality of user-courseware interactions (Keller and Suzuki, 1988). When users perceive that they can control some aspects of the courseware, they are more likely to persist. Keller and Suzuki do not argue for total user control over the courseware but suggest that users require enough control to feel "in charge" and not at the mercy of a rigid CAI lesson.

The sophistication of the CAI model used can influence the user-courseware interactions. Simple models, such as the basic drill and practice model, do not enhance interactions as much as sophisticated models for drill and practice that adapt to the user's responses (Merrill and Salisbury, 1984). Simulation models that involve users as active participants promote improved user-courseware interactions over passive CAI models (Driskell and Dwyer, 1984). More sophisticated CAI models that include techniques associated with gaming, such as score keeping and fantasy, lead to increased motivation and improved user-courseware interaction (Malone, 1981).

6.2 RESPONSE

The responses called for in CAI lessons will affect the quality of the user-courseware interactions. Responding can occur at three levels: attending, covert, and overt (Anderson and Faust, 1973). Attending involves reading through material and making no response. A covert response occurs when the users are asked a question but are not required to enter answers. Rather, the users are told to think about a response. An overt response occurs when a question is asked and the users are required to enter a response to the question. Behavioral conceptions of learning require overt responses during instruction (Skinner, 1968). However, Fleming (1987) reports that both covert and overt activity are successful in promoting learning. The essential idea is for the users to engage in the mental processing of the content; the actual entering of responses may not be essential (Fleming, 1987). Thus, either covert or overt responding is superior to simple attending. Overt responding has some additional advantages (Fleming and Levie, 1978); the responses are easier to evaluate and corrective feedback or reinforcement can be provided freely. Overt responses also allow the simplified collection of formative evaluation data.

The type of response required should match the instructional intent of the lesson (Kearsley and Hillelsohn, 1982; Jonassen and Hannum, 1987). The learning of conceptual material is enhanced when users are required to complete cognitive maps or representations of the relationships among the areas (Jonassen and Hannum, 1987). The learning of factual material is improved through repetition and through the use of mental images and mnemonics (Bower, 1970; Fleming, 1987; Stein and Trabasso, 1984).

Finally, the type of response device can influence user-courseware interactions. Keyboards can be effective response devices, and studies have demonstrated the effectiveness of keyboards is not diminished by a lack of typing skills (Morrill, Goodwin, and Smith, 1968). A single keystroke is easy and efficient as a method of responding but may limit the level and type of mental processing engaged in by the user (Jonassen and Hannum, 1987). Research has shown that a single keystroke has less or equal effectiveness when compared to constructed responses (Ullman and Sparzo, 1968). Research has demonstrated that questions calling for a single keystroke as a response may contain cues for the response and thus discourage users from paying attention to the text (Schwade, 1984).

Response devices other than keyboards have been shown to be effective in promoting smooth user-courseware interactions. Touch screens are effective alternatives to keyboards, especially when used in simulations to mimic buttons or knobs on equipment (Kearsley and Hillelsohn, 1982). A mouse, joystick, or trackball also can be effective response devices.

Regardless of the type of response device used, CAI lessons should allow users to alter or edit their responses. The user-courseware interactions are impeded when users are not given the opportunity to make changes to their initial input.

6.3 FEEDBACK

An important advantage of CAI is the ability to provide immediate feedback to users (Alessi and Trollip, 1985; Leiblum, 1982). Research has demonstrated repeatedly that feedback promotes learning (Anderson, Kulhavy, and Andre, 1971; Cohen, 1985; Kulhavy, 1977).

The type of feedback supplied can have an effect on learning. Research has shown that constructive feedback explaining how a user can correct an error is better than simple feedback that only indicates that the user is wrong (Collins, 1984; Malone, 1981). Simple confirmation is effective feedback when the user responds correctly but is less effective after incorrect responses (Fleming, 1987; Kulhavy, 1977). Attempts to motivate users by using positive comments in feedback generally have been unsuccessful (Mosley, et al., 1984). However, feedback that is surprising or novel has been shown to engage the user and motivate them to continue the instruction (Campanini, 1981).

There is a relationship between the type of learning and the timing of feedback. Immediate feedback is necessary for the learning of new content (Bardwell, 1981). Lower level objectives are more likely mastered with immediate feedback. Delayed feedback and end-of-session feedback are helpful when teaching higher order, more abstract material (Gaynor, 1981). Other research provides support for delayed feedback (Kulhavy and Anderson, 1972; Surber and Anderson, 1975).

6.4 LESSON CONTROL

The interactions of users with the courseware are influenced by the manner in which certain aspects of the lesson are controlled. There are three methods for controlling CAI lessons: program, user, and adaptive. In program control courseware, all the control decisions, such as content sequencing, have been made in advance and are programmed into the CAI lesson. In user controlled courseware, control decisions have been turned over to the user. In adaptive courseware, the control decisions are made in the courseware as the user goes through a lesson. The courseware adapts to the user's responses according to a series of rules that determine the relationship between the input and what should happen next in the lesson.

Numerous research studies have investigated program control, user control, and adaptive CAI models. No one method for controlling CAI lessons has been shown to be superior to other methods in all instances. There are, however, some consistent findings from the research regarding which type of CAI control is most effective in specific situations. User control is appropriate for some aspects of a lesson, while other aspects belong under adaptive control (Carrier, Davidson, and Williams, 1985; Schloss, Wisniewski, and Cartwright, 1988; Tennyson, Park, and Christensen, 1985; Tennyson and Rothen, 1979; Tennyson, Tennyson, and Rothen, 1980). Studies have demonstrated that program control is the best control option under certain circumstances (Steinberg, 1977; Tennyson, Tennyson, and Rothen, 1980). The research support for different methods of lesson control is presented below.

1. Certain aspects of a lesson have been found to be best under user control, including the amount of practice that the user receives (Judd, Bunderson, and Bessent, 1970). The instructional pace can be adequately determined by the users (Wittrock, 1978). Users with high prior knowledge demonstrate their ability to control the instructional strategy (Hansen, 1981). Users have been shown to be successful when given the option to control access to lesson overviews

(Campanizzi, 1979). Several studies have demonstrated that user control can be improved through advisement (Johansen and Tennyson, 1983; Tennyson and Buttrey, 1980).

2. Some parts of a lesson, such as difficulty level of the practice (Fisher, et al. 1975), instructional method (Carrier, Newell, and Lange, 1982; Tobias, 1972), lesson sequence, and structure, have been shown to be best when under program control (Ross and Rakow, 1981; Rubincam and Olivier, 1985).

3. Finally, adaptive control is appropriate for some aspects of instruction. The adaptive control of the context of the instruction to match characteristics of users has been shown to be effective (Ross and Rakow, 1981). Adaptive control of content sequence and display time was shown to be effective in a study by Tennyson, Park, and Christensen (1985). Ross (1984) in a series of studies found support for adaptive control of instructional support and context of lessons. Park (1984) conducted several studies of adaptive control of instructional sequences that found support for adaptive control.

While research does not point to one control option as superior, some trends have been found in the research. While user control is not as successful as some proponents had hoped, limited control is shown to be appropriate in certain circumstances. User control, when coupled with advisement, can be superior to user control without advisement. There are promising research findings for the use of adaptive control models in CAI courseware. Some control decisions should be left to lesson designers.

7. EMERGING TRENDS AND FUTURE DIRECTIONS

CAI has become a more ubiquitous instructional medium in recent years and promises to continue growing. Research has clearly demonstrated the instructional effectiveness of CAI, although the potentia, may not yet be fully realized. Much courseware is poorly designed and does not fully utilize the capabilities of the computer (Bialo and Erikson, 1985; Gonce-Winder and Walbesser, 1987; Bitter and Gore, 1984; Mandl and Hron, 1986). Komoski (1984) reports only five percent of the available CAI can be considered high quality courseware.

7.1 CAI COURSEWARE DESIGN TRENDS

Numerous authors suggest improving the effectiveness of CAI courseware by paying more attention to instructional design during courseware development stages. (Duchastel, 1986; Flouris, 1989; Gagne, Wagner, and Rojas, 1981; Montague, Wulfeck, and Ellis, 1983; Poppen and Poppen, 1988; Roblyer, 1983; Streibel, 1984; Weller, 1988). There is a trend in CAI courseware toward the increased application of principles from learning psychology and instructional design. In fact, Clark (1983) attributes the achievement gains often demonstrated in comparative studies of the effectiveness of CAI to the use of instructional design principles and not to characteristics of computers. Future improvements in CAI are very likely to be a result of concern for and improvements in the quality of courseware design rather than improvements in computing hardware (Dudley-Marling and Owston, 1987). The trend toward the use of instructional design principles in developing courseware has begun and will likely grow stronger.

A related design trend is development of theoretically-based approaches to courseware design. Advances in cognitive psychology are influencing CAI research and development (Jonassen, 1988; Tennyson, et al., 1985). Current CAI courseware is based on behavioral notions of learning that are being questioned by advances in cognitive theory (Bonner, 1988; Low, 1981; Shuell, 1987; Wildman, 1981). This suggests a shift in the theoretical basis of CAI to cognitive psychology that is influencing both research and practice. In the future, courseware that reflects advances in cognitive psychology will be developed. Indeed, cognitive principles of courseware development have already been defined and recommended for use in CAI (Hannafin and Peck, 1988).

A technology likely to have a big impact on CAI is artificial intelligence (AI). Considerable resources have been expended for developing more intelligent computer software for a variety of applications. AI technology has been extended to developing CAI systems (Dede, 1987; Mackay, 1988; Stubbs and Piddock, 1985). The concept of applying AI to CAI is not very recent, having been suggested two decades ago by Carbonell (1970). There are several early examples of AI being applied in CAI models (Brown, Burton, and Bell, 1975; Sleeman and Brown, 1982). The amount of effort being expended in developing AI applications for training is changing (Dede, 1987). Early courseware had little "intelligence" built into it (Pliske and Psotka, 1986).

Developers of the early courseware planned everything that was to happen during the lesson. In contrast, intelligent courseware "learns" during the course of user-courseware interaction and modifies its presentation accordingly (Park, 1988). In intelligent CAI, the software constructs a model or representation of what the user is thinking and responds by altering the instructional approach (Sleeman and Brown, 1982; Resnick and Johnson, 1988). Thus, the computer program modifies itself as the user interacts with a lesson and determines whether to give a hint, give a different explanation, or ask a different question. There are several examples of intelligent tutors in use such as Scholar, Sophie, Buggy, and Guidon (Anderson and Reiser, 1985; Anderson, Boyle, and Yost, 1985; Tennyson, Christensen, and Park, 1984).

The application of AI techniques to CAI is growing and will likely continue (Dear, 1986; Dede, 1987). A number of different intelligent tutoring system frameworks have emerged (Becker, 1987; Mandl and Lesgold, 1988; Tchogovadze, 1986). Dede (1986), in a report on the research on intelligent CAI, indicated that although several issues remain to be addressed, intelligent CAI has been demonstrated to be a viable technology. Dede also indicated that intelligent instructional systems are likely to "make a significant improvement in educational quality because their design incorporates powerful features previously unrealized in learning technologies." Intelligent CAI represents an improvement over traditional CAI and will require different design approaches (Dede and Swigger, 1988). The impact of AI on CAI will most likely increase in the future, altering the future's concept of CAI.

Another trend that is affecting training and CAI is that of expert systems (Ahlers, Evans, and O'Neil, 1986). A part of AI, expert systems seek to extract and capture an expert's knowledge in a computer program, allowing less-trained people to improve their performance by accessing the expert's knowledge (Hayes-Roth, 1984; Kearsley, 1985; Michie, 1979; Parsaye, 1985). Several applications of expert systems to training have been suggested. Jonassen (1988) includes expert systems as an application of AI to training. Other applications include expert systems for automating instructional design, media selection, task analysis, needs assessment, and for tutoring (Kearsley, 1986; Merrill and Wood, 1984). One type of expert system "explains" its reasoning to the user, thus doing double duty by both enhancing the user's performance and increasing the user's knowledge.

Numerous hardware trends will have a positive impact on CAI. As computer hardware prices are falling, their capabilities are increasing. Microcomputers now have larger memory capabilities than they had even a few years ago. Many microcomputers exceed the capabilities of the mainframe computers of just a few years ago. Micros with one megabyte of memory are common and four to six megabyte machines are available. Storage capacity of microcomputers, with 40-60 megabytes of hard disk storage and high density floppy disk drives, has vastly increased. Optical storage devices such as compact disc-read only memory (CD-ROM) increase the ability of microcomputers to manipulate vast amounts of stored information. Interactive optical discs, compact disc-interactive (CD-I), are increasing the capacity of courseware. Very sophisticated hardware is now available to support more sophisticated CAI models and more elaborate multimedia courseware. Higher resolution graphics and the integration of video and computer output have enhanced CAI displays. The interaction of users and courseware is changing as a result of advancements in speech recognition and synthesized speech.

Computing and communications hardware is causing the distinction between work and training to become fuzzy and perhaps even nonuseful as instruction is being built into applications software used at work stations. By embedding training directly into applications software, workers move seamlessly in and out of training while routinely performing their jobs. Embedded or concurrent CAI is likely to increase rapidly in the next few years. Future training will be on-line and on demand. To some extent, this will cause a reconceptualization of CAI from stand-alone packages to parts of larger applications.

Technology now exists that integrates a variety of media and allows the user to move freely through instructional material. The technology is called hypermedia or hypertext and is beginning to have a substantial impact on delivering training via a computer. The basic structure of hypertext and hypermedia is a series of nodes which contain information and connecting links. Hypertext nodes contain text information; hypermedia nodes can contain information in the form of graphics, digitized speech, audio recordings, pictures, animation, film clips, and other sensory media. Hypertext is suggested to be appropriate for instructional applications by Tsai (1988). Hypertext moves away from fixed instructional sequences and pre-programmed instructional approaches to a freer form of courseware which is more consistent with a cognitive view of learning (Jonassen, 1986).

While there undoubtedly are other developments that might impact CAI in the future, the developments or trends identified here will likely have the strongest influence. The concept of CAI has evolved considerably since CAI first appeared three decades ago. Hundreds of studies have examined different aspects of instruction delivered via computer. Over three hundred articles were reviewed in the preparation of this report. The knowledge base for what constitutes effective CAI is large and growing.

7.2 IMPLICATIONS FOR AN ADVANCED CAI MODEL

There is a gap between the current practice of CAI and research findings. Most models fail to take advantage of the knowledge base that currently exists. Indeed, a great deal of CAI courseware packages remain little more than "page turners," failing to take advantage of the accumulated research or the capability of computers. While there are some notable exceptions, such as the adaptive model of Tennyson, most CAI lags behind the state of the art by a considerable amount. Options for an advanced model, consistent with the available results from research conducted in the area of CAI, are described in Section 6 (*Advanced Computer Aided Instruction Model for the NALDA Project*).

8. REFERENCES

Ahlers, R. H., Evans, R. A., and O'Neil, H. F., "Expert Systems for Department of Defense Training," *Journal of Computer Based Instruction*, **13**(2), 2 (1986).

Alessi, S. M., "Fidelity in the Design of Instructional Simulations," *Journal of Computer Based Instruction*, **15**(2), 40-47 (1988).

Alessi, S. M. and Trollip, S. R., *Computer-Based Instruction: Methods and Development*, Prentice-Hall, Englewood Cliffs, NJ, 1985.

Anderson, J. R., Boyle, C. F., and Yost, G., "The Geometry Tutor," *Proceedings of the Ninth Annual Joint Conference on Artificial Intelligence*, 595-599, 1985.

Anderson, J. R. and Reiser, B. J., "The LISP Tutor," *BYTE*, 159-175, April 1985.

Anderson, R. C. and Faust, G. W., "The Science of Instruction and Learning," *Educational Psychology*, Dodd, Mead, New York, 1973.

Anderson, R. C., Kulhavy, R. W., and Andre, T., "Feedback Procedures in Programmed Instruction," *Journal of Educational Psychology*, 62, 148-156, 1971.

Anderson, T. H. and Armbruster, B. B., "Studies, Strategies and Their Implications for Textbook Design," in *Designing Usable Text*, ed. T. M. Duffy and R. Waller Academic, New York, 1985.

Andre, M. and Anderson, T. H., "The Development and Evaluation of a Self-Questioning Study Technique," *Reading Research Quarterly*, 14, 605-623, 1978.

Ausubel, D. P., Novak, J. D., and Hanesian, H., "A Cognitive View," *Educational Psychology*, Holt, Rinehart and Winston, New York, 1978.

Avner, A., Moore, C., and Smith, S., "Active External Control: A Basis for Superiority of CBT," *Journal of Computer Based Instruction*, **6**(4), 115-118 (1980).

Baily, R. W., *Human Performance Engineering: A Guide for System Designers*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

Bangert-Drowns, R. L., Kulik, J. A., and Kulik, C. C., "Effectiveness of Computer Based Education in Secondary Schools," *Journal of Computer Based Instruction*, 12, 59-68, 1985.

Bardwell, R., "Feedback: How Does It Function?" *The Journal of Experimental Education*, 4-9, 1981.

Barnes, B. R. and Clawson, E. V., "Do Advance Organizers Facilitate Learning? Recommendations for Further Research Based on an Analysis of 32 Studies," *Review of Educational Research*, 45, 637, 659, 1975.

Becker, L., "A Framework for Intelligent Tutoring Instructional Systems: An Artificial Intelligence Machine Learning Approach," *Programmed Learning and Educational Technology*, 24(2), 145-150 (1987).

Bellezza, F. S., "Mnemonic Devices: Classification, Characteristics and Criteria," *Review of Educational Research*, 51, 247-275, 1981.

Benson, P. J., "Writing Visually: Design Considerations in Technical Publications," *Technical Communications*, 32, 35-39, 1985.

Bialo, E. and Erikson, L., "Microcomputer Courseware: Characteristics and Design Trends," *AEDS Journal*, 18(4), 227-236 (1985).

Bitter, G. and Gore, K., *The Best of Educational Software for Apple II Computers*, SYBEX, Berkeley, 1984.

Bobrow, S. A. and Bower, G. H., "Comprehension and Recall of Sentences," *Journal of Experimental Psychology*, 80, 455-461, 1969.

Bonner, J., "Implications of Cognitive Theory for Instructional Design: Revisited," *Educational Communications and Technology Journal*, 36(1), 3-14 (1988).

Bork, A., "Learning Through Graphics," in *Computers and Communications*, ed. R. J. Seidel, Academic Press, New York, 1977.

Bovy, R. C., "Successful Instructional Methods: A Cognitive Information Processing Approach," *Educational Communications and Technology Journal*, 29, 203-217, 1981.

Bower, G. H., "Analysis of a Mnemonic Device," *American Scientist*, 58, 496-510, 1970.

Brown, J. S., Burton, R., and Bell, A., "SOPHIE: A Step Toward Creating a Reactive Learning Environment," *International Journal of Man-Machine Studies*, 7, 675-696, 1975.

Campanini, S., "Learning Characteristics of the Disadvantaged: Implications for CAI Lesson Design Helpful Hints for Computers and Other Teachers," *Studies in Language Learning*, 3(1), 214-224 (1981).

Campanizzi, J. A., "Effects of Locus of Control and Provision of Overviews in a Computer Assisted Instruction Sequence," *AEDS Journal*, 12(1), 21-30 (1979).

Carbonell, J. R., "AI in CAI: An AI Approach to CAI," *IEEE Transactions on Man-Machine Systems*, 11, 190-202, 1970.

Carey, J. and Siegeltuch, M., *Teletext Usage in Public Places, Research on Broadcast Teletext: Working Paper No. 8*, New York University, Alternate Media Center, New York, 1982.

Carrier, C., Newell, K., and Lange, A., "The Relationship of Learning Style to Preferences for Instructional Strategies," *Journal of Dental Education*, 46, 652-655, 1982.

Carrier, C., Davidson, G., and Williams, M., "The Selection of Instructional Options in a Computer Based Coordinate Concept Lesson," *Educational Communication and Technology Journal*, **33**(3), 199-212 (1985).

Carter, J. A., "Instructional User Feedback: A Literature Review with Implications for Software Development," *The Computing Teacher*, 12(5), 53-55 (1984).

Carter, J. F., "Considerations in the Development of Distance Education Texts," in *The Technology of Text: Principles for Structuring, Designing and Displaying Text*, (2nd ed.), ed. D. H. Jonassen, Educational Technology Publications, Englewood Cliffs, NJ, 1985.

Chambers, J. A., and Sprecher, J. W., "Computer Assisted Instruction: Current Trends and Critical Issues," *Communications of the ACM*, **23**(6), 232-243 (1980).

Clark, R. E., "Evidence for Confounding in Computer Based Instruction Studies: Analyzing the Meta-Analyses," *Educational Communications and Technology Journal*, **33**(4), 249-262 (1985).

Clark, R. E., "Reconsidering Research on Learning from Media," *Review of Educational Research*, **53**(4), 445-459 (1983).

Cohen, J., *Statistical Power Analysis for the Behavioral Sciences*, Academic Press, New York, 1977.

Cohen, V. B., "A Re-examination of Feedback in Computer Based Instruction: Implications for Instructional Development," *Educational Technology*, **25**(1), 33-37 (1985).

Collins, M. T., *The Effectiveness of Computer Delivered Correction Procedures on Low-Performing Secondary Students' Reasoning Skills*, Doctoral Dissertation, University of Oregon, 1984.

Dear, B., "Artificial Intelligence Techniques: Applications for Courseware Development," *Educational Technology*, **26**(7), 7-15 (1986).

Dede, C., "A Review and Synthesis of Recent Research in Intelligent Computer Assisted Instruction," *International Journal of Man-Machine Studies*, 24, 329-353, 1986.

Dede, C., "Artificial Intelligence Applications to High-Technology Training," *Educational Communication and Technology Journal*, **35**(3), 163-181 (1987).

Dede, C. and Swigger, K., "The Evolution of Instructional Design Principles for Intelligent Computer Assisted Instruction," *Journal of Instructional Development*, 11(1), 15-22 (1988).

Driskell, J. E. and Dwyer, D. J., "Microcomputer Videogame Based Training," *Educational Technology*, 24, 11-17, 1984.

Duchastel, P. C., "Intelligent Computer Assisted Instructional Systems: The Nature of User Control," *Journal of Educational Computing Research*, **2**(3), 379-393 (1986).

Duchastel, P. C. and Nungester, R. J., "Adjunct Question Effects With Review," *Contemporary Educational Psychology*, **9**(2), 97-103 (April 1984).

Dudley-Marling, C. and Owston, R., "The State of Educational Software: A Criterion-Based Evaluation," *Educational Technology*, **27**(3), 25-29 (1987).

Duffy, T. M. and Waller, R., *Designing Usable Texts*, Academic Press, New York, 1985.

Duin, A. H., "Computer Assisted Instructional Displays: Effects on Students' Computing Behaviors, Prewriting and Attitudes," *Journal of Computer Based Instruction*, **15**(2), 48-56 (1988).

Dwyer, F. M., "Color as an Instructional Variable," *Audio Visual Communications Review*, 19, 339-416, 1971.

Dwyer, F. M., *Strategies for Improving Visual Learning*, Pennsylvania State University, Learning Services, 1978.

Edwards, J., Norton, S., Taylor, S., Wess, M., and Van Dusseldorp, R., "How Effective is CAI? A Review of the Research," *Educational Leadership*, 33, 147-153, 1975.

Faiola, T. and DeBloois, M. L., "Designing a Visual Factors-Based Screen Display Interface: The New Role of the Graphic Technologist," *Educational Technology*, **28**(8), 12-21 (1988).

Feinburg, S., "Visual Patterns: An Experiment with Technically Oriented Writers," *Technical Communication*, 31, 20-22, 1984.

Fisher, M., Blackwell, L., Garcia, A., and Greene, J., "Effects of Student Control and Choice on Engagement in a CAI Arithmetic Task in a Low Income School," *Journal of Educational Psychology*, **67**(6), 776-783 (1975).

Fleming, M. and Levie, W. H., *Instructional Message Design: Principles from the Behavioral Sciences*, Educational Technology Publications, Englewood Cliffs, NJ, 1978.

Fleming, M., "Displays and Communication," in *Instructional Technology: Foundations*, ed. R. M. Gagne, Lawrence Erlbaum, Hillsdale, NJ, 1987.

Fleming, M., "From Seeing and Hearing to Remembering: A Conception of the Instructional Process," *Instructional Science*, 9, 311-326, 1980.

Fleming, M., "Instructional Displays," in *Instructional Technology: Foundations*, ed. R. M. Gagne, Lawrence Erlbaum, Hillsdale, NJ, 1987.

Flouris, G., "The Use of an Instructional Design Model for Increasing Computer Effectiveness," *Educational Technology*, **29**(1), 14-21 (1989.

Foster, J. and Champness, B., "Attractiveness and Readability of Text and Tables," in *Videotext - Key to the Information Revolution*, Proceedings of Videotex '82, New York, 1982.

Fuller, K., "Beyond Drill and Practice: Using Computers to Build Hypotheses and Design and Conduct Experiments," *Science and Children*, **23**(7), 9-12 (1986).

Gagne, R. M., *The Conditions of Learning*, (4th ed.), Holt, Rinehart, and Winston, New York, 1985.

Gagne, R. M., Briggs, L. J., and Wager, W., *Principles of Instructional Design*, (3rd ed.), Holt, Rinehart, and Winston, New York, 1988.

Gagne, R. M. and Rohwer, W. D., "Instructional Psychology," *Annual Review of Psychology*, 20, 381-418, 1969.

Gagne, R. M., Wager, W., and Rojas, A., "Planning and Authoring Computer Assisted Instruction Lessons," *Educational Technology*, 21, 17-26, 1981.

Gay, G., "Interaction of User Control and Prior Understanding in Computer Assisted Video Instruction," *Journal of Educational Psychology*, **78**(3), 225-227 (1986).

Gaynor, P., "The Effect of Feedback Delay on Retention of Computer Based Mathematical Material," *Journal of Computer Based Instruction*, (2), 28-34, 1981.

Gillingham, M. G., "Text in Computer Based Instruction: What the Research Says," *Journal of Computer Based Instruction*, **15**(1), 1-6 (1988).

Glass, G. V., "Integrating Findings: The Meta-Analysis of Research," in *Review of Research in Education*, ed. L. Shulman, Peacock, Itasca, IL, 1977.

Glass, G. V., "Primary, Secondary and Meta-Analysis of Research, " *Educational Researcher*, 5, 3-8, 1976.

Glynn, S. M., Britton, B. K., and Tillman, M. H., "Typographical Cues in Text: Management of the Reader's Attention," in *The Technology of Text: Principles for Structuring, Designing and Displaying Text*, (2nd ed.), ed. D. H. Jonassen, Educational Technology Publications, Englewood Cliffs, NJ, 1985.

Goetz, E. T. and Armbruster, B. B., "Psychological Correlates of Text Structure," in *Theoretical Issues in Reading Comprehension: Prospectives from Cognitive Psychology, Artificial Intelligence, Linguistics and Education*, ed. R. J. Spiro, B. C. Bruce, and W. F. Bowers, Lawrence Erlbaum, Hillsdale, NJ, 1980.

Goetzfried, L. and Hannafin, M., "The Effects of Locus of CAI Control Strategies on the Learning of Mathematical Rules," *American Educational Research Journal*, **22**(2), 273-278 (1985).

Gold, P., "Educational Software -- New Guidelines for Development," *AEDS Journal*, **18**(1), 41-50 (1984).

Gonce-Winder, C. and Walbesser, H., "Toward Quality Software," *Contemporary Educational Psychology*, **12**(3), 261-268 (1987).

Grabinger, R. S., *CRT Text Design: Prominent Layout Variables Based on Factor Analysis of Models of Computer Generated Text*, Paper presented at the annual meeting of the Association for Educational Communications and Technology, Anaheim, CA, 1985.

Grubb, R. E., "User Controlled Statistics," in *Computer Assisted Instruction: A Book of Readings*, ed. R. C. Atkinson and H. A. Wilson, Academic Press, New York, 1969.

Haber, R. N. and Hershenson, M., *The Psychology of Visual Perception*, Holt, Rinehart and Winston, New York, 1973.

Hagler, P. and Knowlton, J., "Invalid Implicit Assumption in CBI Comparison Research," *Journal of Computer Based Instruction*, 14(3), 84-88 (1987).

Hannafin, M. J. and Peck, K. L., *The Design, Development and Evaluation of Instructional Software*, Macmillan, New York, 1988.

Hannafin, Michael., "The Effects of Orienting Activities, Cueing, and Practice on Learning of Computer-Based Instruction," *Journal of Education Research*, 81(1), 48-53 (1987).

Hannum, W. H., "Techniques for Creating Computer Based Instructional Text: Programming Languages, Authoring Languages and Authoring Systems," *Educational Psychologist*, 21(4), 293-314 (1986).

Hannum, W. H. and Gleason, J. M., *User-Courseware Interactions in CBI: A Selected Review of the Literature*, Paper presented at the annual meeting of the Association for the Development of Computer Based Instructional Systems, New Orleans, 1986.

Hannum, W., *Advanced Instruction Model for the NALDA CAI*, Oak Ridge Associated Universities, 1990.

Hansen, F. C., "An Evaluation of Computer Assisted Instruction (CAI) for Teaching Statistics to Social Work Students," *Dissertation Abstracts International*, 42(10), 4589A (1981).

Hart, F. A., "Computer-Based Training," in *Training and Development Handbook*, ed. R. Craig, McGraw-Hill, NY, 1987.

Hartley, J., "Eighty Ways of Improving Instructional Text," *IEEE Transactions on Professional Communications*, 24, 17-27, 1981.

Hartley, J. and Jonassen, D. H., "The Role of Headings in Printed and Electronic Text," in *The Technology of Text: Principles for Structuring Designing and Displaying Text*, (2nd ed.), ed. D. H. Jonassen, Educational Technology Publications, Englewood Cliffs, NJ, 1985.

Hartley, J. R. and Lovell, K., "The Psychological Principles Underlying the Design of Computer Based Instructional Systems," in *Instructional Software: Principles and Perspectives for Design and Use*, ed. D. F. Walker and R. D. Hess, Wadsworth, Belmont, CA, 1984.

Hawk, P., McLeod, N. P., and Jeane, K. L., "A Study of Graphic Organizers with College Level Geography Students," *Capstone Journal of Educational Research*, 2(2), 70-78 (1983).

Hayes-Roth, F., "Engineering Systems of Knowledge: The Great Adventure Ahead," in *Expert Systems in Government Symposium*, ed. K. N. Karna, McLean, VA, 1984.

Hazen, M., "Instructional Software Design Principles," *Educational Technology*, 25(11), 18-23 (1985).

Heines, J. M., *Screen Design Strategies for Computer Assisted Instruction*, Digital Press, Bedford, MA, 1984.

Hickey, A. E., *Computer Assisted Instruction: A Survey of the Literature*, Entelek, Newburyport, MA, 1968.

Higebee, K. L., "Recent Research on Visual Mnemonics: Historical Roots and Educational Fruits," *Review of Educational Research*, 49, 611-629, 1979.

Hofstetter, F. T., "Perspectives on a Decade of Computer Based Instruction, 1974-1984," *Journal of Computer Based Instruction* 12(1), 1-7 (1985).

Horne, R. E., "Results with Structured Writing Using the Information Mapping Writing Service Standards," in *Designing Usable Texts*, ed. T. M. Duffy and R. Waller, Academic Press, New York, 1985.

Irving, R., Elton, M., and Siegeltuch, M., *The Last Five Months of a Pilot Teletext Service: Interim Results on Utilization and Attitudes*, Research on broadcast teletext: Working paper No. 7., New York University, Alternate Media Center, New York, 1982.

Isaacs, S., "Text Design for Computer Assisted Learning," *British Journal of Educational Technology*, 18(1), 41-51 (1987).

Jamison, D., Suppes, P., and Wells, S., "The Effectiveness of Alternative Instructional Media: A Survey," *Review of Educational Research*, 44, 1-61, 1974.

Jay, T., "The Cognitive Approach to Computer Courseware Design and Evaluation," *Educational Technology*, **23**(1), 22-26 (1983).

Jenkins, J. M., "Some Principles of Screen Design and Software for their Support," *Computers and Instruction*, 6, 25-31, 1982.

Jensen, C., "What Have We Learned in the Past to Improve the Future of Courseware Design," *AEDS Journal*, **19**(1), 28-48 (1985).

Johansen, K and Tennyson, R., "Effect of Adaptive Advisement on Perception in User Controlled, Computer Based Instruction Using a Rule Learning Task," *Educational Communication and Technology*, **31**(4), 226-236 (1983).

Jolicoeur, K. and Berger, D., "Do We Really Know What Makes Educational Software Effective? A Call for Empirical Research on Effectiveness," *Educational Technology*, **26**(12), 7-11 (1986).

Jonassen, D. H., "Hypertext Principles for Text and Courseware Design," *Educational Psychologist*, **21**(4), 269-292 (1986).

Jonassen, D. H., *Instructional Designs for Microcomputer Courseware*, Lawrence Erlbaum, Hillsdale, NJ, 1988.

Jonassen, D. H., *The Technology of Text: Principles for Structuring, Designing and Displaying Text*, (2nd ed.), *Educational Technology Publications*, Englewood Cliffs, NJ, 1985.

Jonassen, D., and Hannum, W., "Research Based Principles for Designing Computer Software," *Educational Technology*, **27**(12), 7-14 (1987).

Judd, W. A., Bunderson, C. V., and Bessent, E. W., *An Investigation of the Effects of User Control in Computer Assisted Instruction Prerequisite Mathematics*, MATHS Technical Report 5., University of Texas, Austin, TX, 1970.

Kearsley, G., "Automated Instructional Development Using Personal Computers: Research Issues," *Journal of Instructional Development*, **9**(1), 9-15 (1986).

Kearsley, G., "Microcomputer Software: Design and Development Principles," *Journal of Educational Computing Research*, 1(2), 209-220 (1985).

Kearsley, G. and Hillelsohn, M. J., "Human Factors Considerations for Computer Based Training," *Journal of Computer Based Instruction*, **8**(4), 74-84 (1982).

Keller, J. M. and Suzuki, K., "Use of the ARCS Motivation Model in Courseware," in *Instructional Designs for Microcomputer Courseware*, ed. D. H. Jonassen, Lawrence Erlbaum, Hillsdale, NJ, 1988.

Kerr, S. T., "Instructional Text: The Transition from Page to Screen," *Visible Language*, **20**(4), 368-392 (1986).

King, W. A., *A Comparison of Three Combinations of Text and Graphics for Complex Learning* (NPRDC TR -76-16). Naval Personnel Research and Development Center, San Diego, CA, 1976.

Kinzer, C. K., Sherwood, R. D., and Bransford, J. D., *Computer Strategies for Education*, Merrill, Columbus, OH, 1986.

Klauer, K. J., "Intentional and Incidental Learning with Instructional Texts: A Meta-Analysis for 1970-1980," *American Educational Research Journal*, 21, 323-339, 1984.

Komoski, K., "Educational Computing: The Burden of Insuring Quality," *Phi Delta Kappan*, **66**(4), 244-248 (1984).

Kulhavy, R. W. and Anderson, R. C., "Delayed-Retention of Facts with Multiple-Choice Tests," *Journal of Educational Psychology*, **63**(5), 505-512 (1972).

Kulhavy, R. W., "Feedback in Written Instruction," *Review of Educational Research*, 47, 211-232, 1977.

Kulik, J. A., Bangert, R. L., and Williams, G. W., "Effects of Computer Based Teaching on Secondary School Students," *Journal of Educational Psychology*, **75**(1), 19-26 (1983).

Kulik, J. A., Kulik, C., and Bangert-Drowns, R., "Effectiveness of Computer Based Education in Elementary Schools," *Computers in Human Behavior*, 1, 59-74, 1985.

Kulik, J. A., Kulik, C. C., and Cohen, P. A., "Effectiveness of Computer Based College Teaching: A Meta-Analysis of Findings," *Review of Educational Research*, **50**(4), 525-544 (1980).

Lamberski, R. J., *A Comprehensive and Critical Review of the Methodology and Findings in Color Investigations*, Paper presented at the annual convention of the Association for Educational Communications and Technology, Denver, CO, 1980.

Larsen, R. (1985), "What Communications Theories Can Teach the Designer of Computer Based Training," *Educational Technology*, **25**(7), 16-19 (1985).

Leiblum, M. D., "Factors Sometimes Overlooked and Underestimated in the Selection and Success of CAL as an Instructional Medium," *AEDS Journal*, 67-77, 1982.

Lesgold, A. M. and Resnick, L. B., "How Reading Disabilities Develop: Perspectives from a Longitudinal Study," in *Theory and Research in Learning Disability*, ed. J. P. Das, R. Mulcahy, and A. E. Wall, Plenum, NY, 1982.

Levie, W. H. and Lentz, R., "Effects of Text Illustrations. A Review of Research," *Educational Communications and Technology Journal*, 30, 195-232, 1982.

Lillie, D. L., Hannum, W. H., and Stuck, G. B., *Computers and Effective Instruction: Using Computers and Software in the Classroom*, Longman, White Plains, NY, 1989.

Low, W. C., "Changes in Instructional Development: The Aftermath of an Information Processing Takeover in Psychology," *Journal of Instructional Development*, 4(2), 10-18 (1981).

Mackay, W. E., "Tutoring, Information Databases, and Iterative Design," in *Instructional Designs for Microcomputer Courseware*, ed. D. H. Jonassen, Lawrence Erlbaum, Hillsdale, NJ, 1988.

Malone, T. W., "Toward a Theory of Intrinsically Motivating Instruction," *Cognitive Science*, 4, 333-369, 1981.

Mandl, H. and Hron, A., "Information Technology," *EARLI News*, December 1986.

Mandl, H. and Lesgold, A. *Learning Issues in Intelligent Tutoring Systems*, Springer-Verlag, NY, 1988.

Martin, B. L. and Briggs, L. J., *The Effective and Cognitive Domains: Integration for Instruction and Research*, Educational Technology Publications, Englewood Cliffs, NJ, 1986.

Mayer, R. E., "Information Processing Variables in Learning to Solve Problems," *Review of Educational Research*, 4, 525-541, 1975.

Mayer, R. E., "Twenty Years of Research on Advance Organizers: Assimilation Theory is Still the Best Predictor of Results," *Instructional Science*, 8, 133-167, 1979.

Merrill, M. D., "User Control in Computer Based Learning," *Computers and Education*, 4, 77-95, 1980.

Merrill, M. D. and Wood, L., "Computer Guided Instructional Design," *Journal of Computer Based Instruction*, 11(2), 60-63 (1984).

Merrill, P. F. and Salisbury, D., "Research on Drill and Practice Strategies," *Journal of Computer Based Instruction*, **11**(1), 19-21 (1984).

Michie, D., *Expert Systems in the Micro-Electronic Age*, Edinburgh University Press, Edinburgh, 1979.

Montague, W. E., Wulfeck, W. H., and Ellis, J. A., "Quality CBI Depends on Quality Instructional Design and Quality Implementation," *Journal of Computer Based Instruction*, **10**(2), 90-93 (1983).

Moore, D. W. and Readance, J. E., *Meta-Analysis of Graphic Organizer Research*, Paper presented at the meeting of the American Educational Research Association, Montreal, Canada, 1983.

Moore, M. V., Nawrocki, L. H., and Simutis, Z., *The Instructional Effectiveness of Three Levels of Graphic Displays for Computer Assisted Instruction*, U.S. Army Research Institute for Behavioral and Social Sciences, Alexandria, VA, 1979.

Morrill, C. S., Goodwin, N. C., and Smith, S. L., "User Input Mode and Computer Assisted Instruction," *Human Factors*, **10**(3), 225-232 (1968).

Morrison, G. R., Ross, S. M., and O'Dell, J. K., "Text Density Level as a Design Variable in Instructional Displays," *Educational Communication and Technology*, **36**(2), 103-115 (1988).

Mosley, M. L., et al., *Computer Assisted Instruction and Continuing Motivation*, Paper presented at the annual meeting of the Association for Educational Communications and Technology, Dallas, TX, 1984.

Murch, G. M., *Visual and Auditory Perception*, Bobbs-Merrill, Indianapolis, 1973.

Orlansky, J., "Effectiveness of CAI: A Different Finding," *Electronic Learning*, 3(1), 58-60 (1983).

Paivio, A., *Imagery and Verbal Processes*, Holt, Rinehart and Winston, NY, 1971.

Park, O., "Empirically Based Procedures for Designing a Response-Sensitive Sequence in Computer Based Instruction: An Example from Concept Teaching Strategies," *Journal of Computer Based Instruction*, 11(1), 2-13 (1984).

Park, O., "Functional Characteristics of Intelligent Computer Assisted Instruction: Intelligent Features," *Educational Technology*, **28**(6), 7-14 (1988).

Parsaye, K., "The Evolutionary Road to Expert Systems," in *Expert Systems in Government Symposium*, ed. K. N. Karna, McLean, VA, 1985.

Phillips, R. J., "Computer Graphics as a Memory Aid and a Thinking Aid," *Journal of Computer Assisted Learning*, 2(1), 37-44 (1986).

Pliske, D. B. and Psotka, J., "Exploratory Programming Environments for Designing ICAI," *Journal of Computer Based Instruction*, **13**(2), 52-57 (1986).

Poppen, L. and Poppen, R., "The Use of Behavioral Principles in Educational Software," *Educational Technology*, **28**(2), 37-41 (1988).Prassley, M., "Imagery and Children's Learning," *Review of Educational Rosearch*, **47**(4) 585-622 (1977).

Rambally, G. and Rambally, R., "Human Factors in CAI Design," *Computers and Education*, 11(2), 149-153 (1987).

Reigeluth, C. M., *Instructional Design Theories and Models: An Overview of their Current Status*, Lawrence Erlbaum, Hillsdale, NJ, 1983.

Resnick, L. B., "Instructional Psychology," *Annual Review of Psychology*, 32, 659-704, 1981.

Resnick, L. B. and Ford, W. W., *The Psychology of Mathematics for Instruction*, Lawrence Erlbaum, Hillsdale, NJ, 1981.

Resnick, L. B. and Johnson, A., "Intelligent Machines for Intelligent People: Cognitive Theory and the Future of Computer Assisted Learning," in *Technology in Education: Looking Toward 2020*, ed. R. Nickerson and P. Zodhiates, Lawrence Erlbaum, Hillsdale, NJ, 1988.

Richards, J., "Adjunct Postquestions in Text: A Critical Review of Methods and Processes," *Review of Educational Research*, 49, 181-196, 1979.

Roblyer, M. D., "Toward More Effective Microcomputer Courseware Through Application of Systematic Instructional Design Methods," *AEDS Journal*, **17**(1-2), 23-32 (1983).

Roblyer, M. D., Castine, W. H., and King, F. J., "Assessing the Impact of Computer Based Instruction: A Review of Recent Research," *Computers in the Schools*, **5**(3/4), 1-149 (1988).

Roblyer, M. D. and King, F. J., *Reasonable Expectations for Computer Based Instruction in Basic Reading Skills*, Paper presented at the annual conference of the Association for Educational Communications and Technology, New Orleans, LA, 1983.

Ross, S., "Matching the Lesson to the Student: Alternative Adaptive Designs for Individualized Learning Systems," *Journal of Educational Instruction*, **11**(2), 42-48 (1984).

Ross, S. and Rakow, E., "User Control vs. Program Control as Adaptive Strategies for Selection of Instructional Support on Math Rules," *Journal of Educational Psychology*, 73, 745-753, 1981.

Rubincam, I. and Olivier, W., "An Investigation of Limited User Control Options in a CAI Mathematics Course," *AEDS Journal*, **18**(4), 211-226 (1985).

Salisbury, D. F., "When is a Computer Better than Flashcards?" *Educational Technology*, **28**(12), 26-32 (1988).

Salisbury, D. F., Richards, B. F., and Klein, J. D., "Designing Practice: A Review of Prescriptions and Recommendations from Instructional Design Theories," *Journal of Instructional Development*, **8**(4), 9-19 (1985).

Samson, G., Niemiec, R., Weinstein, T., and Walberg, H., *Effects of Computer Based Instruction on Secondary School Achievement: A Quantitative Synthesis*, Paper presented at the annual meeting of the American Educational Research Association, Chicago, IL, 1985.

Sawyer, T. A., "Human Factor Considerations in Computer Assisted Instruction," *Journal of Computer Based Instruction* **12**(1), 17-20 (1985)

Schloss, P., Schloss, C., and Cartwright, P., "Efficacy of Four Ratios of Questions and Highlights to Text in Computer Assisted Instruction Modules," *Journal of Computer Based Instruction*, **11**(4), 113-122 (1984).

Schloss, P. J., Wisniewski, L. A., and Cartwright, G. P., "The Differential Effect of User Control and Feedback in College Students' Performance on CAI Modules," *Journal of Educational Computing Research*, **4**(2), 141-149 (1988).

Schloss, P., Sindelar, P., Cartwright, P., and Schloss, C., "The Influence of Error Correction Procedures and Question Type on Student Achievement in Computer Assisted Instruction," *Journal of Educational Technology Systems*, **16**(1) 17-27 (1986).

Schumacher, G. M., Moses, J. D., and Young, D., "Students' Studying Process on Course Related Text: The Impact of Inserted Questions," *Journal of Reading Behavior*, 15, 19-36, 1983.

Schwade, S., "Why Keyboard Entry Convenience Should Not Be Your Only Concern When Choosing Courseware: The Advantages of the Constructed Response," *Educational Research*, **24**(8), 27-30 (1984).

Shuell, T. J., "Cognitive Conceptions of Learning," *Review of Educational Research*, 56, 411-436, 1987.

Simpson, H., "A Human Factors Style Guide for Program Design," *BYTE*, 7, 108-132, 1982.

Skinner, B. F., "Programmed Instruction Revisited," *Phi Delta Kappan*, **68**(2), 103-110 (1986).

Skinner, B. F., *The Technology of Teaching*, Appleton-Century-Crofts, NY, 1968.

Slavin, R. E., "Best-Evidence Synthesis: An Alternative to Meta-Analytic and Traditional Reviews," *Educational Researcher*, **15**(9), 5-11 (1986).

Slavin, R. E., "Best-Evidence Synthesis: Why Less is More," *Educational Researcher*, **16**(4), 15-16 (1987).

Sleeman, D. and Brown, J. S., *Intelligent Tutoring Systems*, Academic Press: NY, 1982.

Stein, L. L. and Trabasso, T., "What's In a Story: Critical Issues in Comprehension and Instruction," in *Advances in the Psychology of Instruction*, Vol. 2. ed. R. Glaser, Lawrence Erlbaum, Hillsdale, NJ, 1984.

Steinberg, E. R., "Review of Student Control in Computer Assisted Instruction," *Journal of Computer Based Instruction*, 3, 84-90, 1977.

Streibel, M., "Dialogue Design and Instructional Systems Design for an Intelligent Video Disk System," *Video Disc and Optical Disc*, 4(3), 216-229 (1984).

Stubbs, M. and Piddock, P., "Artificial Intelligence in Teaching and Learning: An Introduction," *Programmed Learning and Educational Technology*, 22(2), 150-157 (1985).

Surber, J. and Anderson, R., "The Delay Retention Effect in Natural Classroom Settings," *Journal of Educational Psychology*, 67, 170-173, 1975.

Surber, J. and Leeder, J., "The Effects of Graphic Feedback on Student Motivation," *Journal of Computer Based Instruction*, 15(1), 14-17 (1988).

Tchogovadze, G. G., "Some Steps Towards Intelligent Computer Tutoring Systems," *Educational and Computing*, 2(4), 273-277 (1986).

Tennyson, C. L., Tennyson, R. D., and Rothen, W., "Content Structure and Management Strategy as Design Variables in Concept Acquisition," *Journal of Educational Psychology*, 72, 491-505, 1980.

Tennyson, R. D., "Use of Adaptive Information for Advisement in Learning Concepts and Rules Using Computer Assisted Instruction," *American Educational Research Journal*, 4, 425-438, 1981.

Tennyson, R. D. and Buttrey, T., "Advisement and Management Strategies as Design Variables in Computer Assisted Instruction," *Educational Communications and Technology Journal*, 28, 169-176, 1980.

Tennyson, R. D., Christensen, D. L., and Park, S., "The Minnesota Adaptive Instructional System: A Review of Its Theory and Research," *Journal of Computer Based Instruction*, 11, 2-13, 1984.

Tennyson, R. D., Park, O. C., and Christensen, D. L., "Adaptive Control of Learning Time and Content Sequence in Concept Learning Using Computer Based Instruction," *Journal of Educational Psychology*, 77(4), 481-491 (1985).

Tennyson, R. D. and Rothen, W., "Management of Computer Based Instruction: Design of an Adaptive Control Strategy," *Journal of Computer Based Instruction*, 5, 63-71, 1979.

Tennyson, R. D., Welsh, J. C., Christensen, D. L., and Hajovy, H., "Interactive Effect of Information Structure Sequence of Information and Process Learning Time on Rule Learning Using Computer Based Instruction," *Educational Communication and Technology Journal*, 33(3), 213-223 (1985).

Thomas, D. B., "The Effectiveness of CAI in Secondary Schools," *AEDS Journal*, 12, 103-116, 1979.

Tinker, M. A., *Legibility of Print*, Iowa State University Press, Ames, Iowa, 1963.

Tobias, S., "When Do Instructional Methods Make a Difference?" *Educational Researcher*, 11, 4-9, 1972.

Tsai, C., "Hypertext: Technology, Applications and Research Issues," *Journal of Educational Technology Systems*, **17**(1), 3-14 (1988).

Tullis, T. S., "An Evaluation of Alphanumeric, Graphic and Color Information Displays," *Human Factors*, **23**(5), 541-550 (1981).

Tullis, T. S., "The Formatting of Alphanumeric Displays: A Review and Analysis," *Human Factors*, **25**(6), 657-682 (1983).

Ullman, J. D. and Sparzo, F. J., "Differential Effects of Two Types of Unit Quiz Modes on Final Exam Performance," *Journal of Educational Measurement*, 5, 169-173, 1968.

Van Dijk, T. A., "Relevance Assignment in Discourse Comprehension," *Discourse Processes*, 2, 113-125, 1979.

Weller, H., "Interactivity in Microcomputer Based Instruction: Its Essential Components and How It Can Be Enhanced," *Educational Technology*, **28**(2), 23-27 (1988).

Wildman, T. M., "Cognitive Theory and the Design of Instruction," *Educational Technology*, **21**(7), 14-20 (1981).

Wileman, R., *Exercises in Visual Thinking*, Hastings, NY, 1980.

Winn, W., "The Meaningful Organization of Content: Research and Design Strategies," *Educational Technology*, **21**(8), 7-11 (1981).

Winn, W., *The Status and Trends in Visual Information Processing*, Paper Presented at the Annual Meeting of the Association for Educational Communications and Technology, Dallas, TX, 1982.

Wittrock, M. C., "The Cognitive Movement in Instruction," *Educational Psychologist*, 15, 15-29 (1978).

Section 2

HYPERTEXT/HYPERMEDIA: ADVANCED TECHNOLOGY FOR COMPUTER AIDED INSTRUCTION

TABLE OF CONTENTS

LIST OF FIGURES

HYPERTEXT/HYPERMEDIA: ADVANCED TECHNOLOGY FOR COMPUTER AIDED INSTRUCTION

1. INTRODUCTION

Hypertext/hypermedia (HT/HM) show great promise for widespread applicability to the problems associated with seeking and developing information on a computer. Questions concerning the use of HT/HM include: What are the implications for developing CAI and other applications with HT/HM software? Will the potential benefits be worth the investment to switch to a technology that Jonassen (1988) contends is "theory rich and research poor"? This paper will provide information to help answer these questions by defining HT/HM, presenting historical data, giving examples of applications developed with HT/HM software, and citing implications for research and training.

1.1 DEFINITIONS

Hypertext is a relational database composed of nodes (chunks of data or data files) in which movement from one node to the next is made possible by links. Shneiderman and Kearsley (1989) state "the most common meaning of hypertext is a database that has active cross-references and allows the user to jump to other parts (nodes) of the database as desired." These active cross-references are defined as links (Conklin, 1987). Conklin contends that machine-supported links are the essential feature of hypertext as these links contain the programming (invisible to the user) necessary to provide the capacity to jump from node to node. Links can be depicted on the screen as icons (symbols) or words. The underlined words in Fig. 2.1 symbolize links.

> The purpose of the VIDS/MAF form is explained in Volume III of the NAMP manual. Examples of how to fill out the VIDS/MAF form and how it relates to the Aircraft Intermediate Maintenance Department (AIMD) and to the DEPOT are given.

Fig. 2.1. AV-3M data entry forms.

Selecting the link for display requires two steps. First, the user positions the cursor on the underlined word(s) by moving the cursor with a computer mouse or the keyboard cursor direction keys. The user then clicks the computer mouse on the word(s) (or presses the ENTER key) to view the information. For example, a user needs information about how to fill out a VIDS/MAF form. The cursor must be moved to the word Examples (Fig. 2.1). Next, the user chooses the word Examples and the requested information is displayed immediately. If information about the depot is desired, the selection of the word Depot results in that information being displayed. The user can move to and from each node at will and in any order, as shown in Fig. 2.2. A map similar to that shown in Fig. 2.2 could help to reorient users if they become lost. The map can be used as a shopper would use a mall's floor plan map to locate the current position and to decide on a direct route to a desired location.

A second way to define hypertext is by contrasting linear and nonlinear text presentation. Marchionini (1988) describes hypertext as "the electronic representation of text that takes advantage of the random access capabilities of computers to overcome the strictly sequential medium of print on paper."

T. H. Nelson, a pioneer in hypertext research, referred to nonlinear text as the organization of information which allows access quickly and by choice (Conklin, 1987). Nonlinear text is a contrast to linear text which forces users to read from top to bottom, front to back.

Currently, many computer programs are linear. Their structure is hierarchical and access is not flexible. Users cannot jump from file to file (node to node) without using hierarchical menus. Hypertext software allows the linking of nodes in a very flexible, directly accessible, nonlinear arrangement, which gives users quick response. This direct access allows the user to gain an understanding of other related information very quickly, gaining maximum utilization of short-term memory and a better transition to permanent memory.

Conklin (1987) describes the difference between hypertext and hypermedia by pointing out that advances in technology allow the information in the nodes to be not only text but also graphics, digitized speech, audio recordings, pictures, animation, film clips, and other sensory media. This capability is called hypermedia. "Hypermedia is simply multimedia hypertext" (Saffo, 1987).

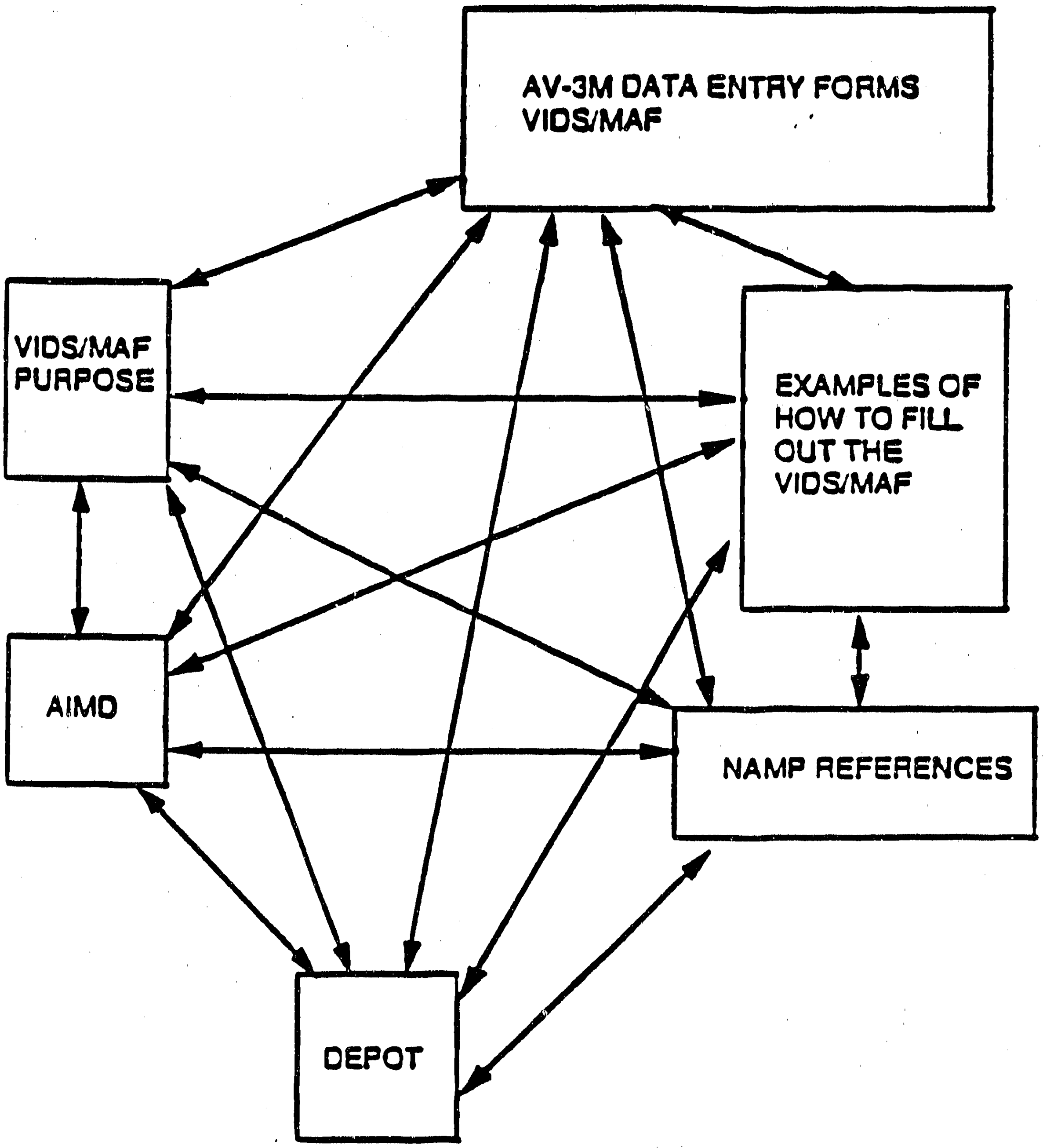


Fig. 2-2. Path map.

1.2 SELECTED HISTORY OF HYPERTEXT/HYPERMEDIA

In 1945, Vannevar Bush, a science advisor to President Roosevelt, wrote an article describing a mechanized way to retrieve information. Bush describes memex as a tool that provides access to a large collection of microfilm and mechanisms for making links between any two pieces of information in the system (Smith, 1988). Bush foresaw the information seeking needs of the modern world, a world that would soon be overwhelmed with vast amounts of information (Bush, 1945). This idea later inspired Douglas Engelbart's development of the first hypertext-like system at Stanford Research Institute in 1968 (Conklin, 1987). Engelbart originally named his system the On-Line System (NLS), later changing it to Augment (Delisle and Schwartz, 1987). NLS incorporated many original ideas now taken for granted, including electronic mail, windows, and the computer mouse, which Engelbart invented (Saffo, 1987).

During this same period T. H. Nelson, who coined the word hypertext (Conklin, 1987), envisioned "a universal library system" containing all the world's literature available on-line worldwide (Shneiderman and Kearsley, 1989). Nelson's system was named Xanadu. Nelson and Andy Van Dam worked together at Brown University on the Hypertext Editing System. Their system was used by the Houston Manned Spacecraft Center for Apollo documentation. This work gave impetus to van Dam and his students in designing the File Retrieval and Editing System (FRESS) (Smith, 1988). Brown University's current system, Intermedia, has also been the benefactor of Nelson and van Dam's design and research (Young, 1988).

Augment and Xanadu provided the base design model for later hypertext/hypermedia software. In 1983, Intermedia provided the application of hypertext to an integrated network of advanced personal computers establishing Brown University's Institute for Research in Information and Scholarship (IRIS). IRIS's goal was to have several work station classrooms in operation by 1988 to provide an atmosphere for research in academic HT/HM applications (Young, 1988).

2. CURRENT APPLICATIONS OF HYPERTEXT/HYPERMEDIA

Hardware, software, and technical advances impact HT/HM applications. If computer technology had remained at the level that existed when personal computers were first introduced in the sixties, only limited capacity hypertext would be available and hypermedia would be nonexistent.

2.1 HARDWARE AND HYPERTEXT/HYPERMEDIA APPLICATIONS

Once hypertext concepts were off the drawing board, early applications and research were hindered by the technology available. Supercapacity microcomputers were not in existence. With the advent of advanced technologies in data storage, computer memory, monitors, and color/graphics cards, applications and research began to uncover the many benefits of HT/HM and the ways information is obtained and processed.

Hypertext requires data storage media that allow immediate access, high volume, and durability at economical prices. Examples of advanced data storage media are Compact Disc-Read Only Memory (CD-ROM), Compact Disc-Interactive (CD-I), Digital Video-Interactive (DV-I), and Optical Storage Cards (Smart Cards). Hypermedia requires monitors that can display multicolor high-resolution graphic, video, or motion picture signals produced by advanced color/graphics cards.

HT/HM applications have immediate response times when there is sufficient computer memory to allow an extended collection of hypertext documents to remain in the background. With enhancements to the first generation of HT/HM developmental software, the need for increased computer memory is imminent because the improved capabilities of the software will require additional memory.

2.2 SOFTWARE AND HYPERTEXT/HYPERMEDIA APPLICATIONS

A software product produced with HT/HM developmental software is referred to as an application. Developmental software is sometimes called "authoring software" or an "authoring package." In this paper, the term HT/HM developmental software will be used when referring to HT/HM authoring software or authoring packages. Fiderio (1988) groups hypertext/hypermedia developmental software according to the way in which it is most often applied: problem resolution, on-line browsing, library or literary-exchange, and multipurpose.

2.2.1 Problem-Resolution Applications

The purpose of problem-resolution applications (PRAs) is to provide a computerized problem-solving atmosphere through capabilities such as super-efficient data entry, modification, structuring, printing, and computer presentations. Fiderio (1988) states that HT/HM developmental software packages used for PRAs "facilitate the organization of information gathered to solve problems, utilizing commands that create or modify internal links among concepts quickly."

An example of the types of problems that PRAs address is how to provide the computer atmosphere in which to facilitate the engineering of accounting software. The amount of programming data to accomplish this task requires a very large database. PRAs facilitate the entry of information in a manner similar to word processing. The benefits of using PRAs are the unrestricted flexibility for structuring information, direct and immediate access, easy modification, and the ability to process a large volume of information, all in the same database. Conklin (1987) notes that hypertext allows the collection of large amounts of relatively unstructured information but adds that such collections are inefficient unless software capabilities support information organizing, browsing, and filtering (how much information is viewed at any one time).

Two diverse examples of HT/HM developmental software packages used for problem-resolution type applications are Engelbart's Augment and the University of North Carolina's Writing Environment (WE) (Conklin (1987), Fiderio (1988)). Augment is a mainframe computer system whose primary application has been in software engineering and project management. WE is a large mainframe application used to research software authoring tools needed in a writer's environment (Conklin, 1987). The creation of both electronic and printed documents is the primary application of WE (Smith, Weiss, Ferguson, 1987).

Other noteworthy examples of problem-resolution HT/HM developmental software packages are Hypertext Bridge (developed by Neuron Data) and Instructional Design Environment (IDE) (developed by Xerox's Palo Alto Research Center). Hypertext Bridge is used in the development of expert system applications with a hypertext graphical user interface that is associated with Apple's HyperCard or SuperCard on Macintosh computers (Martin, 1989). IDE allows the instructional designer to perform all instructional analysis, design, and development within a single computing environment. IDE runs exclusively on Xerox Lisp computers (Jensen, Jordan, Russell, 1987).

2.2.2 On-line Browsing Applications

The purpose of on-line browsing applications (OLBs) is to provide a computer atmosphere for seeking information. Interactive OLBs (such as CAI and on-line reference manuals) developed with HT/HM developmental software packages need clear, understandable screen displays for presenting information and easy-to-operate browsing commands used for accessing information (Fiderio, 1988).

Ease of learning and use is stressed by Conklin (1987), citing Carnegie-Mellon University's ZOG and the University of Maryland's HyperTIES as two examples with these qualities. ZOG, installed on the nuclear powered aircraft carrier USS Carl Vinson in 1982, is commercially marketed as the Knowledge Management System (KMS). ZOG, a mainframe system, has been used primarily for CAI and information databases and on a large network to provide multiple authoring tools. ZOG is used to develop bulletin boards and textual databases, as a CAI tool, and for policy analysis, authoring, communications, and code management. HyperTIES, developed by Ben Shneiderman at the University of Maryland, is used as part of the Holocaust Exhibit at the Smithsonian Institute. Commercial versions run on personal computers. HyperTIES seems to be one of the top choices for CAI development and textual and visual database applications.

A list of other noteworthy OLBs developed with hypertext/hypermedia software follows:

1. The Navy On-board Maintenance Aid Device (NOMAD) is used to lead technicians through appropriate troubleshooting procedures (Stone et al., 1982). NOMAD was installed aboard the USS Kinkaid.

2. A Hypertext Electronic Job Aid for Maintenance helps Army personnel improve maintenance of complex equipment (Stone et al., 1982).

3. Stanford University's Medical School uses the Electric Cadaver to present computerized anatomy lessons and digital dissection (Jerome, 1989).

4. Window Book, an authoring system from Window Book, Inc., enables developers to take bulky, hard-copy documents and transform them into on-line hypertext manuals stored on floppy disk or CD-ROM (Pallatto, 1989).

5. The Service Bay Diagnostic System (SBDS) is a joint project between Ford and Hewlett-Packard to give mechanics on-line diagnostic guidance (Saffo, 1987).

6. The Department of Defense (DoD) Gateway Information System (DGIS), an artificial intelligence-based Common Command Language, utilizes hypermedia capabilities. DGIS's goal is to reduce the many database query languages that DoD personnel need to know to just one language, DGIS (Kuhn, 1988).

2.2.3 Library or Literary-Exchange Applications

Library or literary-exchange applications provide a computer atmosphere that allows users access to international literature. Users can write research papers within a system which automatically handle all legal matters such as copyright details. Several people can work on the same paper at the same time. Literary-exchange applications developed with HT/HM software feature complex multiple structure databases that store collaborative notes, research, electronic mail, documents, and entire libraries (Fiderio, 1988). Fiderio cites T. H. Nelson's Xanadu as an example. Xanadu's long-range goal is facilitating the accumulation of the entire world's literary corpus into one huge, on-line, interactive, hypertext database (Conklin, 1987).

Xanadu's primary applications are large mainframe textual databases and document creation and modification.

2.2.4 Multipurpose Applications

Multipurpose applications provide a wide range of HT/HM developmental software capabilities resident in a large networked system used to develop many different types of applications. Two multipurpose hypertext/hypermedia developmental software systems used in this manner are Notecards and Intermedia (Conklin, 1987). Notecards was developed at Xerox's Palo Alto Research Center and fits the needs of a large corporation or government by servicing a large network of mainframe users. The primary applications are authoring, programming, personal information management, legal research, engineering design, CAI, and studying hypertext. Brown University's Intermedia is an Apple Unix system hypermedia application utilizing the Apple II, IIx, and IIcx series as terminals. Intermedia is used for basic hypertext research and as a tool for professors to organize and present lesson material via computer. Intermedia is also an interactive medium for students to study materials, add annotations, or create path histories for later use.

2.3 COMMERCIALLY AVAILABLE HYPERTEXT/HYPERMEDIA SOFTWARE

Developers should carefully examine software to find tools that best suit identified needs. There is an absence of standardization among first generation HT/HM products. Some are hypertext only; others have both hypertext and hypermedia capabilities. Assorted hypermedia developmental software packages have good internal support for graphics; others are dependent on external graphic support. Devlin and Berk (1989) list the following personal computer (PC) HT/HM software:

1. LinkWay (IBM) is similar to Apple's HyperCard; it is mouse driven, with multimedia capabilities. Enhancements to the initial package will make this product worthy of consideration for CAI and other forms of computer training development.

2. HyperTIES (Cognetics Corporation) is one of the best products on the market but could benefit from a graphics browser (the capability to display maps of screens visited to help reorient users).

3. HyperBase (Cogent Software Ltd.) produces dynamic links allowing multiple link arrangements. In contrast, other hypertext software packages allow only one link arrangement between nodes. Dynamic linking allows documents to be modified to fit the skill level of the user. Future enhancements such as automatic indexers and links to CD-ROM and video will increase the use of this product.

4. Guide (OWL International) is good software that could benefit from enhancements such as better internal support of graphics, digital sound, animation, and links to CD-ROM and video.

5. WILDCARD (Spinnaker Software Corporation) is described by *PC WEEK* writer Lisa Picarille (1989) as giving "PC users full access to Apple Computer, Inc.'s, HyperCard environment."

Stevens (1989) lists the following developmental software packages, which are aimed at the educational market and designed for use on the Apple Macintosh.

1. HyperCard (Apple Computer, Inc.) is good for computer novices. HyperCard does not have color capabilities and requires large amounts of computer memory due to the use of graphic files for text and graphics.

2. Hypergate (Eastgate Systems, Inc.) is similar to HyperCard but has no support for animation, digital sound, or video.

3. SuperCard (Silicon Beach Software, Inc.) is a HyperCard clone with enhancements to printing capabilities.

4. Guide (OWL International) is similar to the PC version.

It is possible to view prototypes of HT/HM interactive education and information retrieval by visiting one of the following sites:

1. The National Demonstration Laboratory for Interactive Educational Technologies, Smithsonian Institution, Washington, D.C. Call (202) 357-4749 for an appointment.

2. The U.S. Department of Education Research Library, Washington, D.C. Appointments can be made by calling (202) 357-6699.

3. The Association for Computing Machinery (ACM) Conferences on Hypertext. The Association's number for information is (412) 327-8181.

4. The Society for Applied Learning Technology (SALT) conferences. For information, call (800) 457-6812; in Virginia, (703) 347-0055.

3. HYPERTEXT/HYPERMEDIA ISSUES

With the advent of HT/HM developmental software come new concerns which need to be addressed when designing training. The main issues are cognitive amplification, learner control, navigation, and cognitive overhead.

3.1 COGNITIVE AMPLIFICATION

Marchionini (1988) quotes statements from hypertext proponents: "These systems model human associative memory" and thus "can serve as powerful cognitive amplifiers." Cognitive amplification occurs when humans, studying new concepts or facts, learn the unknown in light of what they already know. Cognitive psychology contends that a person's current knowledge is organized in a web tied together and accessed by associations. For example, if one associates the way a computer disk drive works with the way a 45 RPM record is played, then any new knowledge about computer disk drives will be understood in the context of this association. Therefore, if existing knowledge structures can be mapped for a novice and an expert, the meshing of the two will result in the novice becoming an expert. Artificial intelligence (AI) has sought to do this through the use of expert systems.

HT/HM's contribution to expert systems is the immediate facilitation of associative links (relationships) that fulfill a novice's quest for knowledge. The comprehension of the relationships suggested by these associative links results in the information from an expert model being firmly planted into the novice's mental knowledge structure. Jonassen (1988) contends that increasing meaningful links (or associations between existing knowledge and new knowledge) will increase comprehension and will promote ease of learning.

Jonassen stresses that learning is a reorganization of the learner's knowledge structure. Jonassen cites hypertext as a tool that has the most promise for mapping subject matter knowledge onto the learner's knowledge structure. Shneiderman and Kearsley (1989) point out that increased learning resulting from the use of hypertext is due to HT/HM focusing attention on the relationships between ideas rather than on isolated facts.

3.2 USER CONTROL

The user control issue is put into perspective by Jonassen (1986), who contends that the main purpose of instructional text is to present information relevant to the needs of users. Users should determine what information and what sequence of access will best suit their individual needs. Advertisement of options is needed by all users. Users should know their options, although research literature does not support giving them complete control of instruction (Jonassen, 1986). Stevens (1989) warns that wandering around in random order gathering facts, concepts, and processes could result in "meaningless understanding."

User control decisions must be predicated on the user's knowledge of and familiarity with computers, CAI, HT/HM, and subject matter. An accomplished user should be given the maximum amount of freedom, while a novice should have very little control over movement within a software program. The use of multiple content structural hierarchies for diverse audiences is one way to implement user control. A pretest could determine the user's level of knowledge, and the system would then place the user at a suitable level. The author should include a suggested path for all levels of users, with novices getting a more controlled path initially and slowly being allowed more freedom. The increased sense of control over the interaction process may produce increased involvement and desire to study and read more, according to Shneiderman and Kearsley (1989).

3.3 NAVIGATION

Getting "lost in hyperspace" (Shneiderman and Kearsley, 1989) means not having enough information about one's location in a hyperdocument relative to the overall structure of the HT/HM database. There are no page numbers or chapters that give spatial clues as to the user's location. Conklin (1987) describes this problem as the tendency to lose one's sense of location and direction in a nonlinear document. The freedom of HT/HM can cause a user to become disoriented, especially if the database is large and unfamiliar.

Designers should build in maps indicating all screens visited plus the current location. Users need the choice to continue or return to any of the previous screens. A graphics browser can be used to accomplish this procedure. Systems with less internal support for graphics use a "historical listing," which is a list of the names of the screens visited. Some systems do not have internal support for a graphics browser, but most will allow external coding to accomplish tracking.

3.4 COGNITIVE OVERHEAD

Conklin (1987) defines cognitive overhead as the additional effort and concentration necessary to maintain several tasks or trails at one time. A novice to computer, CAI, HT/HM, and the subject matter, thrust into an unstructured atmosphere, will have many decisions to make. Stevens (1989) suggests that a new user may be overwhelmed by the varied sights, sounds, colors, and linkages available and calls it a "media mess." Overcoming a lack of knowledge about computers, CAI, and HT/HM can be accomplished in an introductory training session. A pretest or menu to bypass the introduction should also be included, as the user should not be forced to view information already mastered. Users should be free to concentrate on the subject matter and not be frustrated by attempts to operate HT/HM software.

4. HYPERTEXT/HYPERMEDIA RESEARCH

Ullmer (1989), citing Hannafin's (1987) concerns regarding interactive video research, stated, "the face validity of interactive video preempted the research needed to empirically validate the instructional effectiveness of the technology." The preemption of research is also happening in HT/HM.

4.1 STATUS

Questions asked by Hannum (1990) of other research efforts can be adapted to HT/HM research as follows:

1. Can instructional materials be delivered via HT/HM?

   Applications listed in this report attest to the feasibility of delivering instructional materials via HT/HM.

2. Will users accept HT/HM as a method of instruction?

   Research with a group of children showed acceptance and sustained interest by even the slowest users (Harris and Cady, 1988). Harris and Cady's (1988) statements provide insight about the effectiveness of HT/HM as a method of instruction as follows:

   The educational validity of hypertext literature lessons has been proven to us. For the classroom teacher, a student's passive learning style, impaired reading skill, and undeveloped thinking ability can provide disheartening obstacles. Hypertext is a tool which helps in overcoming these obstacles.

   Advanced users will like being able to access specific information without having to follow hierarchical structures.

3. Will students learn from the program?

   Professors at Brown University found that users discussed topics in greater depth as a result of following associative links. In a literature course at Brown, Intermedia was used to help students understand connections between material and ideas covered in the course. The results were that more students took part in the discussions and began to cite bibliographical and historical information from Intermedia (Smith, 1988). Shneiderman and Kearsley (1989) reference this same project stating, "there is some evidence from evaluation of the Intermedia system that it can result in a deeper understanding of the material taught." Harris and Cady (1988) found that even slow users improved their knowledge levels after using a HT program.

4. What subjects are suitable for HT/HM presentation?

   Current research has not produced any restrictions on subject matter.

5. What kinds of users profit from HT/HM as a form of instruction?

   Advanced users are the clear winners due to the quick access and freedom to follow associative paths. However, all users, once over the hurdle of being a novice to computers, CAI, and HT/HM, will benefit from software that allows them to follow a generative idea to information that can refute or substantiate their ideas. The generative idea captures the users' interest; the CAI does not force them to other isolated facts but allows them to substantiate facts before proceeding. Users must understand facts before they can understand a concept composed of the facts.

   Problem solving depends upon concepts being understood. Hence, HT/HM fosters the teaching of problem solving and procedures. According to Marchionini and Shneiderman (1988):

   > Results of many evaluative studies demonstrate that even novices find HT/HM easy and effective to use. Although much remains to be learned about how users apply hypertext for information-seeking, clearly these systems offer distinct advantages for finding facts, browsing knowledge, and acquiring wisdom.

The following statements from educational researchers and other members of the HT/HM community give insight into the concerns for HT/HM research:

- "It is important that baseline data be collected and shared in these early stages of hypermedia use in education so that continued development and revision can be empirically guided rather than haphazard and redundant" (Marchionini, 1988).

- "Only more research will verify the effectiveness of hypertext designs" (Jonassen, 1986).

- "A major line of research that should go hand in hand with the development of hypertext and other electronic document systems is formal, controlled experimental studies of users' interactions with these systems followed by actual use studies to confirm results" (Smith, Weiss, Ferguson, 1987).

- "Since hypertext systems have a brief history of application, we have sparse evidence for their effectiveness, let alone proven principles to guide design.

Significant problems with hypertext plague both authors and users" (Marchionini and Shneiderman, 1988).

Jonassen (1988) sums up the current state of research by observing, "Few, if any, verified principles of hypertext design are available to help designers."

4.2 IMPLICATIONS FOR RESEARCH

Empirical data to guide any type of HT/HM product development is scarce, especially in the area of microcomputer-based HT/HM CAI developmental software packages. The majority of the data provides favorable results, yet findings indicate the need for more research. The main areas of needed research are in the design of HT/HM software, design principles for cognitive instructional systems design (ISD), and the application of HT/HM to new hardware technologies.

4.2.1 Design of Hypertext/Hypermedia Software

To facilitate authoring of CAI and other HT/HM, developers must have appropriate authoring tools. Computer engineers seemed to have had more to do with design of current software than did instructional system designers. Figure 2.3 contains Conklin's (1987) list of capabilities he thinks should be researched and designed into HT/HM software. Figure 2.4 lists the additional capabilities Kearsley (1988) thinks should be researched and considered when designing HT/HM software.

First generation HT/HM software will probably be accepted more readily by developers who have programming skills, as all current versions require internal or external programming. Internal programming is needed for any internal links that are not automatically generated while external programming is needed for software that does not have internal support for links to video, CD-ROM, CD-I, DV-I, smart cards, stereo, graphics, digital sound, or animation.

The widespread acceptance of HT/HM will come when enhancements to current versions can support additional user selected options. For example, the Apple Macintosh provides a computer environment in which options are selected from on-screen pull-down menus. The user points to the desired option on the menu using a mouse. The software automatically performs the task or option the user selects, without the user being aware of what happens internally.

4.2.2 Design Principles for Cognitive Instructional System Design (ISD)

Jonassen (1988) states that hypertext design is theory rich and research poor and offers the following ISD research agendas:

1. How can hypertext be structured for maximum learning?

1. Tracing references; references must be easily followed forward to their referent or backward to their reference.
2. Creation of new references; users can create personal networks or annotate another user's document without changing the referenced document.
3. Information structuring to allow both hierarchical and nonhierarchical organizations to be imposed on unstructured information; even multiple hierarchies could organize the same materials.
4. Global Views (browsers that provide table-of-contents-style views of a hyperdocument's nodes), which would support easier restructuring of large or complex documents, should be available. Global and local (nodes close to the currently displayed node) views can be mixed effectively.
5. Customizing to permit text segments to be threaded together in many ways, allowing the same document to serve multiple functions.
6. Task stacking, which allows the user to have several paths of inquiry activated and displayed on the screen at the same time. Task stacking should include the capability to unwind any given path to the original starting point. An alternative could be the capability to hold a path history in memory to orient the user and provide the option to return to any point on the path history as needed.
7. Collaboration capabilities, which allow several authors to collaborate on the development of the same document, are needed.

Fig. 2.3. Capabilities to design into HT/HM (Conklin, 1987).

1. Capability for creating links between nodes.
2. Capability to switch between instructional system designer mode and user mode to test ideas.
3. Easily recognized and implemented or automatic saving of the item of information being developed.
4. A range of editing functions available such as copying, moving, insertion, and deletion.
5. Availability of lists of link names and index terms.
6. Screen formatting commands.
7. Capability to import existing text or graphic files.
8. Availability of search/replace functions for making changes.
9. Control of color (text or background).
10. Capability to export files to other systems.
11. Capability to initiate other programs from within any software product produced by an authoring system.
12. Support for CD-ROM, DV-I, videodisc, and optical storage media.

Fig. 2.4. Capabilities to design into HT/HM (Kearsley, 1988).

Method 1. Observe how users navigate through relatively unstructured hypertext and develop a path analysis (a mapping study) to classify the prominent paths taken. Relate those paths to individual differences in learning style.

Method 2. Use various models of structured design such as hierarchical and nonhierarchical to expose material to users. Assess the effects of the material on retention.

Method 3. Assess the user's knowledge structures, then assess the differences when users explore structured versus unstructured hypertext.

2. How can knowledge structures be assessed?

   Use semantic networking software, such as Learning Tool or SemNet, which is capable of mapping the cognitive structures of users. From the mapping studies, compare the cognitive structure maps produced by users who have relevant individual differences. Jonassen (1988) states, "Since these tools can readily illustrate the reorganization of knowledge structures, they can be used to assess the effects of hypertext structures on users' knowledge structures."

3. What methods should be used to structure hypertext?

   Method 1. Use inductive design methods utilizing path patterns set by a user during the processing of unstructured hypertext documents. These path patterns are then designed into the hypertext document's structure. Researchers can pre-assess user styles and either assign users to hypertext documents that have structures consonant with or dissonant to their preferred learning style or allow the user to choose. "Preferential matching of users to instructional treatments based upon their individual knowledge structures has been the theoretical goal of designers of intelligent systems. Hypertext provides that possibility" (Jonassen, 1988).

   Method 2. Use deductive design methods that map the expert's knowledge structure onto the user's knowledge structure. Research is needed on how to define the ideal knowledge structure, such as whether to use quantitative methods that entail the development of a cognitive or semantic map of the expert's knowledge through the completion of word associations of all the related concepts in the content domain or to use semantic networking software.

4. How can the integration of hypertext information into the user's cognitive structure be facilitated?

Method 1. Integrate semantic networking software such as Learning Tool or SemNet with hypertext. Use the semantic networking software to see the changes in the user's comprehension and cognitive structures after exposure to hyperdocuments.

Method 2. Use an expert system at the front end to help users access relevant portions or sequences of hypertext. The same system can query the user and, based on the difference between the user's knowledge and that of the expert, use a series of "if-then" rules to access relevant portions of the hyperdocument.

Kearsley (1988) gives instructional systems designers some encouragement and a warning if they are developing or researching HT/HM:

> Anyone who has created interactive instruction will already be familiar with many of the issues involved in authoring hypertext databases. For such individuals, learning to author hypertext means getting used to branching anywhere on the screen under the initiative of the user. It will require you to think hard about the structure and organization of your information, since the reader can now enter and leave documents at arbitrary points.

Morariu (1988) gives the instructional designer a "knowledge transfer" from past ISD experience and a deeper understanding of the research implications involved in designing HT/HM with a model for designing instructional hypermedia. Morariu lists the following components to be identified and specified:

1. User characteristics which include previous knowledge, learning styles, and motivation.

2. Goals and objectives stated in behavioral terms, a full breakdown of the content and measurable outcomes for the entire instructional environment.

3. Pedagogical model (the method used to teach the content), such as a tutorial, simulation, drill and practice, or serendipitous exploration.

4. Navigation, meaning the user interface design that defines how the user can move through the system. For example, how do users know where they are? Can paths be retraced easily? Are graphic icons used for selecting information or do users need to type key words?

5. Structure, or the overall organization of the information, such as hierarchical with topics and subtopics.

6. Format, meaning the media for presenting the content, such as text, graphics, animation, audio, still images, and motion video.

7. Content, the actual information and topics to be conveyed to and explored by the user.

4.2.3 Applying New Hardware Technology

To understand how new technology can enhance HT/HM applications, one needs to know HT/HM's technical requirements. Conklin (1987) points out that, with advances in technology, many types of information including text, graphics, digitized speech, audio recordings, pictures, animation, film clips, and other sensory media can be linked together. The vast majority of current desktop computers cannot support hypermedia. Supporting hypermedia and making it economically feasible would require the following: high memory capacity microcomputers, high capacity storage media, and high resolution monitors, all at low cost.

Most of HT/HM research has been conducted on mainframe computers. As the trend of lower cost, more powerful microcomputers continues, more HT/HM developmental software will be developed.

Ofiesh (1989) provides capacity references in terms of computer disc, pages of data, and minutes of video:

1. CD-ROM (Compact Disc-Read Only Memory). A 600 megabyte CD-ROM can hold 250,000 pages of data or 1500 traditional floppy discs of data.

2. DV-I (Digital Video-Interactive). A DV-I disc can provide up to 72 minutes of full motion, color video.

3. Smart Cards. A smart card is an optical card, a device which uses light (usually from a laser) to code information onto a storage medium (Ullmer, 1989). At present, an optical card the size of a credit card can hold 600 pages of data. A card with a 600,000-page capacity will be available soon.

With these capacities in mind, Shneiderman and Kearsley's (1989) statement that hypermedia "must have high-capacity, fast access, digital storage methods" is easily understood. Ofiesh (1989) predicts that within a few years computers will include the types of disc drives described above.

Optical storage holds great potential for HT/HM due to high storage capacity. Large volumes of data will require a database structure that facilitates direct and immediate access, a capability that HT/HM has almost perfected. With this massive amount of storage available, every reference needed by a student could be included in the CAI. For example, a CAI on the Naval Aviation Logistic Data Analysis (NALDA) system could contain not only lessons on how to structure query statements but also lessons on the Navy Aviation Maintenance Plan (NAMP) and Aviation Maintenance Material Management (AV-3M). The complete CAI, the 13 NALDA instruction manuals, and the NAMP could be entered on just one of the new 600,000 page optical cards.

The technology for high resolution monitors already exists, but the problem is high cost. It is hoped that the demand for high resolution monitors to accompany lower priced computers will lead to a reduction in the cost of the monitors. The cost factor limits research and application, because HT/HM requires high resolution monitors to support graphics, still pictures, video, and motion pictures.

5. IMPLICATIONS FOR TRAINING

Jonassen (1988) concludes that HT/HM's flexibility in structure and style make it the most effective technology system to date for individualizing instruction. What are the training implications for HT/HM in commercial, academic, and military training?

5.1 COMMERCIAL TRAINING

Stevens (1989) provides insight into implications for the use of HT/HM in training by describing CAI which surrounds users with the sights, sounds, and words they will use on the job. Hypermedia can simulate the real world as well as, and in some cases, better than, well-designed interactive video-based systems. Stevens identifies the problem with current CAI as the inability to represent information adequately in the variety of forms needed to match the myriad of ways people learn. Modifications are far simpler in hypermedia systems due to their modular construction, as each piece of graphic, text, picture, or sound is treated as an individual object or module.

Federal Express has begun installation of the first 600 of 1300 interactive stations to provide training to its 6000 customer service and 17,000 courier personnel (Ofiesh, 1989). The projected cost of this project over five years is $40 million. Expected training cost savings for the same five years is $100 million in travel and employee time (Ofiesh, 1989). IBM also points out a $200 million annual savings for its technical and managerial staff training, using similar technology (Ofiesh, 1989). These systems do not use hypermedia; however, the application is the same. Similar savings for interactive courseware can be expected using HT/HM.

Stevens (1989) reports that only four percent of all business training is delivered on or mediated by a computer and most of this training provides information on how to use computers. The limited use of computers for business training may be due to dissatisfaction with the use of existing CAI training packages. Stevens characterizes hypermedia as a tool that will allow "competent" design to be executed with few limitations. Ofiesh (1989) points to the need for commercial application of computer technology systems for training by stating, "The microcomputer and its peripherals are now the great facilitators of creativity. Only by harnessing this creativity can we remain the great inventors and remain competitive in the years ahead."

5.2 ACADEMIC TRAINING

Many hypertext software packages allow the importation of files, slides, video, and still shots from video, thus a developer can bring current information into HT/HM. This process is called "repurposing." Teachers in the fields of art, literature, biology, and medicine are finding hypermedia to be an effective way of automating existing slides and lectures and providing students with remedial study tools. Cornell Medical College has loaded its entire second year curriculum into a hypertext system.

Gary Marchionini (1988), a professor at the University of Maryland, sees three main characteristics of hypermedia that have great potential for academia as compact storage, user control, and interaction potential. Marchionini states, "Hypermedia systems allow huge collections of information, in a variety of media, to be stored in extremely compact form and accessed easily and rapidly." The compact storage allows an increased amount of subject matter, suggested paths by the author, and support materials, such as encyclopedias and dictionaries to be readily accessible.

Marchionini continues, "Hypermedia is an enabling rather than a directive environment, offering unusually high levels of user control." He believes the benefit of this fluid environment is that the student is forced to apply higher order thinking skills while noting that more research is needed to mitigate problems inherent to this environment. These problems include disorientation, distraction, the impact of new technology, and human psychological and sociological implications.

Interaction potential is increased through the use of hypermedia. Hypermedia's flexibility will enable users to create unique tours that can be saved for later study or shared with other users and the instructor (Marchionini, 1988). Hence, there is potential for improved learning, fostered by increased interaction between users and instructors.

5.2.1 Academic Applications of Hypertext/Hypermedia

The following are current academic applications and goals of HT/HM:

1. The University of Southern California's Project Jefferson's goal is to provide a computerized program which integrates the assignments of the freshmen writing program classes, pedagogical goals of the instructor, teaching of library and research skills, and the ability to access on-line information with a minimum of training (Kinnell, 1988).

2. Harvard University's Project Perseus has the study of the classics utilizing hypermedia as its goal (Kinnell, 1988).

3. Brown University's IRIS Project's goal is to help students understand connections between materials and ideas covered in courses (Smith, 1988).

4. The CSILE Project's goal, at the University of Toronto, is to foster higher order learning strategies in acquiring, organizing, evaluating, and communicating knowledge (Shneiderman and Kearsley, 1989).

5. New York's Bank Street College's Palenque Project's goal is the exploration of hypermedia as an instruction method for young children (Shneiderman and Kearsley, 1989).

6. The University of North Carolina's Writing Environment's (WE) goal is to provide a hypertext environment that will help writers transform loose

associative networks of ideas into a hierarchical structure and then write a document in accordance with that structure (Smith, Weiss, Ferguson, 1987).

5.2.2 Medical Applications of Hypertext/Hypermedia

The following medical school applications exemplify the wide range of HT/HM's flexibility:

1. The Electronic Cadaver at Stanford University Medical School teaches computerized anatomy lessons and digitized dissection (Jerome, 1989).

2. The Dynamic Medical Handbook Project, Washington University School of Medicine, St. Louis, Missouri, is a diagnostic tool (Shneiderman and Kearsley, 1989).

3. Cornell Medical College has loaded its entire second year curriculum into a hypertext system (Saffo, 1987).

4. Harvard Medical School's Explorer-1 is used for diagnosis, diagnostic workup, and medical pathophysiology (Shneiderman and Kearsley, 1989).

5.3 MILITARY TRAINING

The military has traditionally been the first to take advantage of developments in educational technology (Ofiesh, 1989). However, Talbert (1988), citing Linn (1988), reports that only recently has the development and utilization of hypertext-based CAI materials for military training begun to be seriously considered. Stone et al. (1982) were among the first to realize the applicability of HT/HM to military training and to identify some of the military's training problems that resulted from rapid advances in technology, such as:

- The volume of the printed documentation, which has accompanied the introduction of complex systems has grown so large that much of it can not be accessed quickly enough to make it useful for field personnel or students.

- The inability of technicians to troubleshoot and repair complex systems rapidly and accurately, which points to a lack of proper training.

- The low reading ability of many younger military personnel.

Additional training problems for the military include large numbers of people to train, higher costs in the face of reduced budgets, and skilled military members leaving for civilian jobs. HT/HM can help military training. HT/HM aided embedded training and artificial intelligence (AI) can help solve one of the military's main problems: the loss of "resident expertise" when members leave the service. Expert systems, using hypertext, which contain the components of the expert model and the student model can be developed. HT/HM is capable of direct and immediate access in AI databases, a capability that can make expert systems even more effective.

Talbert (1988), citing MacNiven (1987), points out that the military has not standardized the computer based training applications they have produced to date; hence, applicability is limited due to the many different types of computers/software being used. The goal of the Computer-Aided Acquisition and Logistic Support (CALS) program has been standardization and reduction of paper-based products. CALS will impact all branches of the military. The CALS program's goal to reduce the amount of paper products used by the military will require large databases. HT/HM can provide the direct and immediate access that must accompany large volumes of digitized information. The efficiency of the CALS program could depend on HT/HM.

Another area that can potentially benefit the military is the conversion of existing databases to hypertext. Large databases which depend on current file structures cannot be directly converted into HT/HM with current technology. A developer would need to make the associative links among the information entries. New databases in hypertext formats are already appearing in the military for equipment manuals and many software companies are furnishing product information in this format.

The savings over current training costs will be a primary factor in the adoption of HT/HM. The cost for travel, facilities, teachers, and curriculum can never be eliminated; however, current budget cuts have caused the military to look for ways to save money and still provide for national defense. Investments in large storage capacity media and HT/HM's cognitive instructional systems designs are capable of providing long-term cost savings. This is a benefit the military cannot ignore. Talbert (1988) cites a conclusion reached by Joseph Psotka (1987) of the Army Research Institute: "The combination of natural language processing technologies, AI database relationships, and hypertext interface capabilities demonstrates a very powerful method of instructional presentation in the military training environment."

6. SUMMARY

HT/HM will increasingly impact CAI, on-line manuals, and other training applications as software packages are enhanced. Empirical data to guide developers on the critical issues of cognitive instructional systems design will add stability to the development of HT/HM. Many training environments are already receiving benefits from HT/HM. On-line maintenance manuals, diagnostic tools, and software documentation are examples of information seeking applications which demonstrate built-in benefit of HT/HM database structures such as providing direct and immediate display of information.

New computer technology, which increases memory, storage capacity, and processing speed, will require a database structure capable of operating efficiently in this new environment. HT/HM has provided promising results in this area. The freedom to access information directly and immediately (regardless of the volume of the data), within a cognitive instructional systems design, makes HT/HM a noteworthy technology for the development of training packages.

7. REFERENCES

Bush, V., "As We May Think," *Atlantic Monthly*, July 1945.

Conklin, G., "Hypertext: An Introduction and Survey," "*IEEE Computer*, September 1987.

Delisle, N. M. and Schwartz, M. D., "Contexts: A Partitioning Concept for Hypertext," *ACM Transactions on Office Information Systems*, April 1987.

Devlin, J. and Berk, E., "Jump from Topic to Topic with Hypertext," *PC Week*, March 20, 1989.

Fiderio, J., "A Grand Vision," *BYTE*, October 1988.

Hannafin, M. J., "Empirical Issues in the Study of Computer Assisted Interactive Video," *Educational Communication and Technology Journal*, 33, 235-247, 1987.

Hannum, W., *Review of Research in Computer Assisted Instruction*, Prepared for Martin Marietta Energy Systems, Inc., Under Contract for the U.S. Navy, by Oak Ridge Associated Universities, Training and Management Systems Division, Oak Ridge, TN, 1990.

Harris, M. and Cady, M., "The Dynamic Process of Creating Hypertext Literature," *Educational Technology, Special Issue: Hypermedia*, November 1988.

Jensen, A., Jordan, D., and Russell, D., *The IDE System for Creating Instruction*, submitted to Applications of Artificial Intelligence and CD-ROM in Education and Training Conference, Arlington, VA, October 1987.

Jerome, M., "The Electric Cadaver," *PC/COMPUTING*, February 1989.

Jonassen, D. H., "Designing Structured Hypertext and Structuring Access to Hypertext," *Educational Technology, Special Issue: Hypermedia*, November 1988.

Jonassen, D. H., "Hypertext Principles for Text and Courseware Design," *Educational Psychologist*, 21(4), 269-292, 1986.

Kearsley, G., "Authoring Considerations for Hypertext," *Educational Technology, Special Issue: Hypermedia*, November 1988.

Kinnell, S., "Information Retrieval in the Humanities Using Hypertext," *On-Line*, March 1988.

Kuhn, A., *The DoD Gateway Information System (DGIS): The Development Towards Artificial Intelligence and Hypermedia in Common Command Language*, Defense Applied Information Technology Center Report Number DAITC/TR-88/009, DTIC/TR-89/4, DAITC, Alexandria, VA, 1988.

Linn, M. C., *Hypermedia and Programming Instruction: Opportunities for Meeting DOD Training Needs*, Position Paper, Instructional Technology Program, University of California, Berkeley, CA, 1988.

MacNiven, Major D. B., *Computer-Based Training Systems: Organizing to Use Them*, Student Report 87-1615, Air Command and Staff College/EDCC, Maxwell AFB, AL, 1987.

Marchionini, G., "Hypermedia and Learning: Freedom and Chaos," *Educational Technology, Special Issue: Hypermedia*, November 1988.

Marchionini, G. and Shneiderman, B., "Finding Facts vs. Browsing Knowledge in Hypertext Systems," *IEEE Computer*, January 1988.

Martin, J., "News Briefs," *PC Week*, July 24, 1989.

Morariu, J., "Hypermedia in Instruction and Training: The Power and the Promise," *Educational Technology, Special Issue: Hypermedia*, November 1988.

Ofiesh, G., "Emerging Technologies of Information and Communication and Their Potential Impact on Education, Training and Human Productivity," *Business Economics*, February 1989.

Pallatto, J., "Manuals Take Hypertext Route," *PC Week*, July 24, 1989. Picarille, Lisa. "Spinnaker Program Gives Hypercard Power to PCs," *PC Week*, May 29, 1989.

Psotka, J., *AI Applications to Machine Translation and Language Instruction*, (Published in a Confidential Report of Knowledge-Based Information Systems, Title unavailable,) The Hague: SHAPE Technical Center, 1987. Copy Furnished to Talbert (1988) by Author.

Saffo, P., "What You Need to Know About Hypertext," *Personal Computing*, December 1987.

Shneiderman, B. and Kearsley, G., *Hypertext Hands-On*, Addison Wesley Publishing Company, Reading, Massachusetts, 1989.

Smith T, John B., Weiss, S. F., and Ferguson, G. J., *A Hypertext Writing Environment and Its Cognitive Basis*, Defense Applied Information Technology Center Report Number DAITC/TR87-033; MDA903-86-C-0345, DAITC, Alexandria, VA, 1987.

Smith, K. E., "Hypertext-Linking to the Future," *On-Line*, March 1988. Stevens, G. "Applying Hypermedia for Performance Improvement," *Performance & Instruction*, July 1989.

Stone, D. E., et al., *A Hypertext Job Aid for Maintenance*, Defense Applied Information Technology Center Report Note 83-5; AD-A133103, DAITC, Alexandria, VA, 1982.

Talbert, M. L., *An Object-Oriented Approach to the Development of Computer-Assisted Instructional Material Using Hypertext*, Defense Applied Information Technology Center Report Number AD-A202-656; AFIT/MA/GCS/88D-1, DAITC, Alexandria, VA, 1988.

Ullmer, E. J., "Videodisc Technology," *Lister Hill Monograph*, LNCBC 89-1, January 1989.

Young, J., "Hypermedia," *MAC World*, 1988.

Section 3

ARTIFICAL INTELLIGENCE: IMPLICATIONS FOR TRAINING

TABLE OF CONTENTS

LIST OF FIGURES

Figure | Page

ARTIFICIAL INTELLIGENCE: IMPLICATIONS FOR TRAINING

1. BACKGROUND

The concept of emulating human intelligence with a computerized system has challenged, confused, threatened, and disappointed humans for many years. Psychologists have long been interested in developing computer programs that simulate human problem solving (Harmon and King, 1985). Part of the attraction to emulating human intelligence is the belief that intelligent machines can work more quickly, efficiently, accurately, and consistently than people.

1.1 DEFINITIONS

Attempts to define artificial intelligence (AI) are often incomplete or unclear. The term "artificial intelligence" is often criticized or challenged because of the debate over the possibility of the existence of a truly intelligent machine.

John Minsky (1968) defined AI as "the science of making machines do things that would require intelligence if done by men." Rich (1983) defined AI as "the study of how to make computers do things at which, at the moment, people are better." Whitby (1988) challenge these definitions because he felt AI as a science had been challenged and Rich's definition was no longer true.

Other current definitions of artificial intelligence are:

- "Artificial intelligence (AI) is an approach to understanding behavior based on the assumption that intelligence can be analyzed by trying to reproduce it. In practice, reproduction means simulation by computer. AI is, therefore, part of computer science" (Garnham, 1988).

- "Since World War II, computer scientists have tried to develop techniques that would allow computers to act more like humans. The entire research effort, including decision-making systems, robotic devices, and various approaches to computer speech, is usually called artificial intelligence (AI)" (Harmon and King, 1985).

- "... A subfield of computer science concerned with the concepts and methods of symbolic inference by a computer and the symbolic representation of the knowledge to be used in making inferences.... A field aimed at pursuing the possibility that a computer can be made to behave in ways that humans recognize as 'intelligent' behavior in each other" (Feigenbaum and McCorduck, 1983).

- "Artificial intelligence is the science of getting machines to emulate human intelligence" (Jonassen, 1988).

1.2 A BRIEF HISTORY OF ARTIFICIAL INTELLIGENCE

McCarthy first used the term "artificial intelligence" to characterize the simulation of intelligent behavior by machines in a proposal for a conference held in 1956 at Dartmouth College, New Hampshire, (Garnham, 1988).

Early attempts to develop an intelligent system relied on Warren McCulloch's research on neural nets as a building block for work in simulating the human brain (Forsyth and Naylor, 1985). Neural nets are defined as "the models of the logical properties of interconnected sets of nerve cells" (Garnham, 1988). Attempts to develop an intelligent system comparable to the human brain, which contains 10 billion neurons, failed due to the limitations of computer hardware and software. Most programs tried to simulate the human behavior needed for recognizing objects and understanding simple text.

The works of Newell, Shaw, and Simon, who were among the early pioneers of AI, had a dominant influence on AI during the 1960s. Newell, Shaw, and Simon claimed reasoning was a result of using heuristics and not the use of logical rules (Garnham, 1988). Heuristics are defined as "a rule of thumb or other device or simplification that reduces or limits search in large problem spaces, and, unlike algorithms, heuristics do not guarantee correct solutions" (Harmon and King, 1985). The attempt to build artificial brains advanced to developing several heuristically guided search strategies, such as algorithms and list processing (Forsyth and Naylor, 1985). Work also began on sets of programs dedicated to a very narrow area of expertise, the expert system.

AI literature from the 1970s is sparse although work on the expert systems continued. Early expert systems focused on a single type of problem. The programs were costly to develop, were slow, did not produce practical results, and were too complex for the computers in existence at that time. During the late 1970s, practical expert systems were developed, which, combined with the development of faster, more powerful, and less expensive computers, provided a catalyst for the existing renaissance in AI.

At present, there are trained AI researchers employed by the federal government. The Department of Defense, particularly the Strategic Defense Initiative and the Strategic Computing Program, along with nearly every major government institution, has invested in the use of AI technology for problem solving, decision making, maintenance of complex systems, and hazardous operations (Schoen and Sykes, 1987).

1.2.1 The Influence of Behavioral Psychology

Historically, behavioral psychology, which emphasizes discrete stimuli and responses as well as objective reward and punishment schedules, has dominated the training environment in large corporations since the 1960s. Behavioral techniques have been effective in developing task analyses, which pinpoint a sequence of actions for describing job requirements. As a result of the emphasis of behavioral psychology in training, the actual performance of a task received attention from trainers in the manufacturing environment.

Throughout the 1960s and early 1970s, behavior-based courses were developed to teach jobs associated with manufacturing, sales, management, and computer programming analysis (Harmon and King, 1985). However, when training for more complex tasks or performance is needed, the conventional behavioral techniques are not adequate for determining what a subject expert knows or for communicating the expert's knowledge to new performers. The techniques of cognitive psychology have been determined to be much better for managing these issues (Harmon and King, 1985).

1.2.2 The Influence of Cognitive Psychology

Cognitive science is "the study of cognition, or the thought processes, structures, and mechanisms used by human beings" (Chabris, 1989). The term cognitive science was coined in the 1970s in an attempt to initiate interdisciplinary research between the fields of cognitive psychology and AI (Garnham, 1988).

Building an expert system or knowledge system requires using the techniques developed by cognitive psychologists (Harmon and King, 1985). Professor Edward Feigenbaum of Stanford University, who is one of the leading researchers in expert systems, calls those who build systems "knowledge engineers." The task of the knowledge engineer is to capture the knowledge of the human expert so it can be encoded into the rules for the knowledge base (Johnson, *Intelligent...*, 1988). Although many of the concepts of cognitive science have been ill-defined and vague for applications in corporate training circumstances, knowledge engineers are beginning to succeed in developing practical techniques based on cognitive science (Harmon and King, 1985).

Mental functioning in information-processing terms is a goal of both cognitive psychology and AI, with research focusing on the development of methods for representing and manipulating information (Garnham, 1988). Glaser and Bassok (1989) state that the analysis of complex human performance is the most important contribution of cognitive science to instruction.

Cognitive learning theory has contributed greatly to the advancement of instructional design in connecting the interaction between stimulus information (information to be learned) and knowledge (information already stored in the memory) (Garnham, 1988; Tennyson and Park, 1987).

> Cognitive theory implies that acquisition of information begins in the working memory, where the learner decodes appropriate necessary knowledge from long-term memory to help in understanding and encoding new information (Tennyson and Park, 1987).

Tennyson and Park also suggest, in compliance with cognitive learning theory, that a learning theory model should address the questions of both *how* and *why* learning occurs.

2. THE USE OF AUTOMATED SYSTEMS IN TRAINING

Currently, the most established form of AI is the expert system, also called a knowledge system. The expert system alone does not provide an expert training system and requires external guidance (i.e., a printed self-paced guide or an instructor) for successful training (Schmidt and Lazar, 1989). Schmidt and Lazar point out that instructional designers, who should determine training objectives and instructional strategy, must work closely with subject matter experts to develop the expert system into a viable training tool; intelligent computer aided instruction (ICAI), or intelligent tutoring system (ITS).

2.1 EXPERT SYSTEMS

Expert systems have existed for approximately 20 years and are intended to do the work of human experts (Garnham, 1988). The main purpose of the expert system is solving problems from the subject domain. Expert systems emphasize the need to teach more than just facts and to explore meta-cognition or how the brain processes information and inferences. Hannum (Personal conversation, 1989) has summarized the impact of expert systems on training: Expert systems reduce the need for traditional training and the development of expert systems demands that the training content be more clearly defined.

The terminology used to refer to expert systems varies from knowledge systems to knowledge-based systems to intelligent knowledge-based systems. The term "expert system" at times may have been overgeneralized to market a product. Definitive descriptions of the knowledge/expert system are:

- "A knowledge system is a collection of AI techniques that enables computers to assist people in analyzing problems and making decisions" (Harmon and King, 1985).
- "An expert system is a program that embodies (some of) the knowledge of a human expert in a domain in which expertise comes with experience" (Garnham, 1988).
- "Knowledge-based systems capture the knowledge and experience of individuals; store their decision-making process used to solve these complex, labor-intensive problems; and then make that expertise available to the application's users" (Florian and Barros, 1989).

The advantages of the expert system in helping people learn how to solve problems are subtlety, flexibility, efficiency, effectiveness, empowerment, and compatibility with the training professional (Siegel, 1989). Because of the influence of expert systems, future training courses will depend on the detailed analysis of heuristics and will teach personnel to apply specified heuristics to training problems.

Presently, small knowledge systems are available that have the capability of solving small, specific problems. As personnel perform tasks that previously required supervision or prior training, these knowledge systems will replace some conventional training and on-the-job training (Harmon and King, 1985). The availability and ease of installation of these systems will be a significant innovation in the analysis and design of corporate training programs.

The common acceptance and use of expert systems is dependent on people outside the AI community learning to use knowledge engineering techniques, the availability of "economical" machines, and the development and the availability of user-friendly knowledge engineering system building tools.

Developers of expert systems began to understand that more than an expert model was needed for improving the explanation of the system's expert decisions. Researchers from instructional disciplines became involved in attempting to develop additional system characteristics to help them understand the user and to know when and how to offer explanations (Johnson, *Intelligent...*, 1988).

2.2 COMPUTER AIDED INSTRUCTION (CAI)

Computer aided instruction (CAI) was primarily developed by educational researchers and training developers in an effort to solve practical problems through the application of computer technology. Early, traditional CAI systems used a theory of instruction based on Skinner's (1968) operant psychology which provided the basis for the linear programmed instruction that was used during the 1950s and 1960s (Park, Perez, and Seidel, (1987)). Using this theoretical framework, the CAI author must anticipate every possible response of the user, predetermine what misconceptions caused an incorrect response, and predetermine the programming for providing remedial material (Jonassen 1988; Wenger, 1987). Principles of learning and instruction of CAI that were influenced by Skinnerian behaviorism were later improved by allowing CAI authors to incorporate various instructional principles in the design and development process.

CAI received criticism in the past because of what Carbonell (1970) calls its ad hoc, frame-oriented, behavioral methods (Jonassen, 1988). CAI systems usually store all instructional components in a single structure, always initiate the instructional process, and offer the student little instructional initiative. Task analysis is used for identifying subtasks and content areas and assorted formats (i.e., tutorial, drill and practice, games, and simulations) are also incorporated.

2.3 INTELLIGENT COMPUTER AIDED INSTRUCTION (ICAI)

"Computer programs that use artificial intelligence (AI) techniques to help a person learn are called intelligent computer aided instruction (ICAI) or intelligent tutoring systems (ITS)" (Kearsley, 1987). Throughout the remainder of this document, the terms ICAI and ITS will be used as ICAI/ITS to indicate their acceptance as synonyms for each other.

Since Carbonell pioneered the application of AI to CAI in 1970, many ICAI/ITS systems have been developed (Garnham, 1988). Although ICAI/ITS systems may take many forms, the components of the system use AI principles and techniques to allow flexibility for the student and the program (Tennyson and Park, 1987). ICAI/ITS is not just an extension of traditional CAI, it is a radical shift because of the difference in methodology (Wenger, 1987). Where CAI uses task analysis for identifying subtasks and content areas, ICAI/ITS uses AI knowledge representations techniques or cognitive analysis to organize domain knowledge into a data structure and procedures (inference mechanisms).

Development of ICAI/ITS was initiated primarily by computer technologists in an effort to explore AI technology. Researchers often examined AI technologies and then choose the subject domain for its adaptability to ICAI/ITS technology. ICAI/ITS principles of learning resulted from the effort to understand the cognitive models of thought processes and rely on cognitive psychology for direction and philosophy. The rapid advancement and convergence of the fields of cognitive science and AI, coupled with the dramatic increase in the power of computers, have driven the technology of ICAI/ITS (Psotka, Massey, and Mutter, 1988).

Dede and Swigger (1988) state that ICAI/ITS systems "understand and purposefully capture (1) the mental dynamics that occur within the student and (2) the progression of instructional process and tasks." ICAI/ITS systems use primarily tutorial inquiry and games and process the stored knowledge, use qualitative evaluation of student responses, diagnose the student's learning needs, and prescribe instructional treatment for each student.

Success of ICAI/ITS has been determined by the system's capacity to perform, technically, according to design. Burns and Capps (1988) state that there are three tests of intelligence an instructional system must pass before it can be considered an ICAI or ITS:

1. The knowledge base or domain must be known by the computer system so that inferences or problem solving occurs.

2. The system must diagnose the learner's approximation of knowledge.

3. The system must have a tutorial strategy that reduces performance differences between students and experts.

Pliske and Psotka state that dramatic improvements and sophistication in the construction of CAI are occurring because of the development of ICAI/ITS (Psotka, Massey, and Mutter, 1988). Most ICAI/ITS systems have been developed and implemented using specific hardware designed for AI research.

2.3.1 Components of ICAI/ITS

Present expert systems are only capable of developing new rules that are similar to rules that already exist (Garnham, 1988). ICAI/ITS has been limited to well-structured subject areas such as electronic repair. Development of sophisticated machine learning capabilities would greatly improve expert systems and ICAI/ITS.

The ultimate goal of an ICAI/ITS is to develop powerful models of each of the three following components cited by Tennyson and Park (1987): expert, student diagnosis, and tutoring modules.

2.3.1.1 Expert module

The expert module, also referred to as the knowledge base, expert knowledge, or model, is the part of the system that provides an abundance of specific and detailed domain knowledge derived from the experts in the field (Burns and Capps, 1988). A task of the ICAI/ITS developer is to encode information into the system's data structure and procedures. Encoding domain information for the expert module is a labor-intensive process, which demands 50% of the effort of building an application.

Anderson (1988) states that the vast amount of knowledge in complex domains ensures that authoring systems will never do the work of discovering and codifying the required knowledge of the domain. Anderson also states that, although there are several methods for encoding information (black box models with issue-based tutoring, cognitive-rule models with model tracing, and declarative systems with repeated questioning or Socratic tutoring), the cognitive modeling approach is the easiest approach for developing powerful tutoring methods.

The cognitive modeling approach requires the tutoring of three types of knowledge:

- Procedural - Conveys knowledge about how to perform a task.
- Declarative - Conveys general knowledge in an organized format so one can use it for reasoning.
- Qualitative - Conveys knowledge that allows reasoning about causal structure and dynamic processes.

The expert module includes the content (declarative knowledge) and the application of that knowledge (procedural knowledge) to solve problems (Tennyson and Park, 1987).

2.3.1.2 Student diagnosis module

Early ICAI/ITS systems focused primarily on the expert module or representation of the domain knowledge with little emphasis on the student's learning behavior or the tutorial strategies for presenting information (Tennyson and Park, 1987).

VanLehn (1988) states that many ICAI systems now adapt their instruction to the student's needs by inferring a model of the student's understanding of the subject matter. He explains that the student module, which is the component of an ICAI/ITS that represents a student's present state of knowledge, and the reasoning process to develop the student model (called diagnosis), must be designed together. The student model must be able to range from representing the novice to the expert.

Tennyson and Park (1987) define the student-model module as the method of representing the student's learning progress in the subject matter to establish hypotheses about the student's errors and misconceptions. The model of the student's state of knowledge is established by comparing the student's performance with the computer expert's behavior.

2.3.1.3 Tutoring module

The tutoring module, also called the curriculum and instruction module by Halff (1988), is a set of instructional specifications that determines how the material is presented to the student. The tutoring module interacts with the student in determining problems to be solved, evaluating performance, providing assistance, and selecting material for remediation (Tennyson and Park, 1987).

Teaching approaches or methods are selected based on the diagnostic information provided by the student modeling process (Tennyson and Park, 1987). Garnham (1988) identifies the teaching approaches that have been explored in ICAI development:

- The most straightforward approach is to pose problems, evaluate the suggested solutions, and provide information about errors so that the student is able to learn from the errors.

- An alternative to the above approach is to pose a simpler problem so the student will discover any misconceptions about the information. This is a more difficult task, because the system has to have a sophisticated model of the student.

- Repeatedly asking the student questions in an attempt to cause the student to think in terms of debugging his or her knowledge is called the Socratic method.

- Allowing the student to learn by practice is called the coaching method.

Halff (1988) states that tutors must exercise some control over curriculum, be able to respond to students' questions about the subject matter and determine strategies for delivering the appropriate help needed by the student. Halff indicates that, even though the development of a number of instructional guidelines (i.e., step theory) and technological tools (i.e., model tracing) represents true progress in the field of ICAI/ITSs, the following major issues regarding curriculum and instruction are unresolved:

- Design principles needed to specify the range of tutoring applications and structure of that range do not exist.

- Precise mechanistic theories that can account for the effectiveness of particular instructional techniques have not been formulated.

- Clear notions of what constitutes an instructional principle and what constitutes an instructionally useful aspect of some particular domain are also not available.

Tennyson and Park (1987) indicate that most ICAI systems have focused on the full development of any one of the three components cited above, and not all of them.

2.3.2 Examples of ICAI/ITS Applications

Hannum (Personal Conversation, 1989) states that one of the goals of ICAI developers has been to develop systems that teach problem solving. Johnson (*Intelligent...*, 1988) reports that although there are a number of ICAI/ITSs in development today, most systems have been developed as laboratory tools to test various methods of computerized instruction and that there are very few expert systems that have been designed to provide instruction.

Although applications of true ICAI/ITSs are currently limited, the following examples represent various emphases of ICAI/ITS development and the possible use of expert systems in training:

- DEBUGGY - Developed by researchers at Xerox Palo Alto Research Center for research purposes. DEBUGGY is an attempt by knowledge engineers to develop instructional systems that are capable of developing models of users (Harmon and King, 1985).

- GUIDON - Developed at Stanford University by William Clancey as a follow-up to MYCIN, a medical diagnosis and prescription system. GUIDON is designed to train medical school students to conduct consultations and contains the entire knowledge base and actual case experiences of MYCIN (Harmon and King, 1985).

- IMTS (Intelligent Maintenance Training System) - Developed by the University of Southern California in cooperation with Search Technology, Inc., IMTS is designed to train individuals in troubleshooting skills and to conduct research in intelligent tutoring (Polson and Richardson, 1988)

- SCHOLAR - Developed by Carbonell, SCHOLAR is one the earliest ICAI/ITSs and engages in English dialogue to teach facts about South American geography. SCHOLAR uses semantic nets of objects to represent the knowledge base. The Socratic method of probing is used in an attempt to allow the student to discover errors (Garnham, 1988).

- SOPHIE II, SOPHIS III (Sophisticated Instructional Environment) - Three successive generations of a system for tutoring electronics. SOPHIE III is very different from SOPHIE I, because the underlying expert is based on a causal model instead of a mathematical model. SOPHIE III contains three modules: the electronic expert, the troubleshooter, and the coach (Polson and Richardson, 1988).

- STEAMER - An advanced CAI instructional system designed to instruct Naval officers about the problems of running the steam propulsion plants that power many naval ships. Although STEAMER is not an expert system directly involved in instruction, STEAMER is an interactive simulation which is a very effective instructional system designed to handle a difficult problem. STEAMER provides some procedural knowledge but primarily provides conceptual knowledge (Harmon and King, 1985).

- WEST - Provides on-line coaching for a mathematics game, compares the student's performance to the expert solutions, constructs a model of the user's misconceptions, and suggests alternative solutions or strategies (Polson and Richardson, 1988).

- WHY - A sophisticated follow-up of SCHOLAR. WHY uses a mixed initiative dialogue, a more comprehensive classification of the types of possible student misconceptions, and teaches more complex subject matter (Garnham, 1988).

2.4 SUMMARY OF DIFFERENCES BETWEEN CAI AND ICAI/ITS

Although CAI and ICAI/ITS have similar goals, there are fundamental differences between the two systems. Some differences are:

- Goals of developers - CAI was primarily developed by educational researchers and training developers in an effort to solve practical problems through the application of computer technology. ICAI/ITS development was initiated primarily by computer technologists in an effort to explore AI technology. Hannum (Personal conversation, 1989) states that one of the goals of iCAI/ITS developers has been to develop systems that teach problem solving.

- Theoretical base - CAI principles of learning and instruction were influenced by Skinnerian behaviorism and were later improved by allowing CAI authors to incorporate various instructional principles in the design and development process. ICAI principles of learning resulted from the effort to understand the cognitive models of thought processes. ICAI/ITS relies on cognitive psychology for its direction and philosophy.

- Structure and process of the system - CAI systems usually store all instructional components in a single structure, always initiate the instructional process, and offer the student little instructional initiative. ICAI/ITS systems process the stored knowledge, diagnose the student's learning needs, and prescribe instructional treatment for each student.

- Methods of structuring knowledge - CAI uses task analysis for identifying subtasks and content areas while ICAI/ITS uses AI knowledge representation techniques or cognitive analysis to organize domain knowledge into a data structure and procedures (inference mechanisms).

- Student modeling - Park, Perez, and Seidel (1987) state that CAI utilizes binary evaluation of student responses, predetermined system response procedures, and quantitative methods, while ICAI/ITS uses qualitative evaluation of student responses. Dede and Swigger (1988) state that ICAI/ITS systems "understand and purposefully capture (1) the mental dynamics that occur within the student and (2) the progression of instructional process and tasks."

- Instructional formats - CAI uses assorted formats (i.e., tutorial, drill and practice, games, and simulation), while ICAI/ITS uses primarily tutorial/inquiry and games.

- Subject matter areas - CAI has been used in many subject areas while ICAI/ITS has been limited to well-structured subject areas such as electronic repair. ICAI/ITS researchers often choose to explore AI technologies and then choose the subject domain for its adaptability to the technology.

- Validation methods and criteria - CAI determines the success of the program by its effectiveness and efficiency in teaching users. ICAI/ITS success has been determined by the system's capacity to perform, technically, according to design.

- Hardware and software - CAI primarily uses general-purpose hardware, programming languages, authoring languages, and authoring systems for developing and delivering programs. Most ICAI/ITS systems have been developed and implemented using specific hardware designed for AI research. ICAI/ITS software development languages have been generally limited to LISP and Prolog. ICAI/ITS authoring programs are becoming more available for microcomputers.

- Machine learning (the ability of the system to improve its own performance) - CAI technology does not support machine learning because every facet of the program is predetermined. Development of sophisticated machine learning capabilities would greatly improve expert systems and ICAI/ITS. Present expert systems are only capable of developing new rules that are similar to rules that already exist (Garnham, 1988).

Figure 3.1 is condensed representation of the differences between CAI and ICAI/ITS.

ISSUE	CAI	ICAI
Goals of developers	To solve practical problems of educational researchers and training developers	To develop systems that teach problem solving
Theoretical base	Learning principles based on behavioral psychology	Cognitive science model
Structure and process of system	Instructional components stored and implemented in one structure	Models of the task, the student, and teaching discourse stored in separate structures
Methods of structuring knowledge	Task analysis for identifying subtasks and content areas	AI knowledge representation used to organize knowledge into data structure
Student modeling	Binary evaluation of student responses; predetermined responses; quantitative methods	Qualitative evaluation of student responses
Instructional format	Assorted formats (tutorial, drill and practice, games, and simulation)	Tutorial/inquiry and games
Subject matter	Almost any subject domain	Narrowly defined subject domains
Validation methods	Degree of instructional effectiveness and efficiency determined by formative and summative evaluation	No systematic evaluation; technical debugging and functional running of system
Hardware and software	General purpose hardware, languages, authoring languages, and authoring systems	AI-purpose hardware; primarily LISP and PROLOG
Machine learning	Predetermined program facets	Ability of system to self-modify or self-teach

Fig. 3.1. CAI and ICAI differences (Dede and Swigger, 1988; Hannum, personal conversation, 1989; Park, Perez, and Seidel.

3. TRENDS FOR ICAI/ITS DEVELOPMENT

A further focus of AI will be the challenge of machine learning, the development of intelligent machines that synthesize knowledge automatically with the ability to add to a static knowledge base (Harmon and King, 1985).

The following trends will have effects on ICAI/ITS development. Computer hardware and sophisticated software are available to support ICAI/ITS and will have a positive impact on training. As larger memory, more economical hardware, increased storage capacity, optical storage devices, interactive optical discs, and Compact Disk-Interactive (CD-I) technologies advance and are more widely used, costs will decrease.

On-line expert advisors will be developed by users through the use of expert system shells or tool kits (Hannum, *Factors...*, 1989). Voice processing features, videodisc, and parallel processing are expected capabilities for the shells of the 1990s with prices ranging from $100 to $700,000 (Eliot, 1989).

It is expected that advances in natural language processing will influence the ability of computers to understand human speech. A great amount of money has been committed to long-term projects on speech perception and production (Garnham, 1988).

The use of natural language processing makes it possible for an expert system to reason by converting plain text into syntactic and conceptual data structures (Beckman and Rogers, 1987). "Natural language processing is the ability of a computer to understand and respond with sentences that are complete and normal to a user. Understanding the nuances of language will take many more years of development" (Halliday, 1989). Speech recognition still has limitations, even though there are promising possibilities. Computers have a difficult time distinguishing similar sounding words, vocabulary is small, and the computer is sensitive to the pitch and tone of the speaker. AI researchers find it easier to design a computer to talk than to listen (Halliday, 1989).

The current technology of natural language processing has attained success when vocabulary is limited to a few words and tasks are well-defined. The ability to process a language full of metaphors, idioms, and ambiguities that require inferences has not yet been achieved.

With the work force of the 1990s becoming smaller because of the "birth dearth," the expertise gained through experience is recognized as a corporate asset that expert systems can help preserve (Soat, 1989).

Machine learning or neural network technology, which crudely imitates the structure and workings of the brain, has existed in theory since the late 1950s and is said to be one of the most interdisciplinary technologies emerging from the laboratory, drawing on the fields of biology, psychology, physics, electrical engineering, and computer science. Neural network technology has emerged recently since computers have become powerful enough to model neural networks (Ubois, 1989).

4. RESEARCH ISSUES

AI research today is concerned with developing natural language processing, "computer programs that can read, speak, or understand language as people use it in everyday conversation," smart robots, and programs which simulate the behavior of human experts by using symbolic knowledge (Harmon and King, 1985).

Years of work by talented people have been required to build successful ICAI/ITS projects. Although many of these projects have progressed through several prototypes, very few have left the laboratory (Wenger,1987). Research is needed in the areas of expert models, student models, curriculum and instruction, application, and evaluation. Training people to use and develop expert systems is a priority.

4.1 EXPERT MODELS

Because work in AI has progressed with little regard for the cognitive process, theories of learning have yet to be incorporated into tutoring systems (Anderson 1988). Anderson identifies the following items of basic research as being needed in the area of expert modules before the extensive use of ICAI/ITS can progress:

- A thorough understanding of the human cognitive process and how to model this process.

- A pedagogy founded in a theory of learning. Since there is not yet a tutoring system that uses a learning model in its computations, it seems evident that there is little understanding of the learning processes by which knowledge is acquired.

- Qualitative process models and natural language processing for tutorial dialogues.

- Methods for teaching the use of cognitive science formalisms to curriculum developers.

Anderson (1988) states that most activity in the area of intelligent tutoring systems is occurring as basic research projects with the goal of acquiring additional knowledge rather than the goal of building useful intelligent tutoring systems. However, intelligent knowledge communication, usually considered to be an application, is now a fundamental research direction in AI. According to Wenger (1987):

> ICAI/ITS research is still far from the ideal goal of a system capable of entirely autonomous pedagogical reasoning, purely on the basis of primitive principles, in domain knowledge as well as in pedagogical

expertise. Whether this goal can be fully reached at all, or how soon, or to what extent it is desirable, are still matters of speculation reflecting the state of the entire field of artificial intelligence.

4.2 STUDENT MODELS

VanLehn (1988) indicates research should address the following issues concerning student models:

- The miscellaneous collection of educational and engineering techniques to establish a well-understood cognitive-diagnosis technology.

- The effectiveness of fine-grained modeling, a student model that describes cognitive process at a high level of detail, versus coarse-grained modeling, a student model that does not describe detailed cognitive processes.

- The following ways to improve student modeling: (1) Applying specific models of learning to diagnosis in order to reduce the space the diagnostic algorithm must search, (2) employing interactive diagnosis, the skill of posing problems, which is a technique that offers almost as much potential as diagnosis, the skill of interpreting the student's answers to problems, and (3) employing chronometric data (i.e., system monitors amount of time between student's actions) for deciding between potential models of human cognition as interfaces improve and computers become more economical.

4.3 CURRICULUM AND INSTRUCTION (TUTORING MODEL)

Halff (1988) identifies research issues in the field of automated tutors:

- <u>Automated tutors and instructional design</u> - To establish a design approach to automated tutors, laboratories for the systematic manipulation of alternative tutoring methods are needed.

- <u>Theories of learning and instruction</u> - A precise theory of learning is lacking in the approach to instruction.

- <u>Modularity: the independence of instructional and domain knowledge</u> - Halff cites the modularity hypothesis that suggests diagnostic and instructional modules can be used across a broad range of domains. Halff suggests conducting studies on tutoring shells or tutor generators, which should develop the rules that guide the design of automated tutors, and conducting studies of propaedeutic representation (i.e., representing knowledge for instruction).

4.4 THE APPLICATION OF ICAI/ITS

Johnson ("Pragmatic..," 1988) identifies the following goals for near-term research and development of ICAI/ITS:

- Refine existing tools by developing ICAI/ITSs - Move to real-world applications and identify software and hardware limitations.

- Explore the development and delivery of ICAI/ITSs on microcomputers - Compare the development capabilities and costs of microcomputers and AI workstations.

- Use existing expert shells for ICAI/ITSs - Use the off-the-shelf frameworks to produce expert modules and then develop and integrate the other portions into the ICAI/ITS.

- Study the cognitive aspects of user interfaces - Place adequate emphasis on the user interface and on the extent to which the user can easily understand and learn from the system.

- Commit to evaluation - Focus on the evaluation of existing ICAI/ITS development tools and the assessment of the value of ICAI/ITSs over conventional CAI and over nonautomated training.

- Integrate intelligent job aiding with intelligent training - Coordinate the efforts of the researchers in the fields of intelligent job aiding and ICAI/ITS.

Johnson states that the field of ICAI/ITSs is in its infancy, requiring labor-intensive development efforts, but ICAI/ITS science and technology is ready for preliminary application.

4.5 EVALUATION OF ICAI/ITS

Littman and Soloway (1988) identify four main issues focusing on the evaluation of ICAI/ITS:

- More examples of evaluations - Models for ICAI/ITS evaluation need to be developed by educational evaluators and designers of ICAI/ITSs.

- Analytic methods for evaluation - Standard techniques for designing educational studies to evaluate the effectiveness of ICAI/ITSs should be developed.

- Partial process models - A means to perform an ICAI/ITS evaluation with only partial models of the student and an incomplete ICAI/ITS should be developed. The integration of the design and evaluation of the ICAI/ITS with the elaboration of the process model may present a solution for this problem.

- Measuring system for hard and easy bugs - Identify similar patterns of bugs in different domains.

4.6 SYNTHESIS OF RESEARCH SUGGESTIONS

Richardson (1988) synthesizes the research and application suggestions made by the contributors (Anderson, VanLehn, Halff, Johnson, and Littman and Soloway) in *Foundations of Intelligent Tutoring Systems*. Richardson makes the following distinction between research and applications: Research is "concerned with the additional knowledge and understanding to build ICAI/ITS," and applications are "concerned with building, with the available knowledge, ICAI/ITSs that can meet the instructional requirements of individuals and organizations."

Richardson's recommendations for basic research are:

- Meta-theory of expert knowledge - Establish a solid foundation in knowledge representation by building a meta-theory of expert knowledge that explains how declarative, procedural, and causal knowledge relate.

- Causal reasoning and qualitative simulation - Develop ICAI/ITSs that investigate tutoring through the use of qualitative simulation and the cognitive theories of qualitative simulation.

- Natural language and tutorial discourse - Develop a theoretical approach for investigating the linguistic character of tutorial discourse, focusing on one-on-one teaching situations.

- Realistic student modeling - Model the acquisition of expert-level skills.

5. ADDITIONAL ISSUES

Issues concerning the implementation and performance of ICAI/ITSs, the integration of learning theories and instruction design principles, and other current concerns are identified as follows (Johnson, "Pragmatic...," 1988):

- Possible areas for application of ICAI/ITSs should be evaluated according to the following considerations:
  - ICAI/ITSs require programming tools and hardware that may not exist when the project begins.
  - Programming environments and hardware capabilities are always in a state of flux.
  - ICAI/ITS development is a labor-intensive effort. Evaluation of ICAI/ITSs requires substantial resources.
- CAI and ICAI/ITS are suitable for areas of application that have the following characteristics:
  - Continuous significant number of students
  - Expensive equipment
  - Unavailable equipment
  - Unsafe equipment
  - Critical skill and knowledge needed
  - Training conducted at remote sites
  - Low availability of instructors
  - High public visibility
  - Need for high volume of recurrent training
- In addition to these characteristics, the following questions must be answered before it is determined that an ICAI/ITS is feasible:
  - Can the area for expertise be clearly defined?
  - Does the human expertise exist in this area?
  - Can human experts communicate their knowledge?
  - Does an ICAI/ITS authoring system and approach fit the needs of the training system?

- Do human and computer resources exist to develop, implement, evaluate, and support the ICAI/ITS?

Tennyson and Park (1987) state that the overall performance of ICAI systems is not satisfactory and cite the following shortcomings as characterized by D. Sleeman and J. S. Brown:

- An inappropriate level of detail of instructional material is produced in response to a student's response or mistake.

- The ICAI systems are not capable of working within the student's own conceptualization to diagnose the student's "mind bugs".

- Excessive ad hoc tutoring and critiquing strategies are used.

- User interaction is too limited.

Tennyson and Park also indicate that the integration of learning theories and instructional design principles would greatly advance the performance of ICAI systems, but the following limitations make development difficult:

- A natural means for communication between the student and the computer is lacking.

- The lack of understanding of the different reasoning processes of individual students limits the system's ability to capture a student's learning process from the knowledge representation of the expert module.

- The labor-intensive work and technical skills required to encode the knowledge domain and the tutorial strategies are significant.

- AI techniques are not as clearly applicable in less structured subject matter domains.

- Most ICAI systems have focused on the development of man-machine capabilities rather than on the issues of learning and instruction.

Tennyson and Park conclude that future development of ICAI systems should be based on instructional theories.

Development of standards is an important issue in the rapid expansion and achievement of the potential of any area of computer science. Although standards usually limit the freedom of researchers to explore all potential avenues, the development of standards is critical to the commercial development and maturity of expert system technology. At present, the lack of standards is due to the immaturity of the technology (Ferris, 1986).

Eliot (1989) states that validation of the expert system is important but is usually more difficult than validation of systems written in a conventional manner. Maintenance of the expert system is critical, because the knowledge of the expert system may change often and is more difficult to update. Eliot indicates that acquisition of knowledge from the domain expert is a difficult task in the development of the expert system.

The capability of AI is difficult to evaluate because of the lack of discrimination between the descriptions of research projects that attempt to demonstrate the potential of a new technology and the operational system that actually provides profitable results to the developer. The development cycle of an operational AI system is long, and there is very little published information because the systems are still in development. Also, companies using AI technology in their operations are sometimes reluctant to provide the results for fear of losing their competitive advantage (Schoen and Sykes, 1987).

6. SUMMARY

AI technology has now been recognized as having realistic potential for applications in problem solving and is moving from being a research tool to being a viable commercial technology. Industry and government have made efforts in the area of AI Applications and the commercial market has recently become visible (Schoen and Sykes, 1987).

Artificial intelligence has advanced the use of cognitive science techniques, influenced the training environment with the development of the expert system, and altered traditional computer aided instruction (CAI).

The potential of cognitive science contributions to ICAI/ITS is great, but there is much to be determined about the application of cognitive science to the technology of ICAI/ITS. Most of the efforts have focused on the development of man-machine capabilities, and there is much needed development on the issues of learning and instruction. ICAI/ITS is more easily developed in a narrowly defined subject domain, like electronic repair, than in broader subject domains. Significant labor-intensive work and technical skills are required to encode the subject domain knowledge and tutorial strategies.

The introduction of the expert system or knowledge system is expected to transform the training environment. Complex skills will become easier to teach when users are assisted by expert programs. The use of the expert system will change the role of what we know as traditional training as trainers learn to analyze problems and specify exactly what knowledge is necessary for job performance (Harmon and King, 1985). Trainers will learn to apply concepts that have been developed by knowledge engineers to address training problems, develop detailed job descriptions, and analyze and design training programs. Tasks will be analyzed differently due to concepts that are fundamental to the development of expert systems. Harmon and King (1985) state that the power of experts exists in the large amount of knowledge stored in their memories and that experts' insights can be transferred to programs used for training.

The feasibility of producing a fully developed ICAI/ITS is hampered by missing pieces and unproven factors. Development of standards for languages and programming tools are critical to the achievement of the potential of ICAI/ITS. Standards for ICAI/ITS development and evaluation have yet to be established due to the lack of maturity of the technology. The components of ICAI/ITS (expert module, student module, and tutoring module) have been developed primarily as separate efforts and have not been integrated into a comprehensive ICAI/ITS model. Each of these components requires further research, development, and standardization.

The investment and resources for developing a comprehensive ICAI/ITS are substantial, considering all the areas of needed research and unknown issues. Many ICAI/ITSs are operational, but most function in narrowly structured knowledge domains. After more is known about a comprehensive ICAI/ITS, the payoff on the investment will be great, but caution should be applied in trying to make widespread use of ICAI/ITSs for all instructional applications.

Trends, such as computer hardware, software, machine learning and neural networks, tool kits, and natural language processing, will have effects on ICAI/ITS development in future years. As the technology advances and is more widely used, the costs for development of ICAI/ITS are expected to decrease.

ICAI/ITSs cannot replace all other forms of instruction. Presently, because ICAI/ITSs are still in a stage of early development, the best use of ICAI/ITSs is problem solving in a restricted domain of knowledge (Hannum, Personal conversation, 1989).

7. REFERENCES

Anderson, J. R., "The Expert Module," *Foundations of Intelligent Tutoring Systems*, eds. M. C. Polson and J. J. Richardson, Lawrence Erlbaum Associates Publishers, Hillsdale, NJ, 1988.

Anderson, V., "Experts on Expert Systems," *CBT Directions*, Weingarter Publications, Inc., Boston, MA, October 1989.

Beckman, T. J. and Rogers, T. H., *Trends in Development of Applications Using Artificial Intelligence Technology*, Research Division, Office of the Assistant Commissioner (Planning, Finance, and Research), 1987.

Bowen, T. S., "Intellicorp to Market 3M-Developed Expert System," *Digital Review*, September 11, 1989.

Burns, H. L. and Capps, C. G., "Foundations of Intelligent Tutoring Systems: An Introduction," *Foundations of Intelligent Tutoring Systems*, eds. M. C. Polson, and J. J. Richardson, Lawrence Erlbaum Associates Publishers, Hillsdale, NJ, 1988.

Carbonell, J. R., "AI in CAI: An Artificial Intelligence Approach to Computer-Assisted Instruction," *IEEE Transactions on Man-Machine Systems*, 11(4) (1970).

Chabris, C. F., *Artificial Intelligence & Turbo C.*, Dow Jones-Irwin, Homewood, Illinois, 1989.

Dede, C. and Swigger, K.,"The Evolution of Instructional Design Principles for Intelligent Computer-Assisted Instruction," *Journal of Instructional Development*, Vol. 11, No. 1, 15-22, 1988.

Eliot, L. B., "Expert Systems in the IC: The Expert Systems Renaissance," *Information Center*, June 1989.

Feigenbaum, E. A. and McCorduck, P., *The Fifth Generation*, Addison-Wesley Publishing Company, Inc., Reading, MA, 1983.

Feigenbaum, E., McCorduck, P., and Nii, H. P., *The Rise of the Expert Company: How Visionary Companies Are Using Artificial Intelligence to Achieve Higher Productivity and Profits*, Times Books (div. of Random House), New York, 1988.

Ferris, J. K., *The Knowledge Oriented Language for Expert System Development*, Intelligent Machine Company, Arlington, VA, 1986.

Florian, V. and Barros, I., "Using a Knowledge-Based System Shell," *CBT Directions*, July 1989.

Forsyth, R. and Naylor, C., *The Hitch-Hikers Guide to Artificial Intelligence*, Chapman and Hall/Methuen London LTD, New York, 1985.

Garnham, A., *Artificial Intelligence: An Introduction*, Routledge and Kegan Paul Ltd. in association with Methuen, Inc., New York, NY, 1988.

Glaser, R. and Bassok, M., "Learning Theory and the Study of Instruction," *Ann. Rev. Psychol.*, Annual Reviews, Inc., 1989.

Halff, H. M., "Curriculum and Instruction in Automated Tutors," *Foundations of Intelligent Tutoring Systems*, eds. M. C. Polson, and J. J. Richardson, Lawrence Erlbaum Associates Publishers, Hillsdale, NJ, 1988.

Halliday, C., "Is There an Artificial Intelligence in Your Future?" *Computer Buyer's Guide*, pp. 16-19, July 1989.

Handler, R., "The Closest Thing to the Human Brain," *Information Center*, June 1989.

Hannum, W. H., "Factors Impacting IRS Training," *Trends in Technology for Training*, 1989.

Hannum, W. H., Personal Conversation, November 1989.

Harmon, P. and King, D., *Expert Systems: Artificial Intelligence in Business*, John Wiley and Sons, Publishers, 1985.

Harvey, F. A., "Programming in Logic and Implications for the Development of Intelligent Technology-Based Instructional Systems," *30th ADCIS Conference Proceedings*, Philadelphia, Penn., Nov. 7-10, 1988. Western Washington University, Bellingham, WA, 1988.

Hoffman, T., "Artificial Intelligence Drives GM's Charley," *Information Week*, July 3-10, 1989.

Johnson, W. B., "Expert System Approaches to Technical Training," Paper Prepared for Presentation at Southcon '89 in Atlanta, GA, March 1, 1989.

Johnson, W. B., "Intelligent Tutoring Systems: If They Are Such Good Ideas, Why Aren't There More of Them?" *Proceedings of the 10th Annual Interservice/Industry Training System Conference, National Security Industrial Association*, Alexandria, VA, 1988.

Johnson, W. B., "Pragmatic Considerations in Research Development and Implementation of Intelligent Tutoring Systems," *Foundations of Intelligent Tutoring Systems*, eds. M. C. Polson and J. J. Richardson, Lawrence Erlbaum Associates, Publishers, Hillsdale, NJ, 1988.

Jonassen, D. H., *Instructional Design for Microcomputer Courseware*, Lawrence Erlbaum Associates, Publishers, Hillsdale, NJ, 1988.

Kearsley, G., *Artificial Intelligence and Instruction: Applications & Methods*, Addison-Wesley Publishing Company, Inc., Reading, MA, 1987.

Kemske, F., "A New Breed of Expert," *Information Center*, June 1989.

Kirrane, P. R. and Kirrane, D. E., "What Artificial Intelligence Is Doing for Training," *Training: The Magazine for Human Resources Development*, July 1989.

Levine, A. S., "FMS Crafts Cash Flow Expert System," *Federal Computer Week*, August 18, 1989.

Littman, D. and Soloway, E., "Evaluating ITSs: The Cognitive Science Perspective," *Foundations of Intelligent Tutoring Systems*, eds. M. C. Polson, and J. J. Richardson, Lawrence Erlbaum Associates, Publishers, Hillsdale, NJ, 1988.

Minsky, M., "Matter, Mind, and Models," *Semantic Information Processing*, ed. M. Minsky, MIT Press, Cambridge, MA, 1968.

Norville, V., "Expert Systems," *Federal Computer Week*, July 24, 1989.

Park, O. C., Perez, R. S., and Seidel, R. J., "Intelligent CAI: Old Wine in New Bottles, or a New Vintage?" *Artificial Intelligence and Instruction: Application and Methods*, ed. G. Kearsley, Addison-Wesley Publishing Company, Inc., Reading, MA, 1987.

Polson, M. C. and Richardson, J. J., *Foundations of Intelligent Tutoring Systems*, Lawrence Erlbaum Associates, Publishers, Hillsdale, NJ, 1988.

Psotka, L., Massey, D., and Mutter, S. A., *Intelligent Tutoring Systems: Lessons Learned*, Lawrence Erlbaum Associates, Publishers, Hillsdale, NJ, 1988.

Rich, E., *Artificial Intelligence*, McGraw-Hill, New York, 1983.

Richardson, J. J., "Directions for Research and Applications," *Foundations of Intelligent Tutoring Systems*, eds. M. C. Polson, M. C. and J. J. Richardson, Lawrence Erlbaum Associates, Publishers, Hillsdale, NJ, 1988.

Schmidt, R. and Lazar, P., "Expert Training Systems," *CBT Directions*, July 1989.

Schoen, S. and Sykes, W., *Putting Artificial Intelligence to Work: Evaluating & Implementing Business Applications*, John Wiley and Sons, Inc., New York, NY, 1987.

Siegel, P., "What Expert Systems Can Do," *Training & Development Journal*, September 1989.

Skinner, B. F., *The Technology of Teaching*, Appleton-Century Crofts, New York, 1968.

Soat, J., "Nothing Artificial About Expert Systems," *Information Week*, August 21, 1989.

Tennyson, R. D. and Park, O. C., "Artificial Intelligence and Computer-Based Learning," *Instructional Technology: Foundations*, ed. R. Gagne, Lawrence Erlbaum Associates, Publishers, Hilldale, NJ, 1987.

Tucker, M., "AI Woos MIS Anew," *Computer World*, June 1988.

Ubois, J., "Neural Networks," *Federal Computer Week*, October 23, 1989.

VanLehn, K., "Student Modeling," *Foundations of Intelligent Tutoring Systems*, ed. M. C. Polson, and J. J. Richardson, Lawrence Erlbaum Associates, Publishers, Hillsdale, NJ, 1988.

Wenger, E., *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*, Morgan Kaufmann Publishers, Inc., Los Altos, California, 1987.

Whitby, B., *Artificial Intelligence: A Handbook of Professionalism*, Ellis Harwood Limited, Publishers, New York, NY, 1988.

Williamson, M., "PC Week Book Excerpt: The History of Artificial Intelligence, Part I," *PC Week*, October 1, 1985.

Section 4

EMBEDDED TRAINING SYSTEMS

TABLE OF CONTENTS

LIST OF FIGURES

EMBEDDED TRAINING SYSTEMS

1. BACKGROUND AND DEFINITIONS

One of the highest priorities in the Department of Defense is the efficient use of human resources (Hayes and Sherrard, 1981). A method of increasing overall efficiency is to increase the productivity of personnel in the execution of their tasks by developing methods whereby all computer-based systems teach the potential operators how to interact with them. Embedded training (ET) is suggested as this alternative instructional method.

Embedded training means different things to different people, as the following set of definitions indicates:

1. Embedded training is a computer-based process for training and evaluating system users in individual or collective skills, utilizing the resources of the object system and associated system support centers (if any) to prepare, present, and evaluate the course of instruction. Embedded training may be configured to operate independently of, under the control of, or sharing functions with computer programs of the operational system. The objective of embedded training is to use the system to provide users of computer-based systems with initial training and sustained practical exercise in those skills and procedures required to effectively employ and operationally maintain the operational system to accomplish system missions (Baker, 1980).

2. Embedded training is on-line instruction which is an integral part of a system or product; instead of isolating learning from what is to be learned, embedded training makes it an ongoing aspect of that system or product (Kearsley, 1985).

3. Embedded training is the concept of combining computer-based training with application software. The term "embedded" characterizes the way in which training is used as opposed to the process by which training is delivered. The question, "Will CBT be combined with the application software, or will it be separate?" is key in deciding whether or not to embed training (Andersen, 1986).

4. The official definition of embedded training by the Department of the Navy is: "training that is provided by capabilities built into or added into operational systems, subsystems, or equipment to enhance and maintain the skill proficiency of fleet personnel" (Reynolds, et al., 1987).

5. The official definition of embedded training by the Department of the Army is: "training that is provided by capabilities designed to be built into or added into operational systems to enhance and maintain the skill proficiency necessary to

operational systems to enhance and maintain the skill proficiency necessary to operate and maintain the equipment end item" (Finley, et al., 1988).

6. Embedded training is defined by Finley (et al., 1988) in a ten-volume series on *Implementing Embedded Training* as that training which results from feature(s) incorporated into the end item equipment to provide training and practice using that end item equipment. The features may be completely embedded within the system configuration, by software application, or a combination of both software and systems configuration, or may be executed by some form of strap-on (e.g., a video disc player) or plug-in (e.g., a floppy disk) equipment, or a combination of embedded and appended components. The feature(s) must include stimuli necessary to support training; they should include performance assessment capability, appropriate feedback, and record keeping.

7. Embedded training refers to training delivered through the workstation when an employee is completing some aspect of a job. This training is of short duration and very job focused. Embedded training is meshed with ongoing work. The employee moves freely between completing a normal task and receiving training when necessary for completing that task. The movement between work and training is transparent. The intent of embedded training is to increase worker productivity while reducing training time and costs. The training is resident on the workstations the employees are using so training can be accessed when needed. Embedded training typically focuses on procedural knowledge rather than declarative knowledge. That is, the training teaches someone how to do something, not information about something. Thus, embedded training is not often used for the initial introduction to a new area (Hannum, 1989).

Common to the various experts' definitions are certain basic concepts. Embedded training is training that is an integral part of a computer-based system, so that a user may move easily from performing work on the system to receiving training on the system as required. The training component may be integrated into the applications software or reside in its own program, which runs concurrently with the applications software. The intent of this paper is not to arrive at a common definition of embedded training but to explore a range of possibilities and potential applications of embedded training to meet a variety of training requirements for computer-based systems.

2. EMBEDDED TRAINING APPROACHES

Literature has identified two approaches to embedded training: integrated and concurrent. In the integrated approach, the training courseware is embedded into the applications software. Concurrent training is a variation on the theme of providing training during the use of applications software. In concurrent training, the computer-based training material resides in its own program which is running alongside, or concurrent with the applications software (Hannum, 1989).

In the integrated approach, training and applications software are combined by embedding training directly into the code of the application. The training and application are integrated into one system. Generally the programming language used to write the applications program, such as COBOL or C, is also used to create the code for the training components.

In the concurrent approach, training and applications software are combined but are not integrated. Using two separate systems, the training system and the application system, training is overlaid onto the application. The systems are separate but run concurrently (Andersen, 1986). Concurrence allows the development of dynamic help systems that take advantage of existing program screens. This not only allows the user to be coached on a specific command or task, but user's keystrokes can be sent to the application to perform the desired operation. This enables the system to be an intelligent assistant, not just a passive help system.

With both integrated and concurrent training, people receive training at their workstations when it is needed. The training is incorporated as part of the work such that a person moves freely from work to training then back to work (Hannum, 1989). The distinction is whether the training material is part of the applications software or is a separate program running at the same time. To the user, both approaches appear to respond in the same manner, but developmentally they are very different. Figure 4.1 identifies the differences between integrated and concurrent training approaches.

APPROACH	TECHNICAL	PROGRAMMING LANGUAGE	EXPERTISE REQUIRED	TRAINING EFFECTIVENESS	COST	ISSUES
INTEGRATED	TRAINING IS AN INTEGRAL PART OF THE APPLICATION	GENERAL PURPOSE LANGUAGE	INSTRUCTIONAL DESIGNER & PROGRAMMER	HIGH	HIGH	- UP FRONT PLANNING - MEMORY - STORAGE - HARD TO RETROFIT
CONCURRENT	TRAINING IS SEPARATE & USED IN CONJUNCTION WITH THE APPLICATION	CAI LANGUAGE AND/OR CAI AUTHORING SYSTEM	INSTRUCTIONAL DESIGNER & PROGRAMMER (FOR CAI LANGUAGE)	HIGH	MEDIUM	- MEMORY - STORAGE - WILL NOT WORK WITH ALL SOFTWARE APPLICATIONS

Fig. 4.1. Embedded training approaches.

3. TYPES OF EMBEDDED TRAINING

Embedded training can range from simple "help" assistance to a complete tutorial that teaches a novice how to use an applications package (Hannum, 1989). A user can be led through a simulation exercise to practice using an applications package or receive training only on request. Though embedded training can take many forms, three general approaches are help systems, tutoring systems, and simulations. A help system provides brief assistance to the user who may have forgotten how to perform a certain procedure. A tutoring system provides more training to the user than a help system but still allows the user to toggle back to an application program. Simulations provide the user with an opportunity to practice an application program without altering live data.

3.1 ON-LINE HELP SYSTEMS

Help systems are the most prevalent form of embedded training and usually consist of a set of screens containing summaries of commands, quick reference tables, and brief descriptions of system features. There is a variety of different levels, types, and ways to provide embedded training in the form of help systems. In each case, the help system is designed to assist the user with completing procedural steps in an application. Help systems are not complete training systems because the information provided is brief. The training that help systems provide is more appropriate for refresher training than initial training.

3.1.1 Levels of Help Systems

Hannum (1989) describes four levels of help systems. The simplest form, or first level, of a help system is a list of acceptable commands or acceptable inputs to the application program. For example, when users encounter input fields in an application program and cannot recall how to enter the data, they can toggle to the help system where a list of currently available commands is displayed. This type of help system is also called user-initiated help.

Second level help systems go beyond just listing commands to include a description of each command. In a second level help system, the user can access more information about each command or about the form for acceptable input. Each command or input field is described in one or two paragraphs. When the user requests help, these paragraphs are displayed. On-line reference manuals are included in this level of help.

Third level help systems include examples of commands or examples of acceptable inputs. The intent is not to list or describe commands but rather to show what acceptable inputs look like. Users can use these examples as models for their responses. Third level help is also called command-driven or key word help.

Fourth level help systems provide procedural steps the user can follow to accomplish certain tasks in an application program. This level of help is more elaborate than the other levels and provides more information to the user. The fourth level help system can be arranged in an if/then table so users can read the table to find the "if" condition that matches their situation and can see the appropriate "then" response. For example, users are able to get directions for changing an address by looking for an example indicating "if you want to change an address, press enter and type the new address in box 26." This level of help shows what to do as well as the specific steps in the procedure. The fourth level of help may be prompt-driven. For example, the system would prompt the user to ask for help when a user error is detected.

3.1.2 Help Messages

Kearsley (1985) describes five kinds of help messages: (1) fixed formats, (2) context-sensitive, (3) prompts, (4) query-in-depth, and (5) dialogue.

Fixed Formats: A fixed format help message is the same regardless of what the user has done. The most familiar type of fixed format help is an explanation of a command. Fixed-format help messages are the simplest type to implement since they are usually stand-alone text files.

Context-sensitive: In contrast, context-sensitive help messages depend on what the user is currently trying to do. For example, if users are in the middle of a word-processing package and want to add, move, or delete some text, requesting help will display information about editing a document. Later, if users are in the process of formatting the same documents for printing and request help, they will receive information about printing. With context-sensitive help messages, the program attempts to provide information specific to the current activity.

Prompts: Prompting-type help messages are similar to context-sensitive ones except they are generated by the system rather than requested by the user. Thus, when a user types a command string in the wrong format or with incorrect parameters, the system responds by displaying the correct syntax or parameters. A sophisticated help system may actually correct the input and display it to the user for verification.

Query-in-depth: Query-in-depth help messages provide multiple levels of response. Each successive level provides more elaboration than the previous level. Thus, if users ask for an explanation of a particular command in a query-in-depth system, the first message may simply identify the syntax and/or function of the command. A further request for help on this same command may provide examples of use or a description of how the command can be used. Alternatively, it is possible to specify the depth of explanation desired.

Dialogue: Dialogue-type help messages allow the user to ask questions in conversational format. Such help systems rely on an understanding of natural language. In a dialogue-type help system, the level of help is dictated by the level of question, although the system may give answers that are not understood by the user.

3.1.3 Design Issues

Clearly, different kinds of help systems require varying amounts of time to create and maintain (Kearsley, 1985). Fixed-format systems are simple to construct. However, any form of context-sensitive help system requires a considerable amount of time to construct, because all likely problems a user might encounter must be identified and programmed. In a complex program or system, the number of different contextual situations requiring specific help can number in the thousands. In fact, it is not unusual for full-fledged help systems to require as much disk space and take as long to develop as the application program itself.

The choice of level in a help system can be made by the designer or by the user. The designer can decide that all the help in a particular application program should be level one (commands only) or can elect to use level three help (commands and inputs) throughout the program. The designer can decide to include different levels of help at different points in the program. For example, the designer might use a list of commands at one point in the program and examples of commands at another point. The level of help a user could receive would be a function of when in the program the help was requested. In each of these cases, the only influence the user has over the help system is whether or not to request help. The specific help a user receives is not individually tailored; any user asking for help at that point in the program receives the same response. This is called a static help system.

Another option, context-sensitive help, gives more control to the user of the system. The designer could tailor each level of help to the user's level of sophistication or build on the most recent entries the user has made. The designer does not have to decide in advance that everybody should have, for example, the third level of help. Context-sensitivity is achieved by actually reading and analyzing application screens. In this manner, the user can choose to see a list of commands (level one help) or examples of commands (level three help), and the system will respond with appropriate help information based on the user's current status in the program.

The user might also be able to specify the level of help desired in a user profile or in real time when help is actually requested. When establishing their profiles, users can indicate that, when requesting help, they want to receive the procedural steps (level four help). Any request for help made while using the application program results in the level four help for that particular user.

A system design could incorporate another approach allowing the user to request any of the four levels of help by pressing a help function key then pressing a numeral 1 to 4 to indicate the level of help desired. For example, the popular word-processing program WordStar allows the user to adjust the verbosity of the help menus from full descriptions to simple listings.

Another alternative is to have all four levels of help layered so a user will get level one (list of commands) help initially. A toggle can be used to switch back to the application program or to get additional help. The "more help" option activates level two help (descriptions of commands). By employing this layered approach or query-in-depth help system, the user can get progressively more detailed information when needed but avoid elaborate help when it is not necessary.

Not all experts in the area consider help systems to be embedded training. According to Baker (1986), "Many people confuse prompts within a computer program (e.g., the help buttons or explanatory subroutines) with embedded training." In point of fact, if one subscribes to the notion that training implies a structured learning environment that is product-oriented, is job-based (typically derived from job/task analyses), entails some feedback loop to provide the user with knowledge of results, and is assessed in terms of specified performance standards, then embedded program prompts are not, by definition, embedded training.

Historically, help systems have seldom been very helpful. Research points out that different kinds of help systems are required by different users (i.e., novice, intermediate, and expert), based on experience levels. Furthermore, the sophistication of users may change quickly as they use a system or device. Kearsley (1985) proposes that help systems only help if they are very specific to the current needs and sophistication of the users, suggesting that knowing when to provide the right kind of example in a help request is basically an inference problem and is in the domain of artificial intelligence. Thus, the design of help systems is a complex task requiring that many variables be considered.

3.2 TUTORIAL SYSTEMS

Tutorial systems differ from help systems in that tutorials provide more information and instruction to the user. Tutorial systems go beyond providing help in using a command and including instruction for learning to use commands. Embedded training in the form of tutorials resembles Computer Aided Instruction (CAI) more than it resembles help systems in many respects because actual teaching, or tutoring, occurs. As with help systems, Hannum (1989) describes four levels of tutorial systems varying in complexity.

First level tutorial systems are "Three E" systems consisting of an example, an explanation, and an exercise. When a user evokes a tutorial system from an application, an example related to that particular part of the application is displayed. The example is context sensitive and shows how to use a command or complete a procedure. An accompanying explanation elaborates the example. Thus, the user sees what to do by viewing the example and determines how it is done by studying the explanation. Finally, the system presents an exercise to determine if the user understands the example and knows what to do. The user returns to the application program and continues working.

Second level tutorial systems provide an on-line training manual that is indexed to the applications program. Requests for help in a second level tutorial system result in the display of the section of the training manual most closely related to that part of the program in which the user is working. The user can browse through the relevant section then toggle back to the application program.

Third level tutorial systems consist of short programmed instruction segments. As with second level systems, third level tutorial systems are designed to provide context-sensitive information. The particular programmed segment that users see is a function of their location in the applications program when they request assistance. The segments are focused on a single topic and consist of typical programmed instruction frames that present a small amount of information about a topic, ask a question, and provide appropriate feedback.

The fourth level tutorial systems consist of intelligent computer aided instruction (ICAI), also called intelligent tutoring systems (ITS), embedded in the applications that require programs.

The goal of an intelligent tutoring system is to provide instruction that combines the individualized, self-paced benefits of interactive computer-based training with the knowledge, understanding, reasoning, and diagnostic and explanatory capabilities of a competent human instructor (Johnson, 1989). When a user invokes this form of embedded training, the training system attempts to diagnose the problem the user is having, ascertain why it happened, and determine what training will provide a solution. This is a difficult task since it requires a model of what the user knows, a model of how the application program works, a model of the knowledge an expert user has, and a model of what an expert tutor would do in teaching this user how to acquire the missing knowledge. These decisions are made dynamically, not stored in the system's memory.

Figure 4.2 presents a very general overview model of the components of an intelligent tutoring system (adapted from Wenger, 1987). In this conceptualization, five distinct models are represented (MacGregor, 1989). The instructional model maintains a record of important aspects of the instructional interaction, such as material that has been covered, problems that have been solved successfully, and explanations that have been given. The student model is a database relating to the student's current understanding of the instructional material, including answers given, concepts mastered, and misconceptions and beliefs that have been formulated. The dialogue model tracks the discourse between the student and the tutoring system and helps ensure a continuity of terms used in interactions with the student. The expert task model is a record of the domain facts, concepts, inferences, and the like that comprise an expert's understanding of the instructional material. Also included in the expert task model are advice and evaluation strategies an expert might use as instructional devices, along with other descriptors of the circumstances under which these strategies are most appropriately used. Finally the pedagogical model

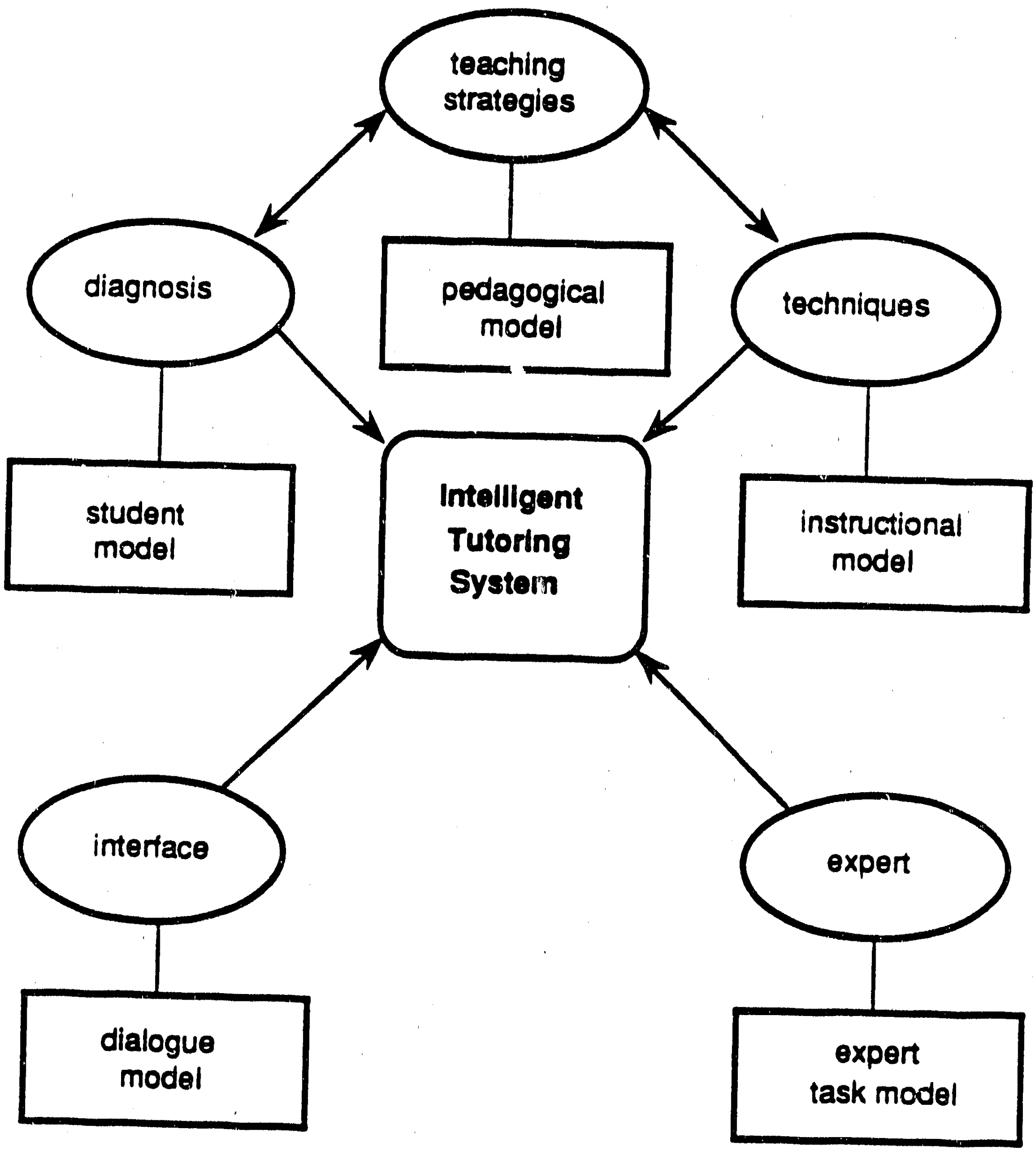


Fig. 4.2 Components of an Intelligent Tutoring System.

represents global strategic decisions about the circumstances under which one of a set of general instructional approaches would be applied, as well as the history of success of the different approaches with an individual user.

Because of the relative complexity of tutoring systems compared to help systems, designers would decide which level of tutoring system to develop for a given application program. It is unlikely that all four levels of tutoring systems would be developed and placed in the same application as is the case with help systems (Hannum, 1989). However, four levels of tutoring systems can be placed in an application program. As with help systems, users can decide which tutoring system level they want in advance or upon request in the embedded training. However, the developmental effort involved in including four levels of tutoring systems is likely to be too great for most purposes. Time and budgets will rarely allow such a luxury for training developers.

Tutorials created with concurrent authoring systems run in tandem with live subject applications. "Because these systems go beyond simulation, the student has the opportunity to learn by using the real software instead of a facsimile" (Sautter, 1987). The result is courseware that facilitates learning-by-doing. The student interacting directly with the live applications system still has the tutorial help available and, can continue working until help is needed.

The practical problem with intelligent tutors is that they are extremely time-consuming to design and implement. A considerable amount of time (often many months or even years) is required to build an adequate knowledge representation and to identify the appropriate tutoring/diagnostic rules (Kearsley, 1985). The design process involves the representation of the subject matter in terms of concepts and their relationships (declarative knowledge) and in terms of if/then rules (procedural knowledge). All of the implicit knowledge about the subject and how to teach it must be made explicit (Kearsley, 1985).

3.3 SIMULATIONS

The use of simulation results in active learning and a high degree of practice, which are two of the most important factors in effective instruction (Kearsley, 1985). In simulations the users interact with the training system as it mimics some or all of the application program. In some instances, the training system will use the actual application program rather than simulating it. The difference between using a simulation as a system for embedded training and working with the actual application programs is that the simulation provides more instructional feedback than the application programs. Most application programs let the user know when an error is made but do little more than flag the mistake. Unfortunately, some application programs accept erroneous data without indication and, thus, even more problems are created. Embedded training simulations let the user respond in a manner similar to the live application but without harming live data.

Hannum (1989) identifies three levels of embedded training simulations on the basis of what is simulated and how it is done. All three levels of embedded training simulations mimic what a user would do in an application program while providing some help or training. Furthermore, well-constructed simulations can guide or coach the user to minimize inefficiency (Kearsley, 1985).

First level embedded training simulations simulate the interface only. A program is developed that looks and behaves like the application program. The screen and keyboard act like the application program, although what is going on behind the scenes may be quite different. It is not necessary for the simulation to process the data in a fashion identical to the application program. Because of this difference in handling the data, data used in an embedded training simulation may have to be restricted. The embedded training simulation may accept identification numbers for only a few people in a few job categories while the application program may accept a wide range of identifications. It is not always necessary or desirable to simulate everything; simulating the interface is often sufficient for training purposes. Simulation can analyze a user's input more closely than most application programs, detect errors and ascertain their cause, and provide more instructive feedback and training as needed.

Second level embedded training simulations use the actual application program but with a training database. Simulated data are used so the person receiving the training cannot compromise the actual database. There is no simulation of the program, only the database. Thus, this form of embedded training simulation acts exactly as the real system. The user is free to experiment with the application program to accomplish certain tasks. The only help or feedback available to the user is that provided by the application program. It is rare to find application programs that make extensive use of training features such as detailed explanatory feedback when mistakes are made.

Third level embedded training simulations use the actual application program with a concurrent training program. With the concurrent training program, the users can toggle from the application program into the training to receive needed assistance then return to the point where they left the application program and continue their work. The training portion of these systems can show the users what to do, let them try it, analyze their errors, give them more elaborate instructional feedback, and then return them to the application program.

Kearsley (1985) believes that an enduring issue in the design of simulations is the degree of realism needed. It would seem that the greater the degree of realism, the more effective the simulation. Simulations that are extremely realistic do not necessarily provide the best learning environment (Kearsley, 1985). A simplified model which reproduces the critical functions of the system may result in faster learning than a highly realistic simulation. Presenting the new user with a complex system typically leads to information overload and frustration. In the final stages of learning, as much realism as possible may be desired to maximize transfer of learning to the actual job or task. Kearsley (1985) proposes that the usefulness of a simulation depends on the quality of the input data for the students to use and the instructional validity of the model on which the simulation is based.

3.4 SUMMARY

The three types of embedded training, help systems, tutoring systems, and simulations, are designed to integrate training with work, reduce time and costs required for training, give the user more control over training, and improve overall productivity. Each type of embedded training represents a promising approach for training. Figure 4.3 identifies the three types of embedded training and the levels of each type (Hannum, 1989).

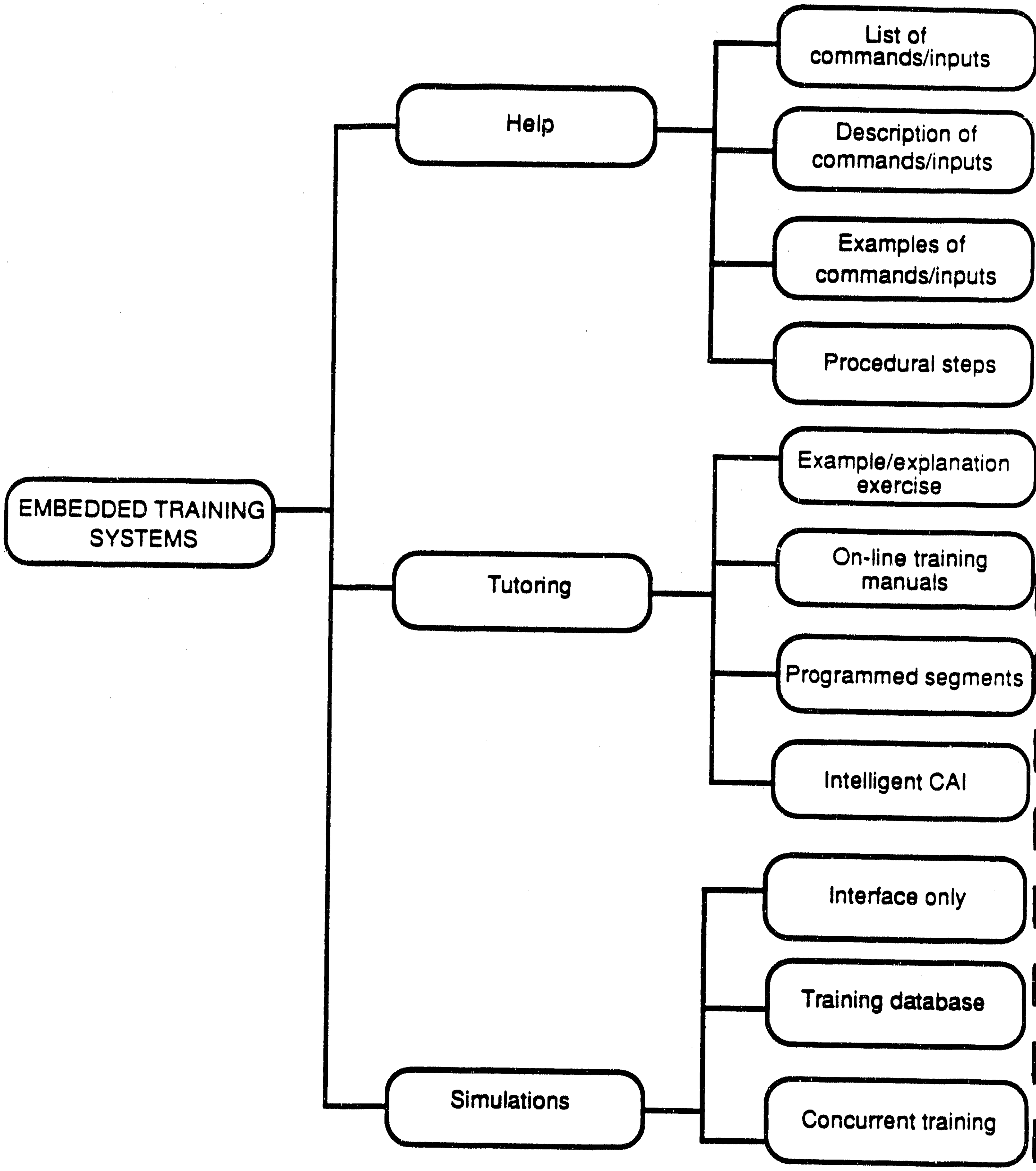


Fig. 4.3. Types of embedded training.

4. APPLICATIONS OF EMBEDDED TRAINING

Embedded/concurrent training is a natural evolution of the automation of many work processes (Hannum, 1989). When the workstation is a computer, use of the system for training is a logical and efficient approach with an added benefit that problems of skill transfer from the training environment to the work environment are reduced. If the training is written so that the learner's use of keys and processes during the training are identical to work applications, a positive transfer of learning is probable.

4.1 CURRENT TRENDS

One of the driving forces behind embedded training is that many new users of computers and microprocessor-controlled devices are not technical personnel and have had no prior exposure to the broad range of skills and concepts that must be mastered before productive work can begin (Hannum, 1989). In addition, many of today's users of automated systems want a high degree of control over the equipment and programs they use (Kearsley, 1985).

Many systems in the military currently employ embedded training capabilities to some extent, although very little is known about the effectiveness of these systems. The analysis and evaluation of existing onboard Navy training systems found few systems that could be considered totally embedded training systems (Reynolds et al., 1987).

Embedded training subsystems range from add-on training subsystems to built-in subsystems (Hardy, 1988). At the low end of the scale, add-on training subsystems can be quickly attached to existing hardware and electronic data connections. At the midpoint are training subsystems that are permanently mounted on the system but are peripheral to the operational hardware. On the high end of the continuum are training subsystems that are totally integrated into the operational hardware.

4.2 MILITARY EXAMPLES

The military has been at the forefront of embedding training into computer-based systems, ranging from sophisticated weapons to automated record-keeping processes. A Tri-Service review of nine selected embedded training systems indicates there is a great diversity among embedded training components within each service and among the services. Each embedded training component reviewed has some favorable characteristics; however, no two systems are similar in design. The only characteristics common to all of the embedded training components reviewed are the perceived benefits of embedded training, the high acceptance level of the embedded training concepts among users, and the fact that each embedded training component uses some type display, input, and control device (Warm et al., 1988). Each system provides skill sustainment training by providing task-related stimuli to the system operators.

A few examples of embedded systems employing computer-oriented features and being used or developed by the military are the Fiber Optic Guided Missile System, the World Wide Military Command and Control System, Ramtek 9400 Color Graphics Display System, and the Abrams Block III.

4.3 NONMILITARY EXAMPLES

A few examples of embedded systems in use in the private sector and in other branches of government include the Westpac Banking Corporation, IBM and DEC, and the National Library of Medicine.

4.3.1 Westpac Banking Corporation

Westpac is currently developing a major new computer system to fully integrate its banking operations at over 2000 offices. Embedded training is being developed as part of the overall system design and consists of four main elements: help, simulation, tutorial, and learning management. The help area is context sensitive and may be requested by the user or generated by the system as a result of operator error. Simulation involves a training database which allows practice in a safe environment. The simulations are linked to appropriate tutorials that monitor and control the use of the simulation. Tutorials provide background information and are able to branch and route learners based on performance. The learning management element collects data about the learner's performance during training and real time usage, and then produces training reports (Bentley, 1989).

4.3.2 Vendors

Many vendors, including IBM and DEC, are offering embedded training programs for both system training and applications software. Tutorials are embedded in a broad range of commercially available software designed to teach users how to operate or maintain the system, including word processing, data entry, database management, typing skill, project scheduling, and resource management. For example, an on-line tutorial is included on the learning diskette of word-processing software packages such as WordPerfect and Microsoft Word. Lotus Symphony is a flexible software package that combines five capabilities in one product and provides five work environments in which to use its capabilities. Each tutorial has unique features and commands. Symphony's Help is an electronic reference manual. When the Help key is pressed, Help provides information about the user's current location. If an error message appears on the screen during work with Symphony, the user can get more information by pressing the Help key. Help is always available and each help screen includes a menu of additional topics, allowing the user to select other screens.

The Xerox Corporation has an interactive embedded training program called STAR*T which teaches the principal features and capabilities of their Ethernet system, STAR (Hart, 1983). STAR includes 14 other embedded training modules which provide an awareness of the system's potential and serve as an electronic job aid.

4.3.3 National Library of Medicine

The National Library of Medicine's on-line information retrieval system called MEDLARS (Medical Literature Analysis and Retrieval System) gives the user access to the world's leading databases of medical information covering over 20 years, 3500 journals, and other specialized databases in fields such as of health administration, toxicology, cancer, population studies, and medical ethics.

The National Library of Medicine is dedicated to full support of all systems users and includes such services as an easy-to-use software package called GRATEFUL MED and several CAI programs such as MEDLEARN, CHEMLEARN, and TOXLEARN.

In addition, MEDLARS on-line help command, EXPLAIN, provides on-line definitions of a variety of terms and features related to the system and its files. EXPLAIN can also be used for on-line explanations of MEDLARS commands or program messages including error messages. The HELP command can be used to display a menu of possible problems and allows the user to select options. Choosing a number from the menu results in a display of possible solutions.

4.3.4 Computer Aided Design (CAD) Systems

The U.S. Army Construction Engineering Research Laboratory (USA-CERL) has developed an AutoCAD training program, based on the embedded instruction concept.

The training design approach involved first studying architects/engineers in the field as they learned to use a computer aided design (CAD) system. A simple computer-based learning environment was designed as an observation tool. An embedded tutorial containing six lessons was then developed within a widely used CAD system. The lessons were programmed in AUTOLISP and AutoCAD graphics were used to create the screen displays. The first two lessons provided a general framework; the next two lessons suggested different design applications; and the last two lessons addressed details of editing and documentation.

During a lesson, users were monitored and feedback was provided if a standard error had been committed or a command had not been practiced. Extra context-sensitive help was offered on some lesson screens when the letter "X" was typed.

The result of the USA-CERL research is a revised, enhanced software package which has been turned over to Electronic Courseware Systems, Inc. and is being marketed as "Teaching Assistant for AutoCAD" (Army Construction Engineering Research Laboratory, 1988).

5. ISSUES AND CONSIDERATIONS

When embedded training is being considered as a training medium, a clear definition must be developed of both the content and the types of training being considered for embedding (Hannum, 1989). Both operational planners and trainers need to agree on the way embedded training should be defined, planned, designed, implemented, and evaluated.

5.1 CONSIDERATION OF EMBEDDED TRAINING AS A TRAINING MEDIUM

The major considerations in whether or not embedded training is appropriate are the nature and complexity of the automated system and the capability and experience of the people who will be using it (Hannum, 1989). Analysis of the automated system should include such factors as:

- Similarity to other systems in terms of user interface - If users are familiar with systems having similar user interfaces, the training needs are simplified. If the interface is new or complex, embedded training is an option to consider.

- Functions and processes - The nature of the system functions and processes which will require instruction and their suitability for the embedded training format should be considered.

- Complexity of tasks - Complex procedural tasks that are difficult to remember or are performed only occasionally are good candidates for embedded training. Developing embedded training to teach simpler tasks may not be cost-effective.

- Prior training or experience of planned users - The prior training and experience of the user will help determine the types and levels of training needed.

In a study sponsored by the Army, an attempt was made to identify criteria for selecting embedded training. A review of eight Army systems was done to evaluate them on 12 attributes, including the nine "Factors of Overall System Embedded Training Decisions" defined by Purifoy et al., (1985) and three additional items. The intent of the review was to ascertain the general characteristics of a range of Army systems and to identify those characteristics that might be most indicative of the probable need for, and success of, embedded training in Army systems (Strasel et al., 1988). The twelve decision factors used in the evaluation are listed in Fig. 4.4.

1. Will the system incorporate a well-defined, consistently used interface that provides for comprehensive information display and system control?
2. Will more than trivial additional displays and controls be required to implement embedded training?
3. Is there a reasonable probability that the system will be available for training utilization, given other system demands?
4. Will training utilization of the system (via embedded training) result in penalties to the reliability, availability, or maintainability characteristics of the primary system?
5. Is it likely that the conversion of the system from training utilization (i.e., embedded training mode) to operational configuration will be time-consuming, require significant effort, or result in problems with system operability after the conversion?
6. Is the computational capability of the primary system sufficient, or inherently expandable, to accommodate embedded training implementation requirements, or can an interface for strap-on computational capability be provided?
7. Will the primary system be subject to frequent or radical changes that could require major changes in the training provided by embedded training?
8. Will embedded training meet training needs more cost-effectively than other possible training approaches?
9. Are there likely to be significant institutional problems with the introduction of embedded training in environments that previously have not had access to interactive training?
10. Do team skill requirements mean that the training requirement may be more complex than would otherwise be the case?
11. Does the need to simulate battle conditions to accomplish effective training make the training requirement more stringent, probably more expensive, and certainly more demanding of the instructor and the instructional program?
12. How many people are to be trained?

Fig. 4.4. Factors to consider in the selection of embedded training.

Other factors which are crucial in deciding if embedded training is to be employed include whether it makes sense for training to influence system design and whether the heavy utilization of resources is justified. Embedded training can have a logistical impact (e.g. planning for equipment deliveries or procuring needed parts), a time impact, and an operating impact on a system. Such effects can contribute to both positive and negative attitudes toward the use of embedded training. Plans for embedded training must be included in the system design stage. Insufficient memory and storage capacity are difficult problems to solve after the fact (Hannum, 1989). Up-front planning for embedding training in operational systems that run on mainframe computers or minicomputers is particularly critical.

Embedded training requires frequent use of the system. Therefore, there is more risk in the areas of system reliability, availability, and maintainability (Hardy, 1988). The parts of the system used for training must be made as rugged as the rest of the system to ensure the maintenance of the required operational rate. For those parts of the training subsystem that are fully embedded in the operational hardware, designers must ensure that failure of the training subsystem does not affect performance of the operational subsystem.

BBN Laboratories suggests a methodology for analyzing embedded training selection and design (Massey, et al., 1986). The process involves the following steps:

- First, an analysis of the training requirements is conducted to determine instructional content. At the same time, an analysis of the hardware and software is conducted to determine the feasibility of embedded training. A small prototype subsystem design effort should be conducted at this point to aid in determining the feasibility and applicability of the various methods of embedded training.

- Once the first step in the analysis is complete, embedded training has been found to be the best training strategy, and the feasibility is established through the prototype, various training techniques and their related technologies are selected. Defining training techniques and related technologies helps to clarify any equipment or software changes required.

- The third major step in the analysis is to determine the impact of the embedded training design on the operational system. The impact on the overall design and development of the system is evaluated in light of the embedded training strategies selected. Modifications to the operational system and to the embedded training are made based on the analysis.

5.2 FEATURES TO BE CONSIDERED

In planning an instructional approach for embedded training, many factors should be considered. A *Tri-Service Review of Existing System Embedded Training Components* was completed in 1988 by Applied Sciences Associates under contract for the Department of the Army (Warm et al., 1988). The objectives of this review were to systematically examine the components and the characteristics of embedded training implementations in a selected set of systems currently operational in the Army, Navy, and Air Force, to explore how embedded training components have been developed and employed, and to attempt to derive principles and "lessons learned" which may be of value in providing guidance for the development of future embedded training components for military systems. Features recommended for consideration include:

- Computer-oriented features - A selection of the best combination of CAI, artificial intelligence, and use of graphics should be considered.

- Training management features - The amount of individualization (adaptive training feature), the type of built-in record keeping for student performance, and the ability of the system to include new materials (particularly scenarios) once the training is in place, are major considerations.

- Automated training features - A decision must be made concerning the amounts and types of feedback the student will receive.

- Scenario control features - Many of the existing embedded training systems in the military present battlefield-type scenarios for the student's use in practicing certain skills; therefore, it must be determined which scenario features should be included, such as allowing the instructor to control the presentation by freezing, playing back, and fast-forwarding the scenario.

- Embedded training and system coordination features - Among the decisions to be made are whether or not the training should be embedded or "strapped-on," can be accessed when the system is off-line, and will be coordinated with other related systems.

All of these factors may have relevance in selecting the best combination of embedded training features to support a particular training application.

5.3 COST ISSUES

Before embedded training can be considered as a practical option in most training situations, the costs associated with production and maintenance must be brought within reasonable limits. Even in their simplest forms, the design and development of embedded training are lengthy and expensive processes.

Major costs are involved in developing and presenting stand-alone CAI. One author suggests that each hour of CAI requires 200 hours, or five weeks, to develop at a cost of $875 per week, so the cost to develop 20 hours of CBT is $87,500 (Webr, 1988). Another author suggests much higher costs, particularly when simulations and complicated answer analysis techniques are employed (Gery, 1987). The costs associated with intelligent tutoring systems are even greater. Cost factors are compounded when the CAI is to be embedded into the actual applications. The embedding process will have a substantial impact on the time and costs associated with the design and development of the system as well as maintenance costs once the system is operational.

Some studies have shown actual cost savings in using embedded training over traditional training approaches. A U.S. Army Corps of Engineers study for the development of embedded training for CAD software found an 8:1 cost advantage for embedded training over classroom training when factors such as salaries and per diem costs were considered (Army Construction Engineering Research Laboratory, 1988). In another cost study, conducted by the Army Research Institute, a savings of six million dollars was expected by employing embedded training for the Tactical Fire Direction System (TACFIRE) compared with the cost of training through schoolhouses and unit training (Germas and Baker, 1980).

Factors that will help reduce the development costs considerably in the future include:

- Development of authoring tools for embedded training that streamline development time by reducing the need for computer programming

- Standards and guidelines for the design and development of embedded training that provide, at a minimum, blueprints for embedded training design consideration

- Sharing information and "lessons learned" on embedded training projects so future projects can avoid some of the mistakes of the past and benefit from past successes

- Advances in state of the art in design of artificial intelligence and expert systems will provide streamlined approaches to developing "intelligent tutoring" capabilities for embedded training. "Artificial intelligence at this time means dealing with logic that is difficult to define and with subjective reasoning made objective, so there is a gigantic set of rules needing to be debugged" (Gralla, 1988). As processes become more defined and systematized, the use of artificial intelligence applications in embedded training will become more feasible.

5.4 GUIDELINES AND STANDARDS

Clear-cut guidelines and standards for embedded training developers are needed. The Army has some existing guidelines such as *A Procedure for Developing Embedded Training Requirements* (Roth et al., 1986) and *Interim Procedures for Embedded Training (ET) Component Design* (Fitzpatrick, et al., 1987). However, the current versions of the guidelines are limited because they are based on work done on systems in which the majority of the system development was completed prior to the design of training and dominated by proceduralized, psychomotor tasks. In the document *Lessons Learned from ET Design Process for ASAS/ENSCE* (Evans et al., 1988), the authors suggest that the guidelines need to be developed for earlier stages of system development, which apply to systems that require more cognitively based training.

Evans et al., (1988) concluded that the existing guidelines for the nomination of embedded training requirements are not comprehensive enough for application to all systems for which embedded training may be appropriate. The review staff determined that the guidelines, as currently written, are not applicable to systems of the following types:

- emerging systems for which there is a limited availability of subject matter expert (SME) support,
- emerging systems for which there is little or no information available on the feasibility of embedded training implementation,
- systems that contain many tasks that are cognitive in nature, and
- systems that will use embedded training for both sustainment and acquisition training.

The Navy has few specific directives addressing embedded training, although OPNAV Instruction 1543.XX is now in preparation (Reynolds et al., 1987). This directive, when completed, will serve to guide the acquisition and, to some extent, the design of embedded training, detailed policies are needed to guide the use of embedded training once it is implemented.

A useful set of standards is *Implementing Embedded Training (ET)* (Finley et al., 1988), a series of documents produced by the Army Research Institute (ARI) for the Behavioral and Social Sciences and the Project Manager for Training Devices (PM TRADE). The series consists of ten related documents that present guidance for combat and training systems developers, including Army Materiel Command (AMC) laboratories, Training and Doctrine Command (TRADOC) Combat Developers and Training Developers, and contractor organizations involved in system development or developing technological thrust areas under independent research and development programs. The series of documents includes guidelines and procedures that support the effective consideration, definition, development, and integration of embedded training capabilities for existing and developmental systems. Figure 4.5 lists the ten volumes and gives a brief description of each.

TITLE		DESCRIPTION	
1.	Overview	1.	Overview of ET, the entire product set, and how and when to use what.
2.	ET as a System Alternative	2.	Guidelines for making a decision about whether to continue consideration of ET.
3.	The Role of ET in the Training System Concept	3.	Guidelines for making an early estimation of training system requirements and determining the potential allocation to ET.
4.	Identifying Embedded Training Requirements	4.	Procedures for iteratively determining the embedded training requirement as information becomes progressively available and detailed.
5.	Designing the ET Components	5.	Procedures for iteratively developing the instructional, software, and hardware design of the ET as the system design evolves.
6.	Integrating ET with the Prime System	6.	Guidelines for integrating ET into the design of the system, including both system software and hardware design.
7.	ET Test and Evaluation	7.	Guidance for preparation of test issues for the Test and Evaluation Management Plan and conduct of implant test and technical test/user test.
8.	Incorporating ET in Unit Training	8.	Guidance for developing the user documentation provided to unit personnel and training for unit personnel on ET use.
9.	Logistics Implications	9.	Guidance to assist the definition of support requirements imposed on the system by inclusion of ET.
10.	Integrating ET into Acquisition Documentation	10.	Guidelines on which of the above volumes to use in preparation of specific acquisition documents. Detailed guidance on how to prepare procurement documentation with generic models of a Statement of Work and ET specification.

Fig. 4.5. Embedded training guidelines and procedures documents.

6. IMPLICATIONS FOR TRAINING

Embedded training is a relatively new phenomenon and thus offers both opportunities and obstacles for instructional design professionals. The potential for creating superior learning situations with embedded training exists, but as with any new technology, there are many challenges.

6.1 EMBEDDED TRAINING AS A TRAINING OPTION

According to Gery (1986), the ideal learning situation occurs when:

- An application is personally meaningful and has a payoff (not an artificial classroom exercise that someone else constructed).
- The timing of the learning experience coincides perfectly with the learner's need.
- The learning experience addresses only what is needed and does not cover superfluous options, commands, or information.
- A "coach" permits self-paced learning in a sequence that makes sense to the learner. This coach does not impose on the learner, responds to inquiries, allows exploration without regard to the average learner, is infinitely patient, is able and willing to repeat information whenever and as often as needed, and never makes the learner feel dumb.

Few instructional designers would argue with Gery's concepts, which are well researched and documented in the field of adult learning (Craig, 1987); however, support for these concepts and the ability to construct learning experiences that meet these criteria in the real world are quite different matters. Embedded training offers some exciting opportunities to meet many of these criteria. Embedded training offers a distinct advantage, because learning does not stand alone as an isolated activity but is an ongoing aspect of using a system or product (Mullins, 1989).

The distinction between work and training is becoming fuzzy as embedded training is being built into applications software used at workstations, thus allowing users to move smoothly in and out of training while routinely performing their jobs. From the user's point of view, embedded training is much more convenient than almost any other form of instruction. It is always available when needed and does not require any significant effort to use (Hannum, 1989).

Because learning does not stand alone as an isolated activity but is an ongoing aspect of using a system or product, some experts believe that embedded training promises to become the dominant form of training for software since it fits people's natural learning styles better than other kinds of training (Mullins, 1989). This form of training is likely to increase rapidly in the next few years. Future training will be on-line and on demand (Hannum, 1989).

Embedded training is a significant departure from other training approaches in several important ways. Embedded training is integrated into the context of task performance, unlike stand-alone training that occurs separately from the tasks for which it is designed (MacGregor, 1989). Integration of training into normal operating regimens provides an opportunity to use contextual information as a basis for dynamically structuring training to specific learning needs (Polson and Richardson, 1988; Wenger, 1987). Embedded training is based on a learner-driven problem-solving style of learning rather than a tutorial or didactic instructional approach (Kearsley, 1985). Kearsley states, "What people really need when learning how to use something new is advice or coaching or guidance on how to accomplish their immediate goals and avoid mistakes."

An advantage of embedded training, from a military point of view, is that it standardizes training across the force, regardless of the soldier's geographic location or command (Hardy, 1988). The lesson content and performance standards originate from one source and are a part of the system. According to James Baker (1986):

> When equipment is delivered to a unit, as part of the basis of issue plan (BOIP), technical manuals and test equipment are issued to help maintain the hardware, test packages are issued to maintain the operating software, but nothing is presently issued to maintain the peopleware.

It is his contention that embedded training packages should be part of the BOIP for all automated data systems. The Army seems to be in agreement with Baker's contention. The Army policy is "embedded training is the training technique which must be considered as an initial alternative in new weapons systems. Any type of equipment end item with a built-in computer, not in full time use is a candidate for embedded training." (*Army Research Institute and Project Manager for Training Devices, Broad Agency Announcement*, 1988).

Clearly, embedded training as a training option will be more commonly employed in the future because it makes sense from a learner's viewpoint. As the technologies and methodologies that support embedded training continue to advance, embedded training systems will become more common.

6.2 THE TEAM APPROACH TO SYSTEM DESIGN AND DEVELOPMENT

Computer systems are becoming more powerful as a natural consequence of technological improvements. Users, however, are not faring as well. According to several research studies, users take advantage of only 40 percent of a complex system's capabilities. Users are often unwilling or unable to explore the full powers of a system because they cannot obtain the information they need to do so (Duffy and

Langston, 1985). No matter how advanced the system's capabilities are, if the user is not able to take advantage of the features, the system is not effectively designed.

Because of the integral relationship between a system and its users, it is becoming increasingly more apparent that multidisciplinary teams must be formed to design and develop new computer systems in order to provide a product that achieves optimal usability. Such teams might include project managers, system designers, system developers, subject matter experts, hardware specialists, cognitive psychologists, instructional designers, and technical writers. The team approach represents some challenges as each member of the team brings different priorities to the design and development process. For example, system designers are concerned with system efficiency and may resist adding advanced help systems or embedded training which are educationally sound but inefficient in terms of system design. System developers may be quite concerned about the increases in development time and the strain on resources that must be considered if training is to be embedded into the system. Instructional designers, on the other hand, traditionally have not been as concerned about system efficiency as they are with good training design. Only by combining the expertise of all the team members will the system and training design represent the best possible product for the users.

Instructional designers must continue to develop their skills and competencies in the hardware and software arenas as well as in instructional design. They must build their understanding of training design in terms of both instruction and system planning. Only through an understanding of instructional design combined with an understanding of the system and its capabilities and limitations can the optimal training system be designed.

6.3 EMBEDDED TRAINING DESIGN FOR NEW SYSTEMS

To develop useful help systems, tutorials, or simulations, it is necessary to know how people will use a system/product and what kinds of problems they will encounter. Therefore, the majority of the time required to develop an embedded training program is spent identifying what people need to know in order to use the system. The process of defining these needs involves observation, interviews, studies, analysis, and identification of task requirements. A tremendous amount of detailed information about what people do and know is needed to build a complex help system, simulation, or intelligent tutor (Kearsley, 1985).

However, when a system is new, there is little factual information available, and future embedded training will most likely be developed as part of new systems rather than "strapped on" to existing systems. Kearsley (1985) suggests there are three sources that can be used by instructional designers to predict user behavior as it pertains to developing systems. The first is experience and user data from existing systems or products that somehow resemble the new system or product. In most cases, something exists which resembles the new product enough to allow reasonable inferences. A second source of information is the vast human factors and psychological literature on human performance limitations. After many years of study, there is a great deal of knowledge available about the parameters of cognitive, affective, and psychomotor behavior. In many cases, it is possible to predict accurately how a person will respond to a specific feature or function of a system or product. For example, we know that practice distributed over time works better than a concentrated effort; this concept certainly is applicable to embedded training. The third and most important source of information about user behavior is human factors experiments involving mock-ups or prototypes of the system/product being designed. According to Kearsley (1985):

> One legitimate problem associated with the development of embedded training for new systems is that it will undergo much change as the product or system is redefined and refined. This often seems wasteful in so far as waiting for the final version of the product would have saved a lot of development. However, as the usability of the system/product needs to be tested and modified, so does the usability of the embedded training components. Thus, the effort and expense involved is a necessary and unavoidable aspect of the system/product.

6.4 MEETING USERS' NEEDS

Computer users are no longer exclusively technical personnel but can be anybody from clerks to chief executive officers. Computer interaction has moved from a monologue to more of a dialogue; people want a high degree of control over programs and machines (Kearsley, 1985).

Embedded training provides the new system user with hands-on experience; as the user gains skill and confidence, proficiency on the system increases significantly (Hannum, 1989). Early, successful hands-on experience can be pivotal in terms of producing the attitudes which enable a naive trainee to become proficient at operating a system; the system is seen as approachable.

However, a user does not remain at an entry level for long. Embedded training systems must also enable the user to progress systematically through multiple levels of training objectives. This feature makes it possible for any student, regardless of entry level skills and knowledge, to enter the system at the appropriate level, receive needed training, and progress to the next level when ready. Thus, one goal of embedded training is to provide a method of dealing effectively with training problems at any level (Baker, 1986).

The need for sustainment or refresher training must also be considered in the training design. Embedded training should not only be used as an initial training vehicle but should serve to ensure the maintenance of proficiency (Baker, 1986; Hannum, 1989).

Even the expert has information needs that should not be ignored in the training design. However, experts need a very different type of information than novice and intermediate users. Advanced help systems are necessary to provide quick, efficient references for the expert user.

All of these factors make the training design complex. Instructional designers are challenged to meet the multiple needs of users as they progress in working with the system.

6.5 EMBEDDED TRAINING AND ITS IMPLICATIONS FOR INSTRUCTIONAL SYSTEMS DESIGN (ISD)

There is little indication that any of the embedded training systems currently operational in the military are derived from a thoroughly worked out set of training requirements. Likewise, there is no indication that any of the embedded training systems now in operation were designed in the context of a total training system (Warm et al., 1988). Because the early attempts at embedded training were driven by experts from the fields of psychology, engineering, and computer science (Kearsley, 1985), ISD was not followed in the development of the majority of existing embedded training systems.

There are some problems with using the ISD for embedded training. It is known, for example, that ISD does not work well for ill-defined systems, where ISD steps are difficult, if not impossible, to follow using existing information sources (Kearsley, 1985). Embedded training also blurs the distinction between training and performance. ISD may need to be expanded to analyze a broader range of performance issues not directly related to training, such as the accuracy of job descriptions and employee motivation factors, prior to the training design.

It is clear that a comprehensive approach to training and system design is needed in the future if instructional designers are to be able to effectively meet the needs of system users. Because many embedded training systems currently in operation have been essentially handcrafted, there are few systematic design principles to follow. The lack of design guidelines slows the development process, since each design team must start from basic principles. Perhaps, a combination of systems engineering design and ISD will provide the right combination of strategies for embedded training design.

7. IMPLICATIONS FOR FURTHER RESEARCH

Embedded training is an emerging approach to training that needs further refinement. There are relatively few controlled experiments investigating whether or not embedded training is superior to, or at least equal to, other forms of training, such as stand-alone computer-based instruction. A limited number of studies have been performed on user behavior with existing embedded systems (Stoddard, 1985). Several operational projects are using embedded training, particularly in the military services. Further research is needed to determine the superiority of embedded versus concurrent training, not only from a training point of view but also in terms of system design and development efficiency and maintainability. Overall, the areas of computer hardware/software, instructional design, and their relationship to embedded training need more study.

7.1 COMPUTER RELATED RESEARCH

Research is needed to devise methods and software that allow designers to create and store graphics more efficiently. Graphics tend to be "memory hogs", slowing response time and decreasing system efficiency. The process of incorporating graphics into embedded training lessons indicates the need for further research.

The feasibility of developing authoring systems to assist in the creation of embedded training must also be explored. Work is being done to develop concurrent authoring systems which run in tandem with applications software. Research into the effectiveness of these authoring systems and the relative advantages and disadvantages of one system over another should be noted. A related factor is the feasibility of developing screen templates to facilitate the development of embedded training.

Additional research is necessary to determine the minimum hardware and software environment required for developing and delivering embedded training. Data on the differences between mechanical and optical disks for storage should aid in improving embedded training design. The ability of embedded training to capture mistakes made by users is a major consideration, since that data may be used to provided prescriptive feedback to the user and to evaluate system efficiency.

7.2 INSTRUCTION RELATED RESEARCH

Additional research should be conducted concerning embedded training from an instructional design viewpoint. One issue, in particular, relates to how and when the embedded training is activated. Areas which need more study are: (1) determining whether embedded training segments should be requested by the user, or (2) if the embedded training application should diagnose user errors on the live system and provide tutoring related to those errors.

Another embedded training issue requiring research relates to the substance or content of the embedded training. An overriding question is "What constitutes effective embedded training?" One issue that warrants consideration is the relative effectiveness of the different types of embedded training (help systems, tutorials, simulations), particularly in terms of the content being taught. The types and levels of interaction that are most effective in particular types of content mastery are also relevant research topics.

Navigation issues require further study. The question of whether embedded training should overlay an applications screen or replace it, warrants some study. It is important to determine the best mix between learner and system control of instruction. Further research into the relative advantages of menu driven versus context-sensitive help merits consideration. More information is needed on the effectiveness of linear training (where the user completes predetermined training segments) versus nonlinear training (where the user is free to skip about within the material). It will also be useful to know if it is more beneficial for the learner or the system to determine when mastery is achieved and when the learner should advance within the instructional system.

The construction of segments of embedded instruction in terms of their relative effectiveness in producing the desired learning outcome is another area requiring research. The relative effectiveness of displaying procedural rules, displaying examples, using practice exercises and feedback, and determining the amount of background (declarative) information needed are relevant issues. The usefulness of different types of graphics such as charts, tables, and flow diagrams also merit consideration by educational researchers.

These items represent a sampling of some of the relevant research issues related to embedded training. Since this training methodology is quite new, there are more questions than answers at this point. The challenge to educational researchers is to answer some of these questions so that embedded training can be designed for maximum efficiency and effectiveness.

8. SUMMARY

The use of embedded training as a training technology will increase rapidly in the next few years because of the many advantages embedded training offers. From the user's point of view, embedded training provides on-the-job training relevant to the task at hand, is always available when needed, and permits the user to learn at his own pace and in a sequence that makes sense to that user. From a system efficiency standpoint, embedded training can optimize the system usage by providing relevant, timely information when and where it is needed. If properly designed, the data collected from embedded training can be used to judge system effectiveness and efficiency and provide data for future enhancements.

The effectiveness of embedded training, however, depends on the effectiveness of the instructional design strategies used in the training design. The decision to embed training, the most effective mix of types of training (help, tutors, simulation, and intelligent tutors), and levels of training are the keys to the success or failure of embedded training. The training design must also consider the needs of novice, intermediate, and expert users, as well as both initial and continuing training needs.

A team approach to system design, where instructional designers work as a part of the system design and development process, is necessary if training is to be embedded into a new computer system. Only through the team approach can the needs of the end user be systematically represented in the system design.

Embedded training is a recent phenomenon that raises many questions about how it can be created effectively and efficiently. Some questions can be answered from related research; other questions remain unanswered and will have to wait until more research has been conducted. Development of authoring tools specifically designed for embedded training and advances in the state of the art in artificial intelligence and expert systems applications will further the field of embedded training. There seems little doubt that embedded training is a powerful training approach that will soon come of age.

9. REFERENCES

Andersen, A., *Embedded Training, The Explosive Growth Area in CBT*, Workshop Presentation to NSPI by Arthur Andersen and Company, 1986.

Army Construction Engineering Research Laboratory (USA-CERL). *Intelligent Embedded Instruction for Computer Aided Design (CAD) Systems*, USA-CERL, 1988.

Army Research Institute and Project Manager for Training Devices, Broad Agency Announcement, Army Research Institute for the Behavioral and Social Sciences, Arlington, VA, January 1988.

Baker, J. D., *A Brief Overview of the Concept of Embedded Training.* Presentation to the Joint Human Factors Society and Washington Academy of Science, May 1986.

Baker, J. D., *Embedded Training: A Brief History*, U.S. Army Research Institute, Alexandria, VA, 1980.

Bentley, T., "Embedding CBT Down Under," *CBT Directions*, June 1989.

Craig, R., *Training and Development Handbook*, McGraw Hill, 1987.

Duffy, T. M. and Langston, M. D., *On-Line Help: Design Issues for Authoring Systems*, Communication Design Center, Carnegie-Mellon University, Report Number 18, November 1985.

Evans, D., Adams, J., Simkins, M. L., Aldrich, R., Dyer, F., and Narva, M., *Lessons Learned from ET Design Process for ASAS/ENSCE*, U.S. Army Research Institute for Behavioral and Social Sciences, ARI Report Number 88-87, September 1988.

Finley, D., et al., *Implementing Embedded Training (ET)*: Volumes 1-10. U.S. Army Research Institute, March 1988.

Volume 1: Overview
Volume 2: Embedded Training as a System Alternative
Volume 3: The Role of Embedded Training in the Training System Concept
Volume 4: Identifying Embedded Training Requirements
Volume 5: Designing the Embedded Training Components
Volume 6: Integrating Embedded Training with the Prime System
Volume 7: Embedded Training Test and Evaluation
Volume 8: Incorporating Embedded Training in Unit Training

Volume 9: Logistics Implications
Volume 10: Integrating Embedded Training into Acquisition Documentation

Fitzpatrick, J. A., Warm, R. E., Ditzian, J. L., and Roth, J. T., *Interim Procedures for Embedded Training (ET) Component Design*, Army Research Institute, Arlington, VA, 1987.

Germas, J. and Baker, J., *Utilization of Tactical Computers to Train Tactical Computer Operators*, U. S. Army Research Institute, Arlington, VA, ARI Report Number 452, July 1980.

Gery, G., *Making CBT Happen*. Weingarten Publications, Boston, MA, 1987.

Gery, G., "The On-Line Coach," *Data Training*, May 1986.

Gralla, P., "Embedded AI: The Look of Things to Come?" *PC Week*, April 19, 1988.

Hannum, W., *Embedded Training Systems*, University of North Carolina, Chapel Hill, NC, (Draft - October 1989).

Hardy, H. C., "Armor Training-1997 - An Application of Embedded Training," *Armor*, November-December 1988.

Hart, F., "The Future of Computers in Industrial Training," *Journal of Instructional Development*, Vol. 6, No. 2, Winter 1983.

Hayes, J. and Sherrard, J., *A Computer Aided Instruction Tutorial for the Ramtek 9400 Color Graphics Display System at the Naval Postgraduate School*, Monterey, CA, Masters Thesis, 1981.

Hill, R., *Fog-M Embedded Training Demonstration Model*, Technical Report, Prepared for Guidance and Control Directorate, Research, Development, and Engineering Center, U.S. Army Missile Command, Redstone Arsenal, AL, 1986.

Johnson, W. B., *Development of a Tool for Authoring Intelligent Tutoring Systems.* Paper Prepared for Technology and Innovations in Training and Education Conference, March 1989.

Kearsley, G., "Embedded Training: The New Look of Computer-Based Instruction," *Machine-Mediated Learning*, Volume 1, Number 3, 1985.

MacGregor, D., *Computer Assisted Training of Telephone Interviewers: Application of an Intelligent Tutoring System Model*, A Research Proposal, MacGregor - Bates, Inc., Eugene, Oregon, 1989.

Massey, D., Harris, M., Downes-Martin, S., and Kurland, L., *Embedded Training Technical Survey*. BBN Lab., Inc., Cambridge, MA, May 1986.

Mullins, M., "Embedded Training: A Bibliography," *Technical Communications*, First Quarter, 1989.

Polson, M. C. and Richardson J. L., *Foundations of Intelligent Tutoring Systems*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1988.

Purifoy, G. R. Jr., Chenzoff, A. P., Harris, C. B., Roth, J. T., and Strasel, H. C., *Fog-M System Task and Training Requirements Analysis for Embedded Training (ET)*, Applied Science Associates, Inc., Valencia, PA, 1985.

Reynolds, R. et al., *Lessons Learned from Currently Fielded Navy Embedded Training Systems*, Naval Training Systems Center (NTSC), Orlando, FL, 1987.

Roth, J. T., Sullivan, G. K., and Ditzian, J. L., *A Procedure for Developing Embedded Training Requirements*, Army Research Institute, Arlington, VA, 1986.

Sautter, B., "Not Just for Training Anymore," *Data Training*, July 1987.

Stoddard, M. L., *Learning Styles and Embedded Training: A Case Study*, Los Alamos National Lab, New Mexico, 1985.

Strasel, H. C., Dyer, Frederick N., et al., *Review of Eight Army Systems: Characteristics and Implications for Embedded Training*, U. S. Army Research Institute for the Behavioral and Social Sciences, ARI Report Number 88-14, April 1988.

Warm, R. E., Roth, J. T., Sullivan, G. K., and Bogner, S. M., *Tri-Service Review of Existing System Embedded Training (ET) Components*, Army Research Institute Report Number 881219019, November 1988.

Webr, J. "Training 101...Instructor-Led or Computer Based: Which Will Work Best for You?" *Training and Development Journal*, June 1988.

Wenger, E., *Artificial Intelligence and Tutoring Systems*, Morgan Kaufman Publishers, Inc., 1987.

Section 5

INSTRUCTIONAL SYSTEMS DEVELOPMENT: THE IMPACT OF COMPUTER TECHNOLOGY

TABLE OF CONTENTS

INSTRUCTIONAL SYSTEMS DEVELOPMENT: THE IMPACT OF COMPUTER TECHNOLOGY

1. INTRODUCTION

In the past 20 years, the basic Instructional Systems Development (ISD) model has influenced the design and delivery of instruction in military, business, and industrial settings. Contributions from cognitive psychology and information technologies have expanded possibilities for the design and delivery of instruction, and are currently reshaping some of the fundamental ISD concepts. This section presents a discussion of current applications of instructional systems development. An overview of developments in cognitive psychology, embedded training, artificial intelligence systems and hypertext/hypermedia and their impact on ISD is presented. Finally, research implications of ISD are discussed in relationship to information processing technologies.

1.1 INSTRUCTIONAL SYSTEMS DEVELOPMENT

ISD provides a conceptual framework for producing effective and efficient instruction. The ISD approach is used for the development of training, in any medium, by providing steps for instructional designers to follow in producing instruction. Several ISD models have been developed from the fields of education, communication, psychology, and systems engineering. Features of the ISD approach, which distinguish it from traditional development, are the instructional analysis, establishment of criteria for the final product, and use of formative evaluation and revision techniques.

The systematic approach utilized by the ISD model allows instructional systems developers to divide problems into manageable parts that function together for the resolution of the original problem. The systems approach assists the developer in clearly defining a problem so that possible alternative solutions can be identified. Second, the process helps in the selection of the best alternatives and develops the most appropriate mix of solutions. Third, the best solutions can then be implemented and evaluated for overall effectiveness. Finally, the systems approach allows revision of the solutions, if necessary.

Application of the ISD model can be useful in developing instruction on topics as diverse as learning to play the piano, operating a machine, speaking a foreign language, and managing computer hardware and software. The ISD process includes a series of interrelated steps that are based on principles derived from educational research and theory. The systems framework is predicated on the concept of a continuing cycle of development with evaluation directly related to objectives of the training process. If an instructional product has defects or a newly designed course falls short of meeting its objectives, decisions may be made to revise the instruction to overcome the implementation shortcomings. Both summative and formative techniques are utilized in evaluating the learning materials. Formative evaluation occurs throughout the instructional process, whereas summative techniques are utilized at the end of an instructional period. The result of an instructional process should enable learners to demonstrate that they have learned new information and/or how to perform a new procedure.

Although a variety of ISD models have appeared in the literature, many of their components are common to most instructional systems design models. Andrews and Goodson's (1980) study analyzes over forty instructional development models that exhibit the following similarities:

- conducting a needs assessment,
- outlining the description of a job or task,
- developing performance objectives,
- developing assessment items that test the learner,
- developing the instruction,
- evaluating the instruction,
- implementing the course or program of instruction, and
- revising the instruction.

Several of the more commonly accepted ISD models were developed by Branson, Briggs and Wager, and Dick and Carey (McCombs, 1986).

Generally the instructional design process combines the components described above into five phases to produce instructional materials which are based on the subject matter, knowledge level and abilities of the learner, learning theories, and technological considerations. The five phases include: analysis, design, development, implementation, and evaluation. The first phase, analysis, refers to finding out what individuals need to know, what they need to do, which of these requirements necessitate training, and how to measure the individual's performance. A thorough instructional analysis includes the identification of goals and related concepts and information the learner needs to achieve a specific task or to perform a procedure. In addition, the analysis enables the designer to create a blueprint for applying a systems approach to instructional design (Branson and Grow, 1987). Phase two, design, includes writing the specifications for how instruction is to be accomplished. Phase three involves the process of developing and following the specifications resulting from phase two, putting together the materials and developing appropriate sequences required to conduct the instruction. Implementation, the fourth phase, is the process of putting an instructional product into place using the instructional model. Phase five, evaluation, refers to determining the overall effectiveness of the instructional material and process. Evaluation of the instruction is ongoing, as well as summative, with revisions to the plan or the product possible at any time (Briggs, 1979).

1.2 IMPACT OF TECHNOLOGY ON INSTRUCTIONAL DESIGN

The Interservice Procedures for the Instructional Systems Development (IPISD) model was developed for the military in an effort to improve interservice training for the Army, Navy, Air Force, and Marines and has been used extensively in these contexts since 1975. Features of this model include the specification of basic inputs, processes, outputs, their interfaces, and feedback control processes. Unfortunately, the interservice model was identified with other unsuccessful self-pacing and individualized learning models and thus suffered a short setback in 1980 (Vineberg and Joyner, 1980). Despite this brief period, the interservice model has continued to influence the development of training for business, education, industrial settings, and the military.

The continuing success of the ISD models in universities and industrial settings is attributed to the close adherence to the structure of the selected model. Following the steps outlined in the various models permits instructional designers to approach systematically the process of design, development, implementation, and evaluation in achieving effective and efficient instruction.

The most widely applied instructional design theory is based largely on the work of Robert Gagne. Even though this model or similar models have been firmly in place for 20 years, a number of limitations are now apparent in the behavioral based systems. The resulting instructional systems are often passive rather than interactive with minimal involvement of the learner beyond being a receptor of information. Most of the first generation ISD models are based on the psychology of the 1950s and 1960s, emphasizing the analytical rather than synthesis of learning. Since these models predate the development of highly interactive delivery systems, a new generation of design methodology is necessary for developing instruction for these systems. Because the systems are new and have much interactive potential, current ISD development practices are extremely labor intensive, especially in developing courseware (Merrill, Zhongmin, and Jones, 1990).

The new technology has established a need for a new generation of instructional design methodology for interactive computer based instructional materials. This new generation organizes instructional design knowledge and defines a methodology for performing the design. An on-line intelligent advisor program to customize instructional delivery and a series of intelligent computer-based design tools for knowledge analysis are characteristics of the new generation of instructional design. (Merrill, Zhongmin, and Jones, 1990).

Currently, the components of the ISD models are being revamped to include principles from cognitive psychology. Recent advancements in artificial intelligence, hypertext/hypermedia, embedded training, and optical disc technology are forcing designers to re-evaluate old theories and principles for designing instruction. The capabilities of technology have already begun to shape both the expectations and the relationship between users and technology in the instructional process.

2. ISSUES RELATED TO THE APPLICATION OF INSTRUCTIONAL SYSTEMS

Problems associated with the development and implementation of the ISD model fall into three primary areas that are directly related to the analysis phase (McCombs, 1986):

1. Lack of detailed, instructionally sound analysis and design procedures is attributed to a lack of knowledge on the part of the instructional designers in both the ISD process and the underlying theories of the model (Andrews and Goodson, 1980).

2. Inadequate knowledge and experience on the part of those applying the procedures are problems that are particularly apparent in the most critical phase of analysis (Dick and Carey, 1978). Designers must analyze instructional requirements which include how knowledge is organized in the learners' minds as well as the role of knowledge at various stages of learning. Learning environments and social contexts are also critical factors in the analysis of the problem.

3. "Overproceduralization" of complex ISD steps and processes to the point of distorting and misrepresenting the overall learning task is also a primary area of concern. ISD is more than a series of sequential procedures for solving instructional problems.

The analysis phase is probably the most important phase in the design process due to its overall impact on the end result. In this first phase procedural and hierarchical issues related to the task being developed are determined. A procedural approach is used when the task to be taught is a series of behaviors that must be performed in sequence. Hierarchical identification is used to determine prerequisite, subordinate skills which must be achieved prior to learning the new task or achieving the defined goal. A combination of the procedural and hierarchical processes is used in many situations that include complex psychomotor skills and specific cognitive tasks. For example, learning how to use the computer with new software requires both procedural and hierarchical processes. Extreme care must be taken during the analysis to ensure a direct relationship between what the student needs to know and what will be taught. Thus, on-site job analysis techniques are absolutely critical in designing effective instructional systems.

Learning theorists agree that the basic ISD model is appropriate for teaching behavior or a mechanical process such as changing a tire, but falls short on complex subject matter involving decision making or problem solving. Several key elements of instructional systems design that are often not considered but are nevertheless critical include learning theory, creativity, and motivation. Many ISD models do not apply enough learning theory, and minimal efforts seem to have been made to include creative and motivational factors in learning. All ISD models require that their procedures or steps be based on current theory and research, logic, experience, and frequent reviews (Andrews and Goodson, 1980; Briggs, 1979; Dick and Carey, 1978; McCombs, 1986; Merrill, 1989).

In the next ten years it is expected that certain forces will influence a number of changes in the ISD model. Although the general steps of the ISD model have remained stable for the past 20 years, the specific approaches, methods, and strategies applied within each step are evolving. New theories of learning will not only deal with external behaviors but will begin to broaden the focus to include learners' internal perceptions and their influence on cognition, motivation, and creativity.

In addition, environmental elements will be considered more carefully in the design process. Environmental elements affecting job performance include the organization, work, personal attitude, laws, and practices. In human performance each person produces outputs leading to positive or negative consequences. If instructional designers fail to consider the impact of environmental influences on the consequences of behavior and learning, ineffective training and inappropriate interventions may be developed (Galagan, 1987).

Thus, the ISD process will continue to utilize and conduct even more thorough examinations of the total human performance system (how people learn and how environments affect learning) to decide what problems exist and the best solutions for these problems. Problems will need to be defined more clearly and precisely to develop appropriate training. The total working environment and its effect on people and their job performances will become an integral consideration in human performance systems.

To meet the increase in acknowledged factors which affect the learning process, today's instructional designers must acquire a more diverse set of skills than the designers of 20 years ago. In addition, designers must use more resources and more sophisticated technology to design instruction. Computer technology and software have evolved so rapidly that designers must constantly update their computer skills. For example, alternative adaptive rather than linear models are now available for developing individualized learning systems (Hooper and Hannafin, 1988). Designers must be aware of trends and resources in order to continue developing quality learning programs.

Instructional designers will have options for combining ISD models and quality computer software to design more sophisticated instruction. Attempts have been made to develop computer-aided design of instructional systems and products. Several applications are presently available which allow designers to automate portions of the ISD process, making the model easier to follow and more capable of improving the efficiency and effectiveness of instruction. The Naval Training Equipment Center has attempted to automate ISD and training programs (Branson and Grow, 1987). There is every reason to believe that these efforts to automate the ISD process will continue and will become increasingly successful.

3. THE EVOLUTION TOWARDS INTELLIGENT INSTRUCTIONAL SYSTEMS

Recent advances in the new technologies, cognitive psychology, embedded training, artificial intelligence systems and hypertext/hypermedia, provide more sophisticated alternatives for ISD. Computer systems, combined with recent developments in learning theory, enable instructional designers to explore the efficacy of nonlinear and intelligent learning models. Intelligent systems are now feasible due to computer technological advances in memory, processors, and most importantly, the availability and cost effectiveness of hardware and software.

3.1 COGNITIVE PSYCHOLOGY

Historically, behavioral psychology has dominated the training environment in large corporations and institutions. Behavioral techniques have been effective in developing task analyses which delineate a sequence of actions for describing job requirements. However, when training for more complex tasks, the conventional behavioral techniques are not adequate for determining what a subject expert knows or for communicating the expert's knowledge to new performers. Advances in cognitive psychology have been determined to be more effective for managing these issues (Harmon and King, 1985).

Buehner (1987) points out that cognitive psychologists have contributed to the knowledge of how humans learn, specifically in the areas of learning theory, memory, and information processing. Theories regarding how individuals apply unique patterns for processing information have resulted from research in cognitive psychology. These patterns include an individual's normal mode of thinking, problem solving, perceiving, and remembering. An individual's cognitive style is thought to include all processes used in information processing: perception, thought, memory, imaging, and problem solving. It is believed that if cognitive styles are considered when planning instructional design, the individual's ability to learn the desired task or achieve the desired outcome will be enhanced.

Consistent principles of learning should be incorporated into instructional design and development. Some of the most commonly accepted learning principles include:

- We learn by instruction and/or by images of doing and by observing others doing (Bandura, 1987; Gagne, 1965).
- Without reinforcement there can be no learning (Skinner,1968).
- Overt practice improves learning and retention (Thorndike, 1921).

These principles, along with others, are being modified towards a cognitive orientation as a result of thorough prescriptive research.

Research also has suggested possible presentation formats for computer-based instruction. The use of both verbal and non-verbal information in screen displays allows learners to complete tasks more quickly and results in increased retention of information. Learning is enhanced when non-verbal information is presented in the center of the visual field and above any text that may be present on the screen. In addition, text should be presented in double spaced format to facilitate reading. These research findings are particularly useful for adapting to various cognitive styles of learning which result in enhanced instructional efficiency, especially in the presentation of computer-based instructional messages that include both visual and verbal information (Buehner, 1987).

Research is now being conducted to determine differences between experts and novices in a variety of content domains. More research is focusing on the transition from novice to expert. By analyzing the knowledge differences of experts and novices, certain contributions can be made to the development of appropriate content for training programs. The emphasis on looking at the cognitive as well as the behavioral aspects of learning will help to bridge the gap between the novice and the expert model. The study of the experts' knowledge and data structures enables the designer to incorporate this information into the training program. Cognitive oriented models will increasingly replace the behavior oriented models, eventually impacting procedures for task analysis, planning of the instructional strategy, and, finally, the overall design of the lesson.

The changes are likely to broaden the overall efficacy of the ISD process. Hannum and Hansen (1989) predict cognitive ISD models will be developed which are more in keeping with technological developments and other factors influencing the ISD models. Entirely new models ultimately may be designed with a more cognitive orientation to procedures related to task analysis, assessment, planning of instructional strategies, and lesson design. Finally, interactive models that enable the learner to move forward, backward and then to branch to related concepts will replace linear models.

Changes in the work place are being affected by recent advancements in technology. Tasks that once required a hands-on approach have now become more cognitive oriented tasks. The assembly line in automobile manufacturing plants is one example of how industry has incorporated technology into the completion of certain tasks. Robots have replaced workers who screwed in bolts and tightened nuts. The skills necessary for completing tasks in the automobile industry have shifted toward the more cognitive skills of monitoring and decision making in the utilization of technology.

3.2 EMBEDDED TRAINING

Embedded training is structured to provide immediate, individualized, on-line access to a full range of information, software, guidance, advice, and assistance. Generally, it provides data, images, tools, assessment, and monitoring systems to permit the user to perform a job with a minimum of support and intervention by others. Thus, within a group of users, no two people may complete the same instructional sequence or see the same examples because the systems will be adjusted to their individual differences (Hannum and Hansen, 1989). Ideally, users can access the various levels of embedded training according to their particular needs. Some of the more sophisticated, intelligent embedded training systems analyze the user's mistakes and automatically access the level of help or instruction necessary to solve a specific problem.

The military has been in the forefront of embedded training by incorporating it into computer-based systems ranging from sophisticated weapons to automated record keeping processes. A review of various types of embedded training for military use revealed that all of the users not only accepted it but perceived benefits from using this type of training (Applied Science Associates, et al., 1988).

Since training is being incorporated into applications software used at workstations, workers can move quickly in and out of training while routinely performing their jobs. On-line and on demand applications training is likely to increase rapidly in the next few years (Hannum and Hansen, 1989). Embedded training is a significant departure from other training approaches, since it is integrated into the context of real time performance. Integration of training into normal operating systems provides an opportunity to use contextual information as a basis for dynamically structuring training to specific learning needs.

3.3 ARTIFICIAL INTELLIGENCE SYSTEMS

Another major development in ISD is artificial intelligence systems. Artificial intelligence (AI) is basically allowing machines to emulate human intelligence (Jonassen, 1988). The challenge of AI research is focused on developing intelligent machines that synthesize knowledge automatically, generate new knowledge, and add it back to the system. Nearly every major government institution has invested in the use of AI technology for problem solving, decision making, maintenance of complex systems, and hazardous operations (Schoen and Sykes, 1987).

Three major areas of AI research and application include robotics, natural language processing, and expert systems. Robotics are used extensively to perform repetitive and/or dangerous tasks in manufacturing. Natural language processing focuses on developing the technology for computers to respond to human language. Expert systems are designed to diagnose problems and make appropriate decisions. These systems not only perform their tasks but learn new information through solving different problems (Harmon and King, 1985; Harvey, 1988).

Building an expert system requires using the techniques developed by cognitive psychologists (Harmon and King, 1985). The task of the AI researcher is to capture the knowledge of the human expert so it can be encoded into the rules for the knowledge base (Johnson, 1988). Expert systems can then be used to perform the same tasks performed by human experts.

It is believed that the introduction of expert systems eventually will have a dramatic impact on traditional training. Complex skills and knowledge will be easier to teach as a consequence of being accessible through expert systems (Dede, 1986). Trainers will learn to apply concepts that have been developed by knowledge engineers to address training problems, develop detailed job descriptions, and analyze and design training programs (Harmon and King, 1985). The expert system has advantages that enable people to solve problems through flexible, efficient, and effective processes, and AI research will continue to focus on the development of methods for representing and manipulating information (Garnham, 1988).

3.4 HYPERTEXT/HYPERMEDIA

Other currently promising technological advances in ISD include hypertext/hypermedia. Computer and software technology that enables the user to move in a nonlinear manner is called hypertext and/or hypermedia. The user is able to move to various parts of a database through cross-links called "nodes." This cross-referencing capability enables the user to access information quickly without having to move in a sequential manner. Shneiderman and Kearsley (1989) state, "The most common meaning of hypertext is a database that has active cross-references and allows one to 'jump' to other parts of the database as desired." Hypertext is composed of nodes containing text, and hypermedia is composed of nodes containing graphics, pictures, video, and other types of media.

Jonassen (1988) contends that hypertext and hypermedia can increase meaningful links between existing knowledge and new knowledge, thus increasing comprehension and promoting ease of learning. Building on the premise that learning is the reorganization of the learner's existing knowledge, hypertext has promise for mapping subject matter knowledge onto the learner's existing knowledge structure. Since hypertext systems have a brief history, there is little evidence available to demonstrate their effectiveness. However, existing hypertext research suggests that learners may be able to significantly deepen their understanding of material taught through the use of hypertext (Shneiderman and Kearsley, 1989). Clearly, hypertext systems offer distinct advantages for finding facts, browsing, and acquiring information (Marchionini and Shneiderman, 1988). Hypertext concept studies have convincingly demonstrated that instruction can be effectively designed, learning is enhanced, and advanced learners do benefit from the medium (Hannum and Hansen, 1989).

Research areas for hypertext/hypermedia include a different way of analyzing structure of knowledge issues and design principles related to this unique form of presenting information. Interactive computing takes on additional, and more effective, dimensions with sights, sounds, and graphics available to augment the learning (Kearsley, 1988). Since learners can enter and leave documents at arbitrary points in their training, designers need to determine the most effective organization of knowledge and tracking schemes for these learning systems. The design capabilities center on the increased flexibility in structure and style for developing complex learning tasks. The limitations of hypertext appear to exist only in the designer's mind, since the increased access to memory and new technologies appears almost limitless.

4. RESEARCH IMPLICATIONS OF ISD

Although ISD models have been used for the past 20 years, many research issues remain in the area of design and development. With the advances in information technologies, research concerning the impact on future ISD models is also needed.

4.1 RESEARCH IN INSTRUCTIONAL DESIGN AND DEVELOPMENT

A distinction should be made between instructional design and instructional development as they apply to research issues. Design research refers to the theory of instruction and training principles, including testing generalizations related to achievement and motivation. Development research deals with the issues of implementation, including decisions on presentation medium or budgetary constraints (Reigeluth, 1983).

Research literature suggests that the development of computer technology for instructional purposes is continuing at an accelerated pace. Thus, today's computer technology outdistances research in instruction, leaving instructional designers in the position of determining ways to employ technology without the benefit of research. Research on instruction continues to be stimulated by technology that is economical and appears suitable for learning. In addition, instructional research continues to be defined and shaped by the context in which instruction occurs or will occur in the future (Gagne, 1986).

Educational practitioners, government, and industrial decision makers seem to be more interested in research results that support privately or publicly held beliefs (Kerlinger, 1977). Descriptive research reports have been used extensively, but policy makers seem to be unimpressed with the results of this research. Sufficient evidence exists to indicate that, in media research, descriptive, counter-intuitive findings are often ignored even when supported over a long period of time. This condition puzzles researchers who are trying to advance instructional technology through their findings, leading many to conclude that researchers need to go beyond descriptive research methods and incorporate more prescriptive methodology in future research to determine more convincing results for the users (Clark, 1983).

Although a number of models exist in the area of design and development, very few researchers have compared one model or theory to another (Reigeluth, 1983). In order to make more conclusive progress and narrow the focus of research, it is believed that theories of ISD must be reduced to a few that explain the same phenomenon. For example, nearly all learning tasks can be taught procedurally (Merrill, 1983) and/or declaratively (Rumelhart and Norman, 1981). Procedural learning emphasizes the sequence of steps in order to attain a goal, whereas declarative knowledge deals with concepts and principles related to the learning. Both approaches are essential in designing training which incorporates the new technologies, such as embedded training or hypertext. More information is needed to definitively conclude which tasks are best taught by which process.

Other relevant research findings in the area of ISD have illustrated that the more successful, self-paced instructional units used in the Air Force are those that closely adhere to the ISD process. Successful implementation of training is more apt to occur with strong management support, a balance of media and printed materials, individual and group activities, high instructor dedication, and adequate instructional resources (Back and McCombs, 1984). Another study by Branson and Grow (1987) concludes that proper training in the ISD principles, maintenance of the team effort, a realistic management plan, and the necessary financial, human, and physical resources are necessary for successful implementation of military training. Oxford-Carpenter and Schultz-Skiner (1984) discuss the importance of successful implementation as characterized by courses that followed the ISD model along with thorough documentation of the process to facilitate communication and accountability.

The results of a 13-year study for successful implementation of ISD models conclude that necessary financial support in human and other resources is critical in the development process. The models must be adapted to needs that change during the project through effective management practices. Finally, it is noted that the user must be adequately trained to use the model's products (Stiehl and Streit, 1984). Briggs (1982) concludes that successful implementation is gained through a thorough analysis and needs assessment to adequately define goals and objectives. Briggs argues that a positive correlation exists between specific agreement of the goals and objectives and successful implementation.

Two important considerations for future research in instructional design and development to help solve current problems are: (1) a greater commitment by researchers to review the existing research in the social science disciplines of psychology, sociology, anthropology, and political science and (2) differentiation between instructional design and instructional development research for the application of prescriptive research approaches. Focusing on these two areas promises to have a greater impact on decision makers in education and training than it has in the past (Clark, 1989).

Other research concerns have focused on individual learning styles as a basis for designing instruction. Learning theorists are suggesting that computer software developers, as well as instructional designers, pay more attention to the learner's preferred way of assimilating and mastering learning. Not every learner's approach to learning is the same. The literature suggests that some individuals in a learning situation are likely to be either deductive and/or inductive depending on their comfort level with the particular task or the outcome they are trying to achieve. Individuals who design software tend to be deductive thinkers, thus organizing the learning process and the software manuals in a deductive manner (Galagan, 1987). Accordingly, the big challenge for learning theorists and instructional designers is to understand the technology and learning process in order to design instructional materials for every type of learner or learning style (Galagan, 1987).

One final significant trend in the design of instruction includes the mechanics of interacting with computer hardware and software. In the past, users of computers have had to memorize a long list of control characters or other commands in order to interact with the program. The most recent software advances require the user to have little or no knowledge of computers. All the user has to do is point or touch the screen to complete functions (Merrill, 1988). The next generation of computer technology will enable the user to complete tasks with the human voice instead of machine commands and programming languages (Merrill, 1988).

4.2 FUTURE RESEARCH

Because of the time, expertise, hardware, and other expense involved, designing instruction which incorporates artificial intelligence, hypertext/hypermedia, and/or embedded training technologies often require considerably more planning and development than traditional classroom instruction. Ely (1987) provides a qualifying perspective on the future of ISD. He indicates that it is the instructional system's design and the utilization of hardware and software which determine the effectiveness of instruction and learning. Technology is merely the vehicle to deliver effective instruction. Clark (1989) further elaborates on the role of technology by stating, "Five decades of research suggest that there are no learning benefits to be gained from employing different technology in instruction, regardless of their obviously attractive features or advertised superiority. The best current evidence is that media are mere vehicles that deliver instruction but do not influence learners' achievement any more than the truck that delivers our groceries causes changes in our nutrition."

The results from using earlier ISD models in industry, business, and the military were very promising in the 1970s and there is reason to believe that increasing the quality of instruction through innovative technology by incorporating empirical research will certainly have an impact on improving future ISD models (McCombs, 1986).

There are many areas to be researched concerning the implications of ISD and other learning-related areas. Several areas that need research include (Hannum and Hansen, 1989):

- determining individuals' learning differences,
- determining how individuals best learn in specific situations,
- operating ISD models more efficiently,
- developing ISD models for innovative technology,
- identifying methods to extract information from expert performers,
- determining the best way to employ technology,
- identifying other fields that make contributions to learning, and
- identifying what instructional problems technology can solve.

Designers may need to modify certain components of the ISD model in order to incorporate recent technological advances. What appears to be needed are improved ISD models tailored to current technology such as hypertext/hypermedia, artificial intelligence, or embedded training. Most ISD models do not provide explicit methodology for designing instruction for hypertext/hypermedia or embedded training (Clark, 1989).

Some experts feel that learning theorists and designers need a new "mind set" before technology can help improve instruction (Galagan, 1987). Others have suggested that designers will be forced to work within the limits of the technology and must start thinking more about what they want to teach and how to teach it. Finally, Gery (1989) states that the most significant advances in improving human performance (how humans learn) will not come from innovative technology but from the way in which the technology is applied.

5. SUMMARY

The successful implementation of ISD is directly related to the expertise and commitment of designers to perform diligently each step in the ISD process. The ISD process is basically characterized by five steps including: analysis, design, development, implementation, and evaluation. The exciting potential of using new technologies in computer-based instruction lies in creating individualized learning programs that can be modified for respective learners and their individual learning styles. If the promise of new technology, coupled with relevant research findings, comes true, training programs designed with an awareness of cognitive style characteristics should not only decrease learning time but should result in more effective instruction. Designers will continue to be challenged to apply higher order cognitive skills through simulations and hypertext technology. The utilization of current research information in the area of cognitive psychology and presentation formats of learning material can certainly facilitate ISD. Finally, the need for modified ISD models to accommodate new technologies, especially artificial intelligence, hypertext/hypermedia, and embedded training, is the current challenge of ISD researchers and developers.

Currently, the components of the ISD models are being revamped to include principles from cognitive psychology. The capabilities of technology combined with new knowledge in human learning have begun to shape both the expectations and the relationship between users and technology in the instructional process.

6. REFERENCES

Andrews, D. H. and Goodson, L. A., "A Comparative Analysis of Models of Instructional Design," *Journal of Instructional Development*, **3**(4), 2-16 (1980).

Applied Science Associates et al., *Lessons Learned from Embedded Training Design Process for ASAS/ENSCE*, U.S. Army Research Institute for the Behavioral and Social Sciences, September 1988.

Back, S. M. and McCombs, B. L., "Factors Related to the Success or Failure of Self-Paced Instruction in Air Force Technical Training: A Literature Review," (AFHRL-TP-84-24). Lowry Air Force Base, CO: Air Force Human Resources Laboratory, Logistics and Technical Training Division, August 1984.

Bandura, A., *Social Learning Theory*, Englewood Cliffs, NJ: Prentice-Hall, 1987.

Branson, R. K. and Grow, G., "Instructional Systems Development," *Foundation of Instructional Technology*, ed. R. M. Gagne, Educational Technology Publications, Englewood Cliffs, NJ, 1987.

Briggs, L. J. ed., *Instructional Design*, Educational Technology Publications, Englewood Cliffs, NJ, 1979.

Briggs, L. J., "Instructional Design: Present Strengths and Limitations, and a View of the Future," *Educational Technology*, **22**(10), 18-23 (1982).

Buehner, L. J., "Instructional Design: Impact of Subject Matter and Cognitive Styles," Education Department, Wittenberg University, Logistics and Human Factors Division, Wright-Patterson Air Force Base, Ohio, February 1987.

Clark, R. E., "Current Progress and Future Directions for Research in Instructional Technology," *Educational Technology Research and Development*, **37**(1), 57-66 (1989).

Clark, R. E., "Reconsidering Research on Learning from Media," *Review of Educational Research*, **53**(4), 445-459 (1983).

Dede, C. J., "A Review and Synthesis of Recent Research in Intelligent Computer-Assisted Instruction," *International Journal of Man-Machine Studies*, **24**(4), 329-353 (1986).

Dede, C. J. and Swigger, K., "The Evolution of Instructional Design Principles for Intelligent Computer-Assisted Instruction," *Journal of Instructional Development*, **11**(1), 15-22 (1988).

Dick, W. and Carey, L., *The Systematic Design of Instruction*, F. Scott and Company, Glenview, IL, 1978.

Ely, D. P., "Education and Information Technology: What are the Questions?" *Trends and Issues in Education*, ed. G. Flaxman, ERIC Clearinghouse on Information Resources School of Education, Syracuse University, January 1987.

Gagne, R. M., "Instructional Technology: The Research Field," *Journal of Instructional Development*, 8(3), 7-14 (1986).

Gagne, R. M., *The Conditions of Learning*, Holt Rinehart and Winston, NY, 1965.

Galagan, P., "Computers and Training: Allies or Enemies?" *Training & Development Journal*, pp. 73-76, April 1987.

Garnham, A., *Artificial Intelligence: An Introduction*, Routledge and Kegan Paul Ltd. in association with Methuen, Inc., NY, 1988.

Gery, G., "Electronic Performance Support Systems," *CBT Directions*, pp. 12-15, June 1989.

Hannum, W. H. and Hansen, N., "Instructional Systems Development in Large Organizations," *Future Directions for Instructional Systems Development*, Educational Technology Publications, 207-225, Englewood Cliffs, NJ, 1989.

Hardy, H. C., "Armor Training 1997: An Application of Embedded Training," *Armor*, November - December 1988.

Harmon, P. and King, D., *Expert Systems: Artificial Intelligence in Business*, John Wiley & Sons, NY, 1985.

Harvey, F. A., "Programming in Logic and Implications for the Development of Intelligent Technology-Based Instructional Systems," 30th ADCIS Conference Proceedings, Philadelphia, PA, November 1988.

Hooper, S. and Hannafin, M. J., "Learning the ROPES of Instructional Design: Guidelines for Emerging Interactive Technologies," *Educational Technology*, pp. 14-18, July 1988.

Johnson, W. B., "Pragmatic Considerations in Research Development and Implementation of Intelligent Tutoring Systems," *Foundations of Intelligent Tutoring Systems*, ed. M. C. Polson, and J. J. Richardson, Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, 1988.

Jonassen, D. H., "Designing Structured Hypertext and Structuring Access to Hypertext." *Educational Technology, Special Issue: Hypermedia*, November 1988.

Kearsley, G., "Authoring Considerations for Hypertext," *Educational Technology, Special Issue: Hypermedia*, November 1988.

Kearsley, G., "Embedded Training: The New Look of Computer Based Instruction," *Machine-Mediated Learning*, 1(3), 279-296 (1985).

Kerlinger, S. N., "The Influence of Research on Education Practice," *Educational Researcher*, pp. 5-12, September 1977.

Marchionini, G. and Shneiderman, B., "Finding Facts vs. Browsing Knowledge in Hypertext Systems," *IEEE Computer*, January 1988.

McCombs, B. L., "The Instructional Systems Development (ISD) Model: A Review of Those Factors Critical to its Successful Implementation," ERIC/ECTJ Annual Review Paper, ERIC Clearinghouse in Information Resources, Syracuse, NY, 1986.

Merrill, M. D., Component Display Theory, in *Instructional Design Theory and Models* ed. C. M. Reigeluth, Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, 1983.

Merrill, M. D., "Don't Bother me with Instructional Design - I'm Busy Programming!: Suggestions For More Effective Educational Software," *Computers in Human Behavior*, 4, 37-52, 1988.

Oxford-Carpenter, R. L. and Schultz-Skiner, L. J., "Army Job Training Development and Testing Practices Compared to the Instructional Systems Development Model," *Training Technology Journal*, 1(3), 32-41 (1984).

Reigeluth, C. M., *Instructional Design Theories and Models*, J. Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, 1983.

Reynolds, R., et al., Lessons Learned from Currently Fielded Navy Embedded Training Systems, Naval Training Systems Center, 1987.

Rumelhart, D. E. and Norman, D. A., Analogical Processes in Learning. In *Cognitive Skills and Their Acquisition*, ed. J. R. Anderson, Lawrence Erlbaum Associates, Hillsdale, NJ, 1981.

Schoen, S. and Sykes, W., *Putting Artificial Intelligence to Work: Evaluating and Implementing Business Application*, John Wiley and Sons, Inc., NY, 1987.

Shneiderman, G. and Kearsley, G., *Hypertext Hands On*, Addison-Wesley Publishing Company, Massachusetts, 1989.

Skinner, B. F., *Technology of Teaching*, Appleton-Century-Crofts, NY, 1968.

Stiehl, R. E. and Streit, L. D., "Factors Assuring Success of an Instructional Systems Model," *Educational Technology*, **24**(8), 23-27 (1984).

Thorndike, E. L., *Educational Psychology*, New York Teacher's College Press, 1921.

Vineberg, R. and Joyner, J. N., Instructional System Development (ISD) in the Armed Services: Methodology and Application, (HumRRO-TR-80-1), Human Resources Research Organization, Alexandria, VA, January 1980.

Section 6

ADVANCED COMPUTER AIDED INSTRUCTION MODEL FOR THE NALDA SYSTEM

TABLE OF CONTENTS

LIST OF FIGURES

ADVANCED COMPUTER AIDED INSTRUCTION MODEL FOR THE NALDA PROJECT

1. BACKGROUND

Computer aided instruction (CAI) has become a primary means of providing training in a variety of organizations. In the years since its inception, CAI has proven its ability to deliver efficient and cost-effective training to individuals. Section 1, *Review of the Research in Computer Aided Instruction,* of this report contains details of the research related to computer aided instruction.

1.1 HISTORY

Numerous studies over three decades have demonstrated that users completing CAI lessons are more likely to achieve instructional objectives than users using more traditional training approaches. A major finding from these studies is that people using CAI lessons consistently required less time to reach the instructional objectives than users receiving traditional training. Most users rated the CAI lessons more positively than they did other forms of training, especially traditional methods. Certainly, CAI seems to be an acceptable method of training from the user's point of view.

1.2 ADVANTAGES

CAI frees organizations from many constraints of traditional training approaches, especially the lockstep requirements of classroom instruction. It is not essential that everyone receiving training start at the same time, in the same location, and at the same point in the material. Organizations can provide more individualized training to employees while reducing the cost per employee. Given such positive results both from research and practical experience, there is little wonder that CAI has become a primary training method in many organizations.

The emphasis should now be on improving the training delivered via CAI. Displaying poorly conceived and designed materials on computer screens will not improve the instructional effectiveness of the materials. It is the appropriate use of the capabilities of computers, in conjunction with well-designed training materials, that results in effective CAI.

2. CAI FOR THE NALDA SYSTEM

The Naval Aviation Logistics Data Analysis (NALDA) system is an automated information system that provides data to support numerous Navy activities. The NALDA system is complex; therefore, its use and maintenance require comprehensive training. Personnel from different fields, such as safety, supply, and engineering, use the NALDA system, and the lack of hardware uniformity used to access NALDA, makes a generic training approach difficult. Martin Marietta Energy Systems, Inc., Data Systems Research and Development (DSRD) staff conducted a study of the training needs of NALDA system users in 1986. The study indicated that traditional formal NALDA training could be enhanced with CAI.

2.1 DATA SYSTEM RESEARCH AND DEVELOPMENT REPORT

Several approaches to NALDA training were evaluated. CAI with concurrent links to the mainframe application was recommended. Several CAI authoring systems were considered and the Electronic Publishing System (EPS) from Intellisance was selected as the best authoring package to use for developing the NALDA CAI.

The content, as well as the methodology, of the existing NALDA training program was examined in a comprehensive needs assessment and analysis of NALDA. It was concluded that NALDA users should have a knowledge of basic data base management systems concepts, specific System 2000 (S2K) information, and Maintenance and Material Management (3M) data, in addition to NALDA-specific knowledge.

Based on the analysis of NALDA training needs, specifications which include methods of presentation, lesson navigation, lesson flow control and testing, delivery methods, and design standards were developed. A curriculum for NALDA training needs and a proposed plan of action were also specified. Prototype lessons were developed and a program generator was created to simplify CAI development.

2.2 PURPOSE

The intent of this report is to build on previous work by exploring current state of the art CAI practices and describing additional effective approaches to instruction. An advanced instructional model for the design and delivery of training is described in Appendix B. Three previous DSRD reports contain detailed background information regarding NALDA training and the development of the prototype CAI. These reports are *Recommended CAI Approach for the NALDA System,* (Duncan, et al., 1987), *Design Specifications for NALDA CAI - Phase II Interim Report* (Twitty, et al., 1987) and *NALDA CAI Prototype - Phase II Final Report* (Handler, et al., 1989). The present report focuses on the practical application of research in the design of an advanced CAI model.

2.3 ASSUMPTIONS

The starting point in designing an advanced CAI model is a set of assumptions about training and its purpose:

- Training exists to fill identified performance needs.
- Specifically, training is designed to increase the knowledge and upgrade the skills of employees.
- Training should be based on the requirements of the job and should lead to tangible results.

Training is not an end in itself but a means of reaching organizational goals. Training should result in a relatively uniform, high level of achievement among users. Training should not produce a normal distribution of results in which many people are not successful, but rather training should be held accountable for producing results, such as users achieving mastery of the content. Training should be efficient in terms of time spent in training and the costs devoted to the training effort. These are some fundamental assumptions concerning the role of training in an organization and should guide the design and development of any advanced CAI model.

2.4 THEORETICAL BASIS FOR CAI DEVELOPMENT

In addition to any assumptions made about the role of training, the design of an advanced CAI model is rooted in and influenced by certain theoretical assumptions about learning. Theories of instruction are based directly on theories of learning. Various CAI approaches can be derived from the study of instructional and learning theories such as the themes of behavior and cognition.

2.4.1 Behavioral Theory

Behaviorism was the dominant learning theory in the late 1950s and early 1960s when CAI was initially formulated. Thus, CAI was firmly oriented towards behavioral ideas of learning and instruction. Behavioral theory held that learning was a change in behavior caused by the consequences of behavior. Good learning environments require that desired behavior be clearly defined and reinforced when it appears. This necessitates that the user make frequent responses. If the behavior is unlikely to appear, then it must be shaped by reinforcing successive approximations of the desired behavior. Since only the correct behavior or approximations thereof can be reinforced, the instruction must proceed slowly and include many prompts to ensure the correct responses. For the same reason, users should be allowed to advance through lessons at their own pace. Early CAI operationalized these behavioral ideas about learning in the form of automated programmed instruction. The computer presented a series of frames, each of which gave the user some brief information (usually no more than a sentence), called for a response, and then provided reinforcement in the form of confirmation that the response was correct. Most responses were correct due to the extreme amount of repetition and heavy prompting behavioral psychology.

As the practice of CAI design and development continued in the behavioral tradition, researchers of human learning began questioning the adequacy of these behavioral principles. Many of these principles were found not to be essential for learning. Users can learn without frequent overt responding; reinforcement isn't necessary; users can follow instruction that isn't in a simple linear sequence; users can advance in bigger steps; users don't have to receive frequent reinforcement; and they can learn even while making errors. The rise of cognitive theories of learning, beginning in the 1970s, offers a different perspective on learning and, in turn, a different theoretical basis for the design of CAI.

2.4.2 Cognitive Theory

Cognitive theory stresses the organization and structure of material, a user s prior knowledge about the content to be learned, the meaningfulness of the material, grasping the "big picture," active processing of the material, and understanding the underlying concepts and principles of the content.

Current research and thought are in accord with cognitive theory and in opposition to behavioral theory. The design of an advanced CAI model that incorporates recent thinking about learning will be based on cognitive, rather than behavioral, psychology. The assumption, based on considerable research, is that cognitive theory more adequately accounts for and explains human learning. Thus, the designs of the CAI models presented in Appendices A and B are guided by concepts from cognitive psychology.

3. INSTRUCTIONAL APPROACH FOR CAI MODEL

Two different approaches are taken in the design and development of the advanced CAI model for this project. The approaches differ in orientation depending on what is to be taught. One approach is knowledge-oriented; the other is skills-oriented. This dichotomy is predicated on the distinction between the learning of knowledge and the learning of skills. In the analysis of the NALDA training needs and in the specification of a curriculum for training, both increased knowledge and upgraded skills are identified as necessary results of training.

Considerable research and theory have provided evidence for distinctions between the learning of knowledge and skills. Research also provides support for different instructional approaches in teaching knowledge and skills. Given that different instructional conditions are necessary for efficient teaching of knowledge and skills, the advanced CAI model for this project will incorporate two distinct instructional approaches. (See Appendix A for the overall NALDA CAI model.)

3.1 HYPERTEXT FOR KNOWLEDGE

One expected NALDA training outcome is knowledge of the 3M system which requires an understanding of many facts and their interrelationships. In order to master 3M, the user must be able to understand the particulars in context, not in isolation. The 3M instruction should make interrelationships explicit so users can develop an understanding of how the content is organized. Ideally, the instruction should allow the users to explore the 3M content by moving about within the system to see how one fact is related to another. Allowing the user to have freedom of movement within a course of study is consistent with the design specifications for lesson control and lesson navigation specified in previous DSRD reports.

Hypertext is a relational database composed of nodes of information. Movement from one node to the next is made possible by links. Hypertext allows users to move about freely within the content domain, exploring relationships among items of content. When users are able to move about in such a fashion, they develop more stable representations of the knowledge in their own cognitive structures. Users also form more links between the new knowledge and their existing knowledge, making the new knowledge more meaningful and therefore less easily forgotten.

To create a hypertext environment in the advanced CAI model, the 3M knowledge must be organized and structured in the form of a web diagram or network. Figure 6.1 shows a representation of a web network of knowledge. A diagram or network shows facts, or pieces of knowledge, as nodes. The relationships among the facts are shown as links or lines connecting the nodes.

The hypertext portion of the advanced NALDA CAI model for this project must support the development of structures for the 3M knowledge that underlies successful job performance. The content nodes along with the connecting links must be built

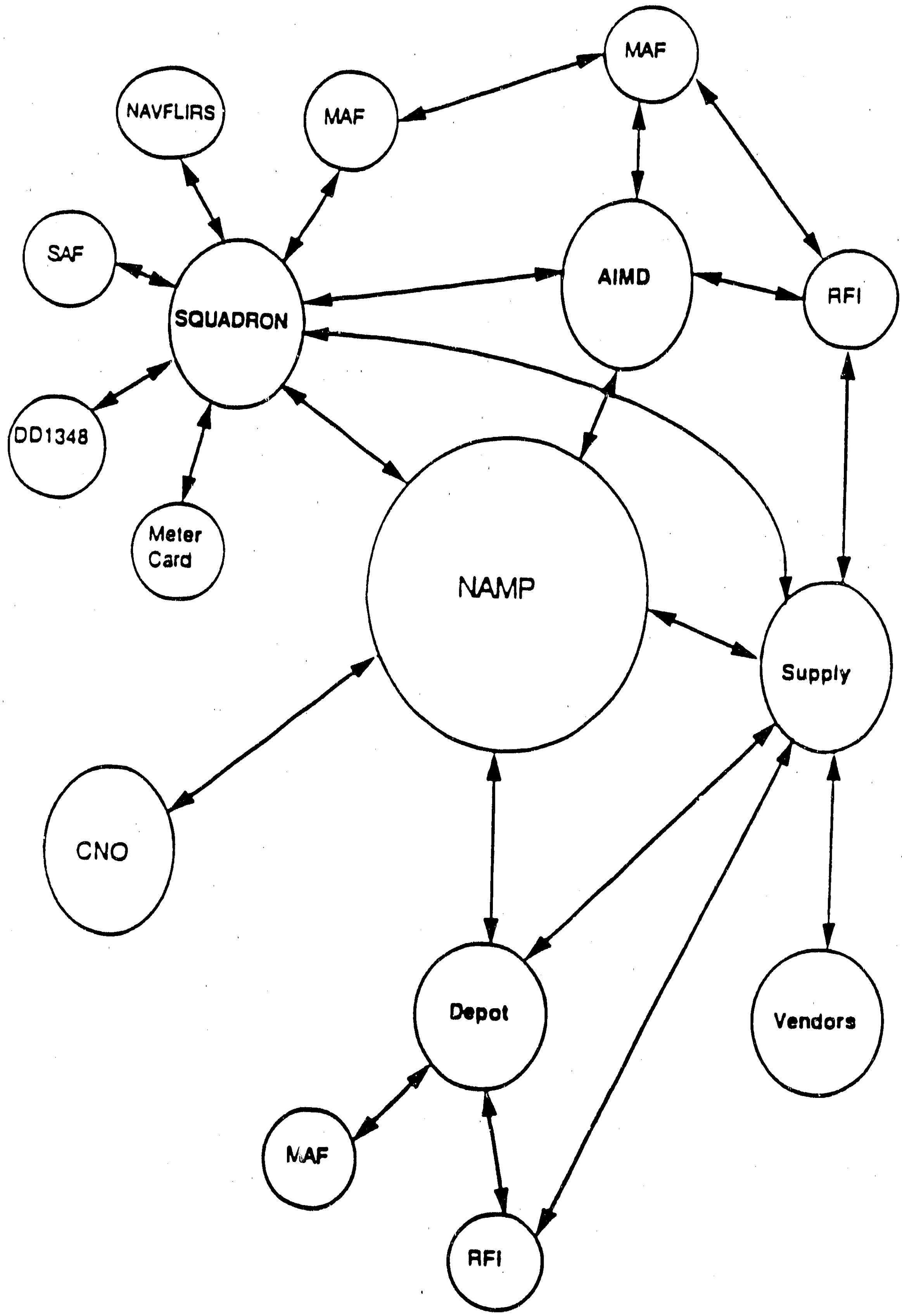


Fig. 6.1. Web network for AV-3M.

into the computer-based instructional model. Several hypertext programs will support such development. In a hypertext program, users can "browse" in the knowledge structures, exploring any links of interest. User controlled browsing makes content more meaningful. When there is a question about the relationship between items A and B, the answer can be sought through the use of hypertext. Different users with differing questions may explore individual interest paths within the 3M information.

3.1.1 Hypertext and Individual Differences

Hypertext adapts to individual differences among users by allowing each user to move about freely within a content area. Each user can explore the 3M knowledge base, focusing on areas of specific need. The DSRD needs assessment identified a wide range of user profiles; therefore, an instructional model that recognizes individual differences is of major importance. Some users will be novices; others will be experienced. Personnel from supply, maintenance, safety, and engineering will be using NALDA. Each user has different information needs and priorities. Some users will need structure while others will make use of the freedom to explore.

The hypertext segment of the CAI model will afford advanced users free access to facts they may have forgotten. If a user has a general understanding of a knowledge domain and the pertinent facts in the domain, understanding can be increased through the use of hypertext.

The novice, on the other hand, would use a structured approach to the hypertext knowledge base. If, as a novice, the user has little understanding of the structure or facts, a guided tour of the knowledge base is needed. Following a structured overview and guided tour, the full features of hypertext are made available to these users. The hypertext portion of the advanced CAI model allows users to view a knowledge base by following paths suggested by experts. Navigating along the recommended paths can prevent the user from feeling lost.

3.1.2 Hypertext from a User's Point of View

Users of the hypertext system can move through the hypertext in one of three manners, based on their familiarity with the content. A user with relatively high familiarity would move about the knowledge base at will, exercising user control over topic selection and sequence. An experienced user could use the hypertext system as a reference tool, looking up a forgotten fact or exploring the relationships among previously learned facts.

A user having moderate content familiarity with a portion of the knowledge base could use the hypertext system in a similar manner. The average user hasn't mastered all of the knowledge base but is familiar with some of it. When the average user needs assistance, help from the hypertext system is readily available. Users would receive an indication of their current position in the structure of the knowledge base and suggestions for which paths to pursue based on how experts organize the knowledge.

Because novice users would not know many facts in the knowledge base or their interrelationships, they would follow a structured approach to using the hypertext system. The novice would be "walked" through the knowledge base. First, users would be shown the main categories of the knowledge base. The list of categories is a structured overview of the content. Next the novice would be presented with details on one of the major topics. The detailed list is similar to a list of chapter headings in a book and is considered to be a second-level elaboration of content. The novice is slowly given more freedom to explore the knowledge base contained within the hypertext system. The new user will be able to ask for help in navigating through the hypertext system at any time. The help function will reveal the overall structure of the knowledge base and suggest a path to follow. In time, the novice user will be able to use the hypertext system without instructional support.

3.2 SIMULATIONS FOR SKILL DEVELOPMENT

The second approach proposed in the advanced CAI model is simulation with embedded tutorials. The simulation portion of the advanced CAI model is designed to be used after the user has 3M knowledge and wishes to develop skill in using the NALDA system. It is not suggested that simulation be used solely as a "discovery learning" approach in which someone with no knowledge is placed in a simulation and allowed to explore without constraint. Before entering a simulation, users should have some background knowledge and understanding of the purpose of the simulation. The simulation is designed to help the user develop skills for using the NALDA system as efficiently as possible, and, therefore, the simulation contains instructional support not found in the live system.

3.2.1 The Simulation Design

Users will be pretested and will move into the simulation mode when pretest scores indicate their readiness. If a user's test score indicates insufficient background information, the student will be branched to a tutorial before being allowed to enter the simulation.

Once in the simulation, all users have access to brief embedded tutorial segments tailored to any specific procedures required. The previous analysis of NALDA training, completed by DSRD, found that the NALDA users differ considerably in their prior knowledge. The original design specifications indicate that the training approach should accommodate individual differences among users. By using a pretest and adjusting the instruction to fit the user's level of prior knowledge, the advanced CAI model will accommodate the various needs of the individual users.

The simulation aspect of the overall advanced CAI model will be used to enhance users' skills. Specifically, the advanced CAI model will simulate the operation of the NALDA system. Once basic NALDA information has been acquired, the learners will move to a simulation. While in the simulation, the user can view an example of a particular procedure and then try the procedure. When the learner attempts a procedure, the simulation will provide feedback concerning the adequacy of the user's performance. In this manner, the advanced CAI model provides a supportive environment in which the user can apply information learned. Instructional simulations of this type will provide an opportunity for learning and practice before exposing a novice to the live system.

Embedded training is a key aspect of the simulation portion of the advanced CAI model and has been suggested in earlier recommendations. Embedded training represents an effective instructional approach to the needs of NALDA training. Some computer-based simulations create a total discovery learning environment in which users are free to explore but have little guidance or assistance available. The simulation in the advanced instructional model will include on-line assistance in the form of embedded training. While working through the simulation, users will have access to assistance in the form of an electronic tutor. The electronic tutor allows users, encountering difficulties while in the simulation, to access the embedded training module and get help related to their specific problem. If a user cannot remember the specific format for a command that is required for retrieving an item of data, the embedded training system can supply the command structure as well as an example of its use. When ready, the user can move back to the simulation at the exact point of departure and enter the correct command.

The goal of the simulation is to teach how to use the actual system. The simulation does not have to duplicate all aspects of the operational system; rather, it is only necessary to capture the user interface. The simulation deviates from the actual system in that it provides more instructional support for the user than does the actual system. The instructional simulation will furnish more elaborate feedback to enable the user to understand and correct mistakes. Thus, the simulation extends beyond the actual system by providing instructional support.

While in the simulation mode, users can access three levels of help. The first level provides a description of the commands currently available. The second level of help gives an example for each command. The third level explains, or teaches, each available command. The help system is layered, so a user can ask for more help as needed. Previous DSRD reports indicate a need for different levels of on-line help with the flexible use of open entry and exit points throughout the simulation. The advanced CAI model incorporates these recommendations in the design of the simulation.

3.2.2 Simulation from a User's Point of View

Users entering the simulation should have some prior knowledge related to the NALDA system. A brief pretest will ascertain if a user has enough of the prerequisite knowledge to enter the simulation. Users without the necessary knowledge will be branched to a tutorial segment for instruction. Users with the prerequisite knowledge can enter the simulation directly.

When the user is in the simulation, a scenario related to NALDA use is presented, and the user is asked to respond to the situation. For example, the user could be given the situation in which information from a NALDA database is needed, thus requiring the construction and entry of a series of commands to retrieve the information. The simulation would respond to each command by showing facsimiles of the actual NALDA system responses that would be generated by the user's commands errors and all.

In addition to showing a simulation of actual NALDA outputs that would be generated by a user's actions, the simulation portion of the advanced CAI model provides the user with instructional support that is lacking in the actual NALDA system.

When the user makes an error during the simulation, the actual response that would be generated by the NALDA system is displayed. In addition the simulation provides the user with an explanation of what is wrong and how to correct it. After the user makes an error and receives the NALDA system feedback, an instructional window opens over a portion of the NALDA screen and an explanation of the error appears in the new window. The user is then led through the correct command sequence by following the step-by-step directions that appear in the instructional window. The instructional window closes, and the user returns to the simulation for additional practice.

The user can request help at any time if feeling lost or confused. For example, when the simulation presents a scenario in which a particular command must be used that the user doesn't remember, the user can request help. This request for help is followed by the opening of an instructional window which contains the syntax for the requested command. The user can then return to the simulation or request additional help. If more help is requested, the user receives second level help for that command in the form of a completed example. If this is sufficient, a user returns to the simulation. If not, the user receives a third level help in the form of a brief tutorial segment about how to construct the specific command. After viewing the third level help, the user returns to the simulation mode. Once the simulation can be completed without evoking the help system, the user is ready to go to the live system. To allow the user to move easily to the NALDA system, the advanced CAI model must create a simulation that accurately mimics NALDA in terms of user interface and system response. While in training, the user is not actually on the live system, since this would slow system response and incur telecommunications costs. Rather, the user works on a simulated system and can move back and forth between embedded help and the simulated system. The alternative of embedding help or training in the live system is not incorporated into the advanced CAI model because of potential problems with system response and capacity, problems with data security, and the need for more elaborate analysis of users' responses than the live system allows. From an instructional perspective, the simulated system has many advantages over the live system.

4. DESIGN CONSIDERATIONS

Among the many areas to be considered in the design of an advanced CAI model are standards, screen design, control options, administration and management issues, and concern for the user.

4.1 STANDARDS FOR THE ADVANCED INSTRUCTIONAL MODEL

Three categories of standards have been identified as fundamental to the design of CAI: instructional design, lesson design, and user-courseware interaction.

Standards relating to instructional design include identifying and analyzing instructional goals, stating objectives, assessing performance, sequencing content, presenting content, and evaluating results. Lesson design standards encompass such areas as lesson flow, instructional events, screen format, and lesson control features. The quality of user-courseware interaction depends on setting standards for stimulus, response, and feedback attributes.

4.1.1 Instructional Design

Instructional design standards involve identifying objectives, matching the lesson to the audience, and deriving instructional content. The objectives for an advanced CAI model should be stated in measurable terms which are based on a task analysis of the job for which the training is being developed, and should be appropriate for the intended audience. Objectives serve to guide the development of lessons and should be specific and tangible so that they communicate the intent of the lesson in a clear, unambiguous manner. Therefore, objectives should be stated in measurable terms. For training to be both effective and efficient, objectives must be based on job tasks identified for training in a formal task analysis. Meeting training objectives should allow users to enhance their on-the-job performance. The objectives should be realistic, neither too hard nor too easy for the intended audience. Meeting training goals should narrow the gap between the requirements for successful job performance and the current level of functioning of the trainees.

Instructional designers should consider who their audiences will be. Lessons should be based on an analysis of the audience's prior knowledge and skills to prevent a lesson from duplicating content already mastered. An accurate assessment of the user population will also prevent undue frustration and failure caused by beginning a lesson at a point beyond the user's current level of expertise. The examples included in the lesson should fit the audience and exemplify the content of the lesson. Likewise, the vocabulary should be appropriate for the user.

The instructional content should closely match the objectives stated for the lesson. Content not related to objectives in a direct manner should be discarded. CAI lessons, as well as other forms of training, should be driven by the need to know; lessons should not be "knowledge dumps" of content. All content which directly supports the objectives should be included allowing the CAI lessons to become focused.

Instructional design standards will be used to develop the advanced CAI model to ensure the focus and adequacy of the training and will guide the development of both the hypertext and simulation aspects of the overall model.

4.1.2 Lesson Design

The CAI lessons will be structured to promote learning. The instructional events model developed by Gagne (1985), based on the information processing model of learning, is recommended for use in structuring CAI lessons. Gagne's model for structuring a lesson consists of nine instructional events:

1. Gain the users' attention.

2. Inform the users of the objectives.

3. Stimulate recall of prerequisite knowledge.

4. Present new stimulus materials.

5. Provide learning guidance.

6. Elicit the performance.

7. Provide feedback on the performance.

8. Assess the adequacy of the performance.

9. Provide for transfer and retention.

The use of Gagne's lesson structure dictates the parts of a CAI lesson and the flow within the lesson. In the advanced CAI model, this structure is used for the tutorials that precede the hypertext and simulation instruction. (See Appendix A for a diagram of the instructional model.)

Lessons developed according to the stated plan begin by gaining the users' attention, possibly with instructionally relevant graphics. Next, users are told the performance expectations. Then, the users are reminded that the lesson takes into consideration their current background knowledge and skills. All of this sets the stage for new learning. At this point, new stimulus material is presented and the user is guided through it. Following the introduction of some new material, users are asked to respond, and they receive feedback. The sequence loops: new stimulus material, guidance, the user's response, then feedback. After a chunk of related material is completed, the responses are assessed and retention and transfer activities are initiated. Retention is promoted by delayed practice and reviews. Transfer is enhanced by including a variety of examples and exercises for the users.

The suggested lesson structure is based on cognitive learning theory and provides a much richer training environment than older drill and practice or tutorial models based on behavioral learning theory. The proposed approach represents an extension of traditional CAI models and is consistent with the original design standards.

4.1.3 User-Courseware Interactions

One of the primary attributes of CAI is its ability to provide interactive instruction to the user. In most training approaches, such as a video-based, text-based, and lecture-based, users have little opportunity to interact with the instruction. In carefully designed CAI training, the user interacts with the courseware on a frequent basis. The quality of these interactions is important in determining the effectiveness of the lessons. Three factors influence the quality of the interactions: stimulus, response, and feedback.

The stimulus for learning is well organized and provides the user with clear directions. The context of the instruction is familiar and meaningful. The instruction poses questions based on objectives. The questions are interspersed throughout the lesson and a set of review questions appears at the end. The lesson has periodic summaries and reviews to reinforce key points. Cues in the lesson focus the users' attention on salient points of the instructional stimulus. Finally, the CAI lesson maintains an appropriate level of difficulty and complexity by adapting to the level of the user. The advanced CAI model incorporates these standards, particularly in the tutorial and simulation segments.

The lesson calls for responses that closely match the stated objectives. The CAI lesson allows users to edit or make changes before their responses are evaluated. The simulation portion of the advanced CAI model calls for responses directly related to the job and thus to the objectives of the lesson. Users' responses should form connections among elements of the material and relate new material to previously learned material. The hypertext portion of the advanced CAI model fosters the development of these relationships among the content elements. Sophisticated answer analysis routines are used to evaluate users' responses. At a minimum, the answer analysis checks responses for common errors or misunderstandings, misspellings, and synonyms. By using instructional simulation instead of the actual applications software, the advanced CAI model allows such analysis of answers. The model reflects the use of artificial intelligence but stops short of an intelligent tutoring system that is self-modifying.

A provision for appropriate feedback is a key feature of effective CAI. Feedback is based on an analysis of users' answers and provides specific information to the users about mistakes or errors and explains incorrect responses. An advanced CAI model provides a hint following an incorrect response and confirmation following a correct response. Attempts to increase users' motivation through the use of "great work!," "good job," and the like must be monitored as their overuse is not effective and should be avoided. Feedback should be precise, brief, and clear to the user. The advanced CAI model uses such feedback in the simulation portion to ensure successful instruction.

The instructional design, lesson design, and user-courseware interaction standards all contribute to more effective CAI lessons and are therefore incorporated into an advanced CAI model. These standards closely follow the direction established in earlier phases of the NALDA CAI prototype development. Any apparent differences are due to the sequential nature of the reports. The goal of the present phase is to extend and update previous work. The differences in CAI design reflect an evolution in model development.

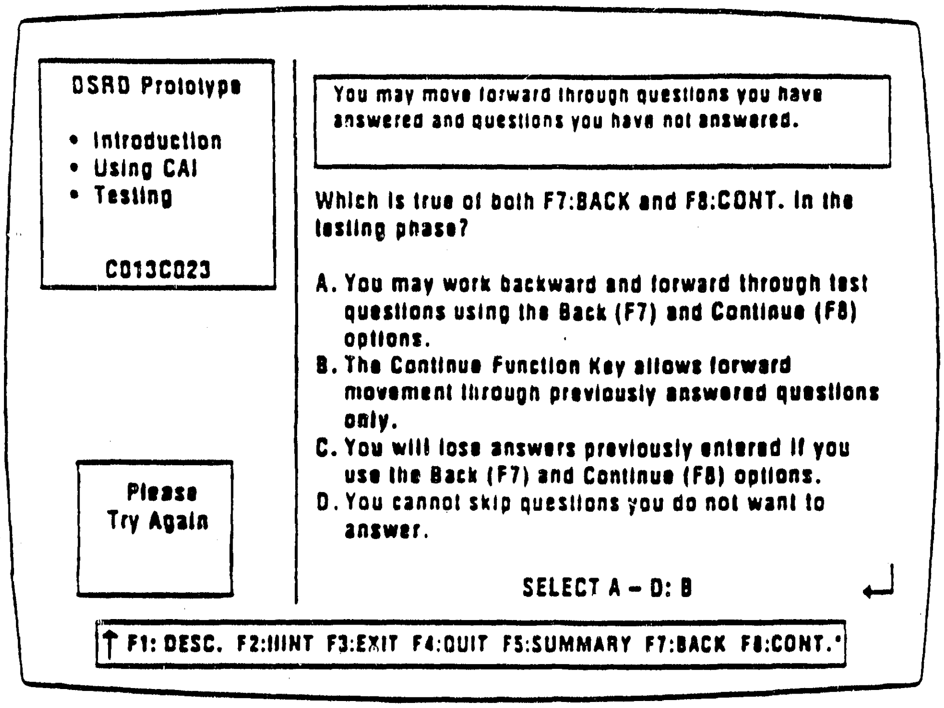
4.2 SCREEN DESIGN

The development of an advanced CAI model incorporates factors related to individual screen displays. The original screen design standards are the basis for the advanced instructional model. The suggested screen design standards are divided into three categories: layout, graphics, and text. The screen layout is free of clutter and uses a considerable amount of white space. (Approximately 60% of any screen is free of text or graphics.) Similar types of screens have a consistent format; all screens containing questions look alike and screens that provide help are similar in format. The placement of headings on each screen type and the location of the cursor are consistent. Functional areas, or windows, are used to establish parts of the screens that serve a particular purpose, such as providing directions or feedback. Specifically, the left portion of the screen is dedicated to aiding navigation by including the lesson and section name as well as the screen number when the screen is one of a sequence. The bottom of the screen contains a list of commands available for help, exit, and other navigation. Any text or question appears above the area designed for the user's response. Feedback appears above the screen text. These areas are made distinct by background color or borders and are separated by blank spaces. These design standards are the same as those presented in *NALDA CAI Prototype: Phase II - the Final Report* (Handler et al, 1989). Figure 6.2 is an example of both a criterion, or question, screen and a review text screen which incorporate the principles of effective screen design.

Graphics used in an advanced CAI model have an instructional purpose. Graphics are not used to adorn a screen or liven up the text but to convey an instructional message and be an integral part of the lesson. The graphics are also easily understood by the users. "Glitz" is not necessary in a CAI lesson and may detract from the instructional value of the lesson.

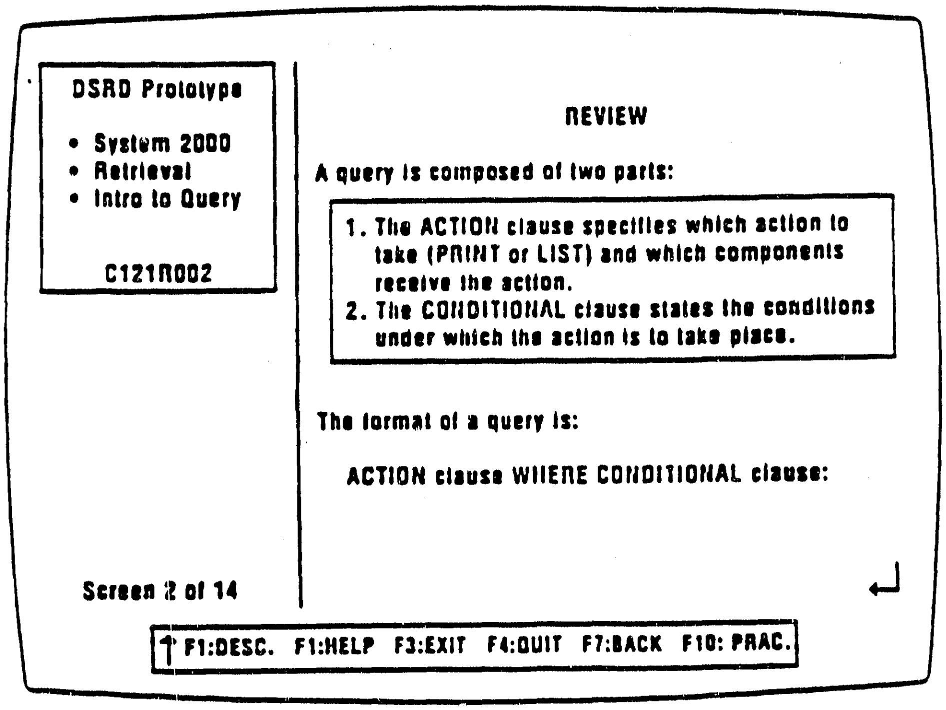
Instructional text is well organized and tightly written. The wording is clear and concise, and the vocabulary is appropriate for the audience. Screen text is in mixed case letters and the body of the text is left justified. The rate of presentation of the text is controlled by the user and not the system. Text does not scroll; instead, a new

Y-GA 88M-9959



Criterion screen with feedback.

Y-GA 88M-9952



Review text screen.

Fig. 6.2. Sample NALDA CAI screens.

screen is presented when needed. Highlighting text by inverse video and/or flashing characters is avoided except in rare instances. Color combinations are pleasant and afford adequate contrast to facilitate reading. Color choices should be limited and consideration must be given to individuals who are colorblind.

4.3 CONTROL OPTIONS

In CAI lessons there are three options for controlling the lesson flow or sequence.

In *programmed* control CAI, decisions about when to advance, skip content, show an example, present an exercise, and display an elaborate explanation are all prespecified and under control of the software. All users receive the same instructional content presented in the same order.

In *user* controlled CAI, learners have a choice regarding which topic to study as well as which instructional event to view next. Users can elect to see an example, get an explanation, look at an illustration, take a posttest, or get help as desired. Users determine what will be seen and in what sequence.

In *adaptive* CAI, the lesson adjusts, or adapts, to the user's needs. Adaptive CAI models include algorithms that dictate the adaptations. For example, the decision rule could be that if a user missed over 30 percent of the questions on a content topic, a remedial explanation of the topic would be given; otherwise the user would advance to the next topic. In the advanced CAI model, adaptive control is used in the tutorial module to adjust the presentation rate which was determined based on pretest performance.

An advanced CAI model forgoes program control as too restrictive and avoids total user control, since research has demonstrated that most users make poor choices in such a situation. Thus, an advanced CAI model is more appropriately conceived of as an adaptive instructional model, adjusting to the users during the learning process. Users may be given control over some aspects of the lessons when the control is preceded by system-generated advice. In essence, an advanced CAI model is an adaptive model that might, at times, give the users override options. This is particularly apparent in the hypertext portion of the advanced CAI model. The user is free to move about at will, but has the option to refer to a guide that shows an expert's organization of the knowledge and an accompanying path through the hypertext system. The advanced CAI model proposed in this report extends the initial CAI design although the hypertext portion of the model is restricted to experienced users.

4.4 ADMINISTRATION AND MANAGEMENT

The advanced CAI model includes administrative and management capabilities. The course administrator registers students, monitors their progress, prepares reports, and updates the curriculum.

The administration and management portion of the CAI model allows the course administrator to add a student to a specific course, transfer a student to a different location, delete a student from a course, and create reports of student progress as well as course evaluation reports.

4.5 USER CONSIDERATIONS

The advanced NALDA CAI model design was guided by concerns for meeting users' needs. The enhancements proposed in this report support this philosophy.

Information about the user of the CAI model must be considered in planning lesson content and the use of the model. Although it may now sound trite, the CAI model is "user friendly." The advanced CAI model has open entry and exit so the user can enter and leave the lessons at will. In both the hypertext and simulation modes, the user is able to restart at the last exit point or at the beginning of a segment. A user can skip the initial tutorial units by demonstrating mastery on the pretest. The CAI model is designed to provide instruction appropriate to the user's job and their "need to know." Control options include adaptive control of tutorials and user control of hypertext. Navigational aids are provided to help users maneuver through lessons. The structure of the lesson in the advanced CAI model is stated explicitly to aid users in navigation.

Help is available and readily accessible at all times through the embedded training option. Help includes assistance with the course content as well as with using the CAI system itself. The help is layered, ranging from brief reminders to very detailed information. The help screens are activated in a consistent fashion and are easily distinguished from other screen types.

5. DEVELOPMENT MODEL

The model used to guide the development of this advanced CAI model is presented in Appendix B. This development model specifies each step necessary in creating lessons, beginning with the establishment of goals for the course and concluding with the documentation of the software. The model identifies the inputs to each step, the specific processes to be followed in each step, and the resulting outputs. It is based on state-of-the-art instructional systems development (ISD) principles and is consistent with all previous DSRD developmental approaches. The development approach is designed to meet the specific needs of NALDA training and incorporates the structure of previous prototypes.

This development model incorporates some basic instructional design features common to other training approaches and some features unique to an advanced CAI model. While basic instructional design features are not a part of a CAI model, they are vital to successful CAI development and should be followed.

Two distinct training outcomes are desired from the advanced CAI system and each requires a different instructional procedure. The two types of knowledge to be gained from the advanced CAI are procedural knowledge and declarative knowledge.

Allowing users to remain in their work environment while being trained is an important feature of the advanced model since much of what needs to be learned is procedural. Some aspects of the instruction occur in tutorial CAI lessons before a new trainee begins the simulation that teaches the applications software used on the job. These lessons teach the procedures required by the applications software in a modified tutorial fashion. The trainees receive a brief description of the purpose of the applications software and how the application is used. They see an example of the application being used with each step highlighted. A step-by-step "walk-through" of the required procedures from the user's perspective is given next. This is not a description of all the applications program can do, nor is it a documentation of all the features of the program; it is a user-oriented description of how to do certain procedures with the software. Following this initial orientation and walk-through with the applications program, the user is provided an opportunity to enter commands necessary for successful completion of a job. The user is working with a simulation of the application and not the actual program.

Other aspects of the CAI lesson emphasize declarative knowledge, the facts that must be acquired to support smooth, efficient job performance, such as a knowledge of the 3M system. The advanced CAI model presents facts in an organized manner related to the user's job and need to know. The user sees the facts in hierarchical fashion and the explicit relationships among the facts. The user is free to browse through the facts, exploring at will in a hypertext environment. The high-level structure of the content is made apparent, and the user follows any connection to discover relationships among bits of declarative knowledge. The user zooms in or zooms out, obtaining more or less detail about a piece of knowledge. This hypertext feature supports the user in mentally forming a structured representation of the knowledge. The users have more control over these aspects of the lesson.

The advanced CAI model allows the successful design and development of a NALDA CAI that effectively incorporates both procedural and declarative instructional goals. Users who have successfully completed the NALDA training course will have acquired considerable knowledge of database management, the 3M system, and the NALDA system and will have the skills necessary to use the specific databases that make up the NALDA system. Successful job performance depends on these training outcomes being met.

6. CAI MODEL DEVELOPMENT

Ideally, the CAI model should run on the same hardware people use to perform their work in order to provide a smooth transition from work to training and back again. However, when there is a problem with conducting training on the hardware used to access the NALDA system, the training courseware may be installed on a stand-alone personal computer.

The development of courseware for the advanced CAI model is facilitated in several ways. Lesson templates are constructed and reused to ease the burden of "from scratch" development. Lesson components can be "plugged" into these templates to speed development. The program generator is used to create computer code. The existing program generator may need to be enhanced to incorporate features specified in the advanced CAI model. Graphic packages and scanned images are used to facilitate incorporation of screen graphics. Rather than constructing computer code to generate a graphic image, an acceptable paper image is found or created and digitized for incorporation into the CAI program. As an alternative, a graphics program could be used to create the image directly on a computer. The image would then be transferred into the CAI program. Finally, hypertext shells are used to develop the hypertext portion of the advanced CAI model to lessen the amount of programming required. All of these techniques are consistent with the existing approach to CAI development defined in *NALDA CAI Prototype: Phase II - Final Report* (Handler, Bryant, Duncan, et al., 1988).

7. SCENARIO

A visual depiction of the NALDA CAI model is presented in Appendix A. The model shows the relationships among the introductory tutorials, the 3M hypertext instruction, and the NALDA simulations.

Assume that a new user has just been assigned to receive training using the NALDA CAI system. The user signs on to the system and receives an introductory overview describing and demonstrating the major CAI features and procedures. Next, an overview of the lesson, listing its major parts and its organization, is presented, and a pretest is given. Figure 6.3 illustrates the features of the adaptive tutorial portion of the NALDA CAI model. Note that the model adjusts to the users on the basis of pretest scores as well as on performance within the lesson.

Using Gagne's instructional events module, the tutorial lessons will be similar in basic structure. A lesson can get the user's attention by showing a graphic of someone doing the job for which the user is being trained. Next, the user will be shown a diagram indicating where this lesson falls in the overall scheme of the course. Lesson objectives will be stated, and options within the lesson noted. If the lesson relates to extracting data from a database, the situation will be described, i.e., a user wants to generate a report based on an item of data in the database that reflects a modification to an F-18 aircraft engine. The lesson will then show an example simulating the process of data being extracted. The CAI lesson will go through the process used to request the data step-by-step; showing what the operator did at each step and highlighting the system's response. The user will then see a flowchart of the steps followed in the procedure. If the user has to make any decisions as a part of using the procedure, these decisions will be indicated in a decision table. This table will be organized in an IF...THEN...ELSE convention for ready access.

At this point, or at any point in the instructional process, the user can enter the hypertext portion of the advanced CAI model to explore related knowledge. Figure 6.4 illustrates the hypertext model for NALDA CAI showing how users can move through three major topics and two subtopics. Once a sub-topic is selected, the user can view a variety of lesson components including overviews, examples, exercises, definitions, and elaborations on the content. Users move freely through the hypertext material which is designed to improve understanding of the 3M system.

The simulation portion of the NALDA CAI presents a simulation of the NALDA application programs. The user practices specific procedures using the simulation and receives normal NALDA responses. Additionally, since this is a simulation, the user gets more elaborate feedback and can access a more elaborate help system than the NALDA system provides. If successful in the simulation, the user moves ahead to the next topic. If the user is not successful at any point in the simulation, the user's errors are diagnosed to determine the nature of the problem. Figure 6.5 illustrates the model by which user errors are diagnosed and remedied. Users making errors are placed in a tutorial segment that deals with the specific problem.

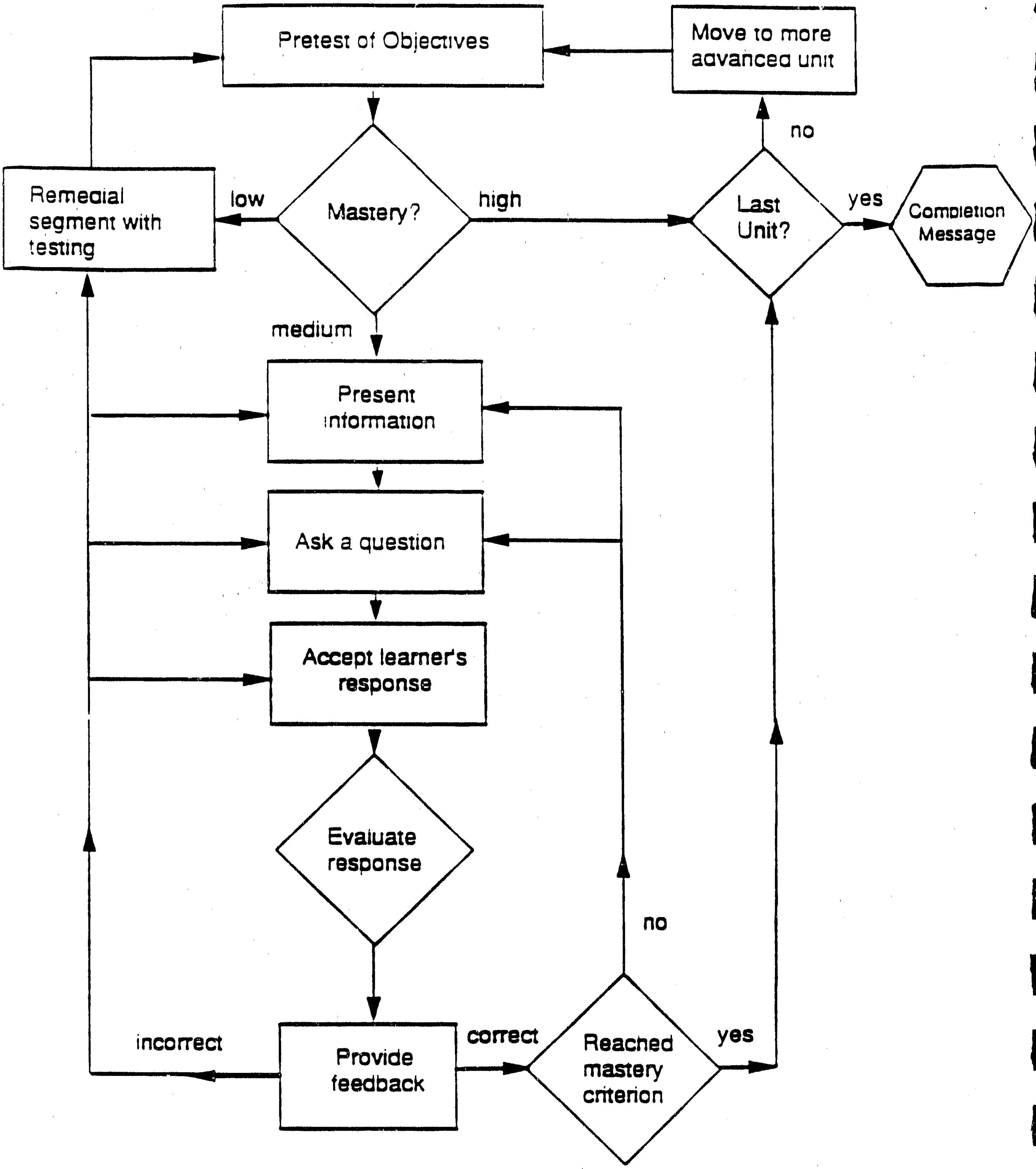


Fig. 6.3. Adaptive/tutorial CAI model.

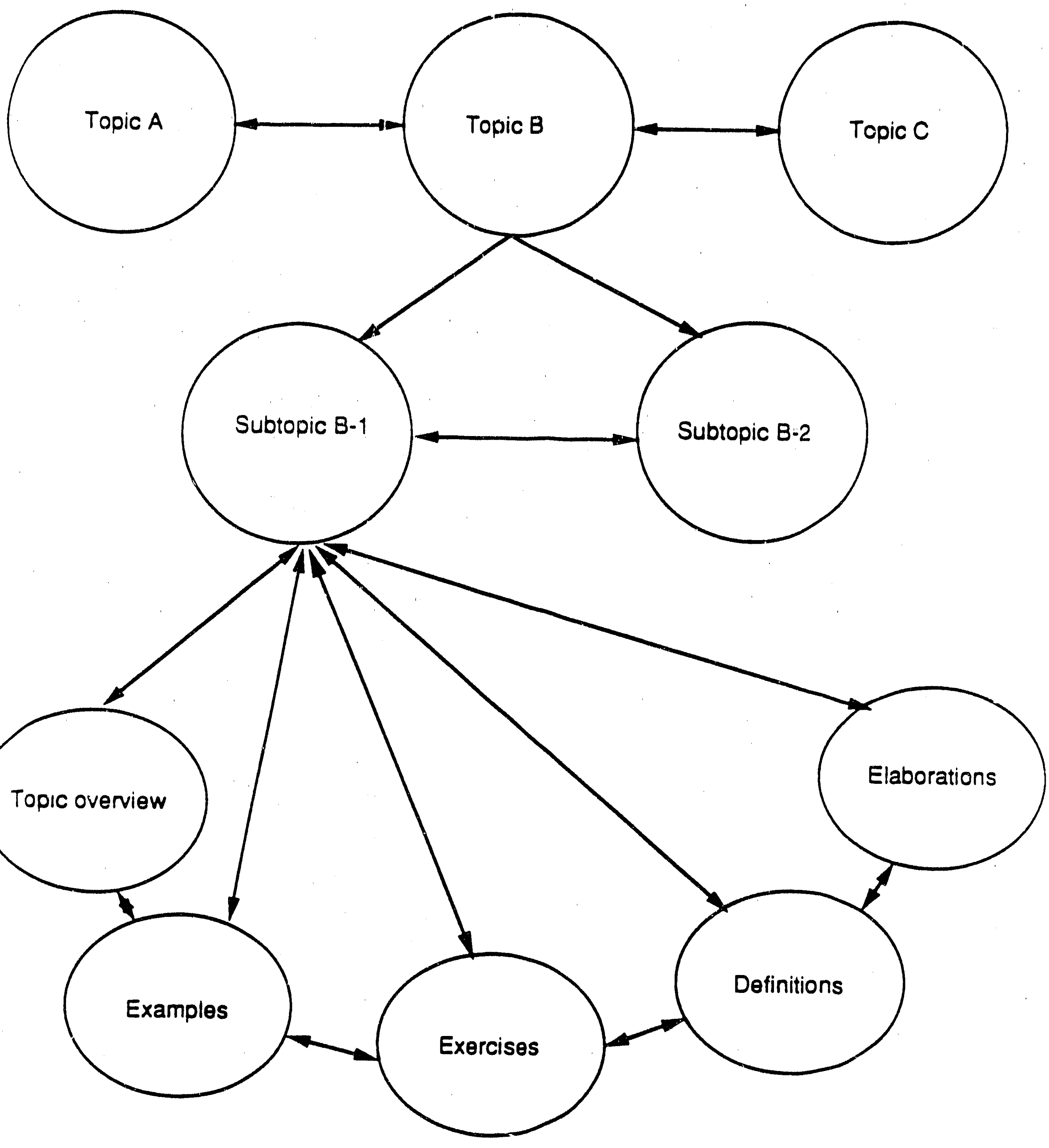


Fig. 6.4. Hypertext NALDA CAI model.

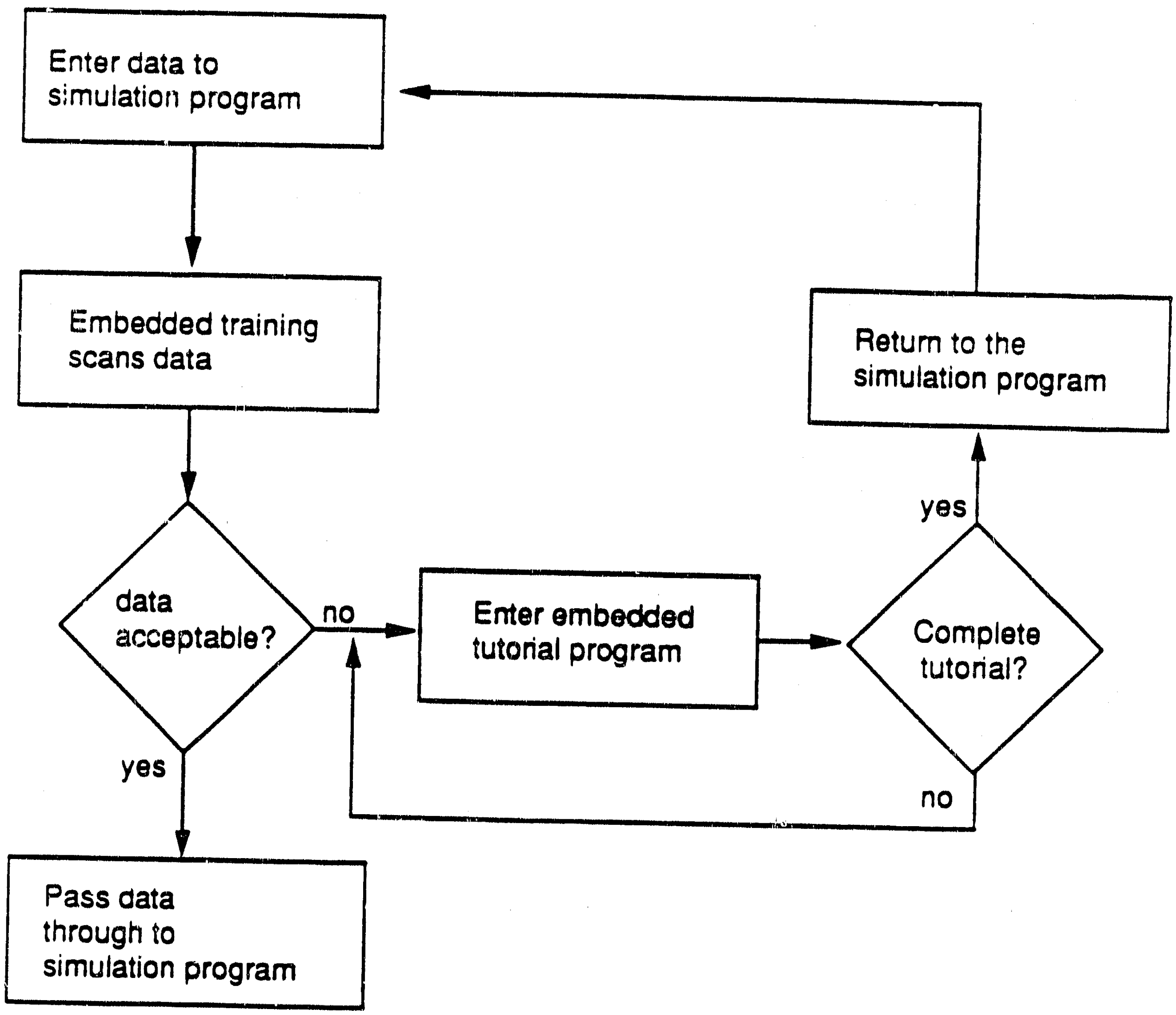


Fig. 6.5. Simulation with embedded tutorial.

They receive a more detailed explanation of basic concepts or procedures and view an example, using split-screen techniques, that show the example on the right and the CAI lesson generates a summary of the steps to follow, highlighting any special problems the user might encounter. The user continues the simulation until proficient and can then move into the NALDA operating environment using the applications program.

8. SUMMARY

The intent of the advanced NALDA CAI model is to make maximum use of instructional technology to provide NALDA system users with effective and cost-efficient training. The CAI model incorporates features that distinguish it from more traditional CAI. The advanced CAI model is considerably more than an electronic page turner, which presents programmed text or technical manuals on-line. The user-courseware interactions prevent this "reading" effect. The complexity of an advanced CAI model requires that it be an interactive computer-based system, not text-based.

Developments in artificial intelligence are incorporated into the CAI model in several ways. Answer analysis routines are based on an understanding of errors and some natural language processing techniques to ascertain a user's response in simulations. The gap between expert and novice knowledge guides the content development. CAI lessons are based on helping a novice develop an understanding of content that more closely approximates the content knowledge of an expert. The knowledge, as represented by an expert, is the basis for establishing the hypertext portion of the CAI model, which conveys the declarative knowledge. The adaptive control features of the advanced CAI model utilize certain artificial intelligence features. Indeed, the software is making decisions about which instructional material to present next. The advanced CAI model does, however, stop short of incorporating the self-modifying features of some artificial intelligence approaches. At this time in the evolution of artificial intelligence, self-modifying features are still too experimental for use in the proposed CAI model.

Training is made available on demand to support simulations of job performance. Job relevance is a key issue in the design model. A person using on-line training is able to move between simulations and tutoring modules easily and at will, to support smooth learning. The user can access the help system or the tutorial portion of the model for additional instruction. As previously indicated, the CAI model uses hypertext as the basis for conveying the declarative knowledge in support of job performance. Users browse through the hypertext at will to explore the interrelationships among 3M content items.

9. REFERENCES

Gagne, R. M., *The Conditions of Learning* (4th edition), New York, Holt, Rinehart and Winston, 1985.

Duncan, L. D., Halsey, P. J., Handler, B. H., MacGregor, D. G., and Sparks, S. G., *Recommended CAI Approach for the NALDA System,* ORNL-6340, Oak Ridge National Laboratory, Oak Ridge, TN, 1987.

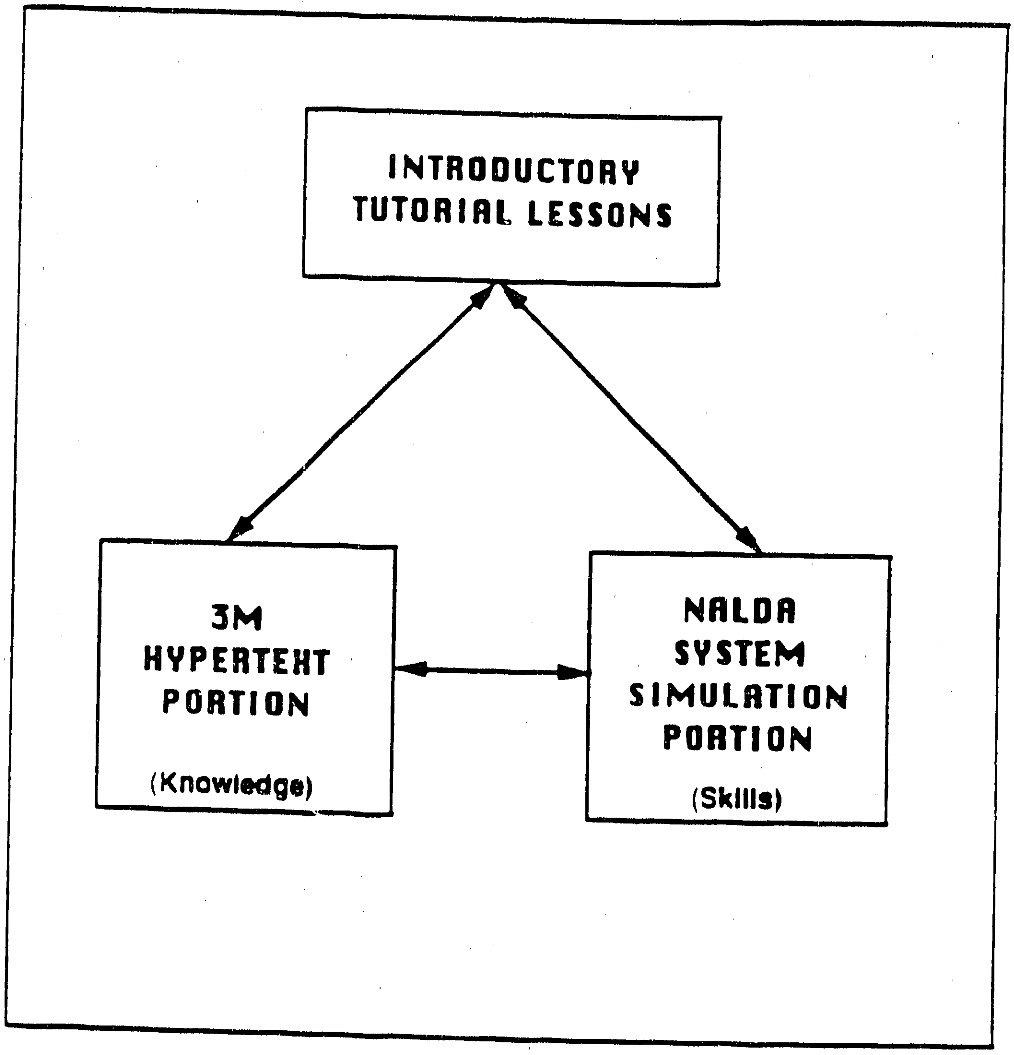
Handler, B. H., Bryant, R. A., Twitty, A. F., Duncan, L. D., Hallbick, A. M., Alvaro, D. R., Voorhees, V. M., and Moeller, J. D., *NALDA CAI Prototype - Phase II Final Report*, K/DSRD-71, Data Systems Research and Development, 1989.

Twitty, A. F., Handler, B. H., Duncan, L. D., Halsey, P. J., Bryant, R. A., Shaffer, K. E., Hallbick, A. M., and Alvaro, D. R., *Design Specifications for NALDA CAI - Phase II Interim Report*, ORNL/TM-10595, Oak Ridge National Laboratory, Oak Ridge, TN, 1987.

Appendix A

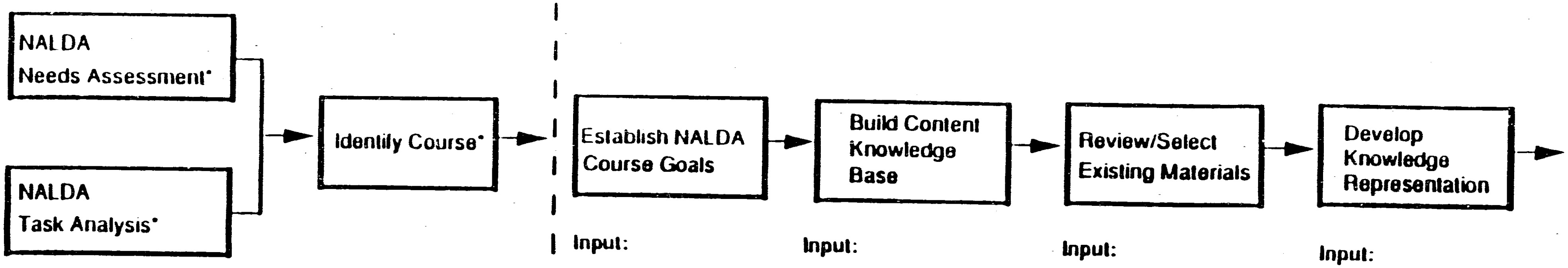
Overall NALDA CAI Model

OVERALL NALDA CAI MODEL



Appendix B

Advanced CAI Model



*Accomplished prior to CAI development

Establish NALDA Course Goals

Input:

1. Information from needs assessment, task analysis, discussions with job experts, and existing unit materials

Process:

1. Review data from needs assessments & task analysis
2. Review existing unit goals & unit materials
3. Discuss job tasks with experts
4. Draft unit goal statements
5. Verify unit goal statements

Output:

1. Listing of tangible unit goals

Build Content Knowledge Base

Input:

1. Course goals
2. Available resources
3. Existing courses
4. Content experts
5. Task analysis data

Process:

1. Read materials
2. Attend courses
3. Discuss with content experts
4. Review worker functions
5. Identify knowledge gaps
6. Annotate task analyses
7. Discuss with work team
8. Verify knowledge base

Output:

1. Increased knowledge of work team
2. Categorized source documents
3. Outline of knowledge base

Review/Select Existing Materials

Input:

1. Course goals
2. Outline of knowledge base
3. Task analysis data
4. Existing materials

Process:

1. Examine existing materials in light of course goals and knowledge base
2. Determine correspondence of materials with goals
3. Assess usefulness of materials in light of the content and audience
4. Select materials for use

Output:

1. Index of usable materials

Develop Knowledge Representation

Input:

1. Course goals
2. Outline of knowledge base

Process:

1. Conduct a procedural analysis of steps to be performed

Output:

1. Performance sequence

Analyze Course	Establish Course Objectives	Sequence Course Content	Establish Unit Goals	Analyze Units	Establish Unit Objectives
Input:	**Input:**	**Input:**	**Input:**	**Input:**	**Input:**
1. Course goals 2. Outline of knowledge base 3. Performance sequence	1. Course goals 2. Outline of knowledge base 3. Performance sequence 4. List of pre-requisite skills & supportive knowledge	1. Performance objectives 2. Performance sequence 3. Listing of pre-requisite skills & supportive knowl-edge for each major step (task)	1. Sequenced course objectives 2. Outline of knowledge base 3. Performance sequence	1. Unit goals 2. Outline of knowledge base 3 Performance sequence	1. Unit goals 2. Course goals & objectives
Process:	**Process:**	**Process:**	**Process:**	**Process:**	**Process:**
1. Categorize the type of performance for each major step (task) 2. Identify pre-requisite skills 3. Identify supportive knowledge	1. Identify the specific behavior at each step 2. Determine the criterion level 3. Specify important conditions	1. Examine the nature of the relationship among the objectives 2. Consider positive transfer	1. Break each course objective down into smaller parts 2. Group related parts	1. Categorize the type of performance for each unit goal 2. Identify pre-requisite skills 3. Identify supportive knowledge	1. Identify specific behavior for each unit goal 2. Determine the criteria level 3. Specify important conditions
Output:	**Output:**	**Output:**	**Output:**	**Output:**	**Output:**
1. Listing of pre-requisite skills & supportive knowledge for each major step (task)	1. Performance objectives for the course	1 Sequential ordering of course objectives	1. Unit goals	1. Listing of pre-requisite skills & supportive knowledge for each unit goal	1. Performance objectives for the unit

Sequence Unit Content	Establish Lesson Goals	Analyze Lesson	Establish Lesson Objectives	Sequence Lesson Content
Input:	**Input:**	**Input:**	**Input:**	**Input:**
1. Performance objectives for the course and unit 2. Performance sequence	1. Sequenced unit objectives 2. Outline of knowledge base 3. Performance sequence	1. Lesson goals 2. Outline of knowledge base 3. Performance sequence	1. Lesson goals 2. Unit goals & objectives 3. Course goals & objectives	1. Performance objectives for the course, unit & lesson 2. Performance sequence
Process:	**Process:**	**Process:**	**Process:**	**Process:**
1. Examine the nature of the relationships among the objectives 2. Consider positive transfer	1. Break each unit objective down into smaller parts 2. Group related parts	1. Categorize the type of performance for each lesson goal 2. Identify prerequisite skills 3. Identify supportive knowledge	1. Identify specific behavior for each lesson objective 2. Determine the criterion level 3. Specify important conditions	1. Examine the nature of the relationships among the objectives 2. Consider positive transfer
Output:	**Output:**	**Output:**	**Output:**	**Output:**
1. Sequential ordering of unit objectives	1. Lesson goals	1. Listing of prerequisite skills & supportive knowledge for each lesson goal	1. Performance objectives for the lesson	1. Sequential ordering of lesson objectives

Analyze Hardware & Software Capabilities

Input:

1. Identification of available hardware/software
2. Documentation of hardware/software
3. Access to hardware/software

Process:

1. Examine specifications
2. Try out hardware/software
3. Determine constraints of hardware/software in light of guidelines for advanced instructional model

Output:

1. Report on hardware/software analysis
2. Identified constraints

Select Programming & Development Tools

Input:

1. Hardware analysis
2. Software analysis

Process:

1. Review documentation
2. Discuss with programming team

Output:

1. Report on programming & development tools for CAI
2. Software development plan

Analyze User Capabilities

Input:

1. Information about end user

Process:

1. Identify knowledge level
2. Identify skill level
3. Identify work history
4. Identify education history

Output:

1. Assumptions about user capabilities

Review Course Content Outline and Knowledge Representations

Input:

1. Sequenced objectives
2. Listing of prerequisites, skills & supportive knowledge

Process:

1. Consider important instructional conditions
2. Determine how to present course content

Output:

1. Report on instructional strategy

Design Advanced Instructional Model for CAI

Input:

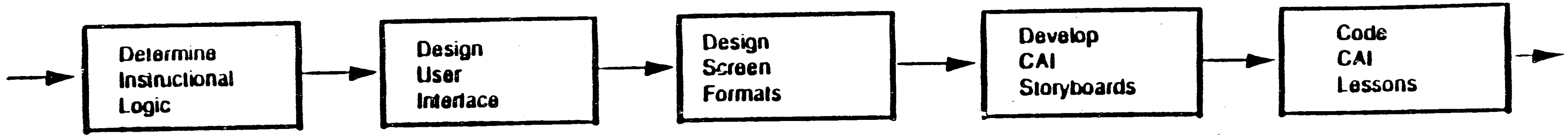
1. Guidelines for advanced instructional model
2. Hardware/software analysis
3. User capabilities
4. Instructional strategy

Process:

1. Examine guidelines for advanced instructional model in light of constraints
2. Identify & make trade-offs

Output:

1. Documentation of design for advanced instructional model



Determine Instructional Logic	Design User Interface	Design Screen Formats	Develop CAI Storyboards	Code CAI Lessons
Input: 1. Design for advanced instructional model 2. Course/unit/lesson content & sequence 3. Instructional strategy	**Input:** 1. NALDA phase 1 & 2 reports 2. Report on R&D literature 3. Report on state-of-the-art projects 4. Advanced instructional model 5. Flowcharts of instructional logic	**Input:** 1. NALDA phase 1 & 2 reports 2. Report on R&D literature 3. Report on state-of-the-art projects 4. Advanced instructional model 5. Flowcharts of instructional logic 6. User interface guidelines & standards	**Input:** 1. Screen design standards 2. User interface standards 3. Instructional logic flowcharts 4. Lesson objectives 5. Content knowledge base 6. Instructional strategy	**Input:** 1. Storyboard 2. Screen design standard 3. User interface standards 4. Programming & development tools 5. Code generator
Process: 1. Determine components to CAI lessons 2. Determine movement among & through components 3. Identify decision points in CAI lessons 4. Determine how decisions are to be made	**Process:** 1. Determine general guidelines for user interface 2. Identify interface points 3. Categorize types of interface points 4. Design user interface standards	**Process:** 1. Establis' general screen design standards 2. Categorize screen types 3. Establish specific standards for each screen type	**Process:** 1. Create lesson components 2. Determine how to communicate content 3. Lay out each screen	**Process:** 1. Generate code
Output: 1. Detailed flowcharts of instructional logic	**Output:** 1. Set of general guidelines for user interface 2. Set of specific standards	**Output:** 1. Set of general guidelines for screen design 2. Set of specific standards & templates	**Output:** 1. CAI storyboard (one page per screen) containing text, graphics, & sequencing information	**Output:** 1. Draft courseware

Debug CAI →

Input:

1. Storyboards
2. Draft courseware

Process:

1. Try out courseware
2. Identify programming bugs
3. Review coding

Output:

1. Debugged courseware

Field Test CAI Lessons →

Input:

1. Courseware
2. Information on intended users

Process:

1. Identify field test site
2. Identify individuals
3. ~~Complete individual~~ lessons
4. Test & debrief individuals

Output:

1. Data on achievement & acceptance of CAI

Analyze Field Test Data & Review CAI →

Input:

1. Field test data
2. Courseware

Process:

1. Identify weak points/ problem areas
2. Review & recode

Output:

1. Final field-ready courseware

Document the Instructional Content →

Input:

1. Courseware
2. Storyboards

Process:

1. Develop written guide for students
2. Develop written guide for course managers

Output:

1. Student guide
2. Course manager guide

Document the CAI Software

Input:

1. Courseware
2. Flowcharts
3. Programming notes

Process:

1. Describe the purpose of each segment of code
2. Document conventions used

Output:

1. Programmers' reference guide

Section 7

GLOSSARY

7. GLOSSARY

Analysis - The study of a functional area prior to implementing a new set of procedures (possibly automated).

Application Software - Programs that perform a specific function such as word processing, database retrieval, calculations (i.e., spreadsheets), graphics design, etc.

Artificial Intelligence (AI) - A process by which solutions to problems are found or constructed through the use of computer systems.

Attribute - An optional characteristic of a graphic entity, such as color, flashing, bold, reverse video that changes the appearance of that entity but does not change its basic nature or position on the screen.

Author - Individual or team responsible for designing and creating the instructional content of a course.

Authoring - Primarily, the process of designing and creating the instructional content of a course; may also involve design and creation of control mechanisms, user interface, record-keeping and reporting functions, etc.

Authoring Language - Specialized computer programming language designed to meet a common set of requirements for computer based instruction development.

Authoring System - Software used to create source files that contain commands and/or statements included in the authoring language.

Behavioral Psychology - The branch of psychology dealing with those aspects of organisms which can be observed, recorded, and studied in an effort to predict future behavior.

Box Score - A summary treatment used to tally differences between alternate methods in a research project.

Branching - The control of sequencing in a computerized program.

Chunking - Any series of displays that achieves the instructional objective of conveying a single concept or single thought.

Cognitive Psychology - The branch of psychology dealing with the mental processes involved in thinking and perception.

Computer Assisted (or aided) Instruction (CAI) - The use of a computer system as a tutor. The computer presents instruction to students using text and graphics to illustrate important points and allows students to interact with the computer to practice the skills being taught.

Computer Aided Learning (CAL) - Synonym for CAI.

Computer Based Instruction (CBI) - The use of interactive computers to enhance instruction, including both CAI and CMI.

Computer Based Training (CBT) - Synonym for CAI.

Computer Managed Instruction (CMI) - Using a computer system to control and/or monitor students' paths through instructional material by storing data on student performance and prescribing future instruction based on this data.

Concurrency - The ability to hold two or more programs in memory at one time. As applied to training, it is the ability of one program (training) to control another program (the application) by passing command and keyboard information to it.

Concurrent CAI - Computer aided instruction which not only teaches the student, but also allows the student to access and practice with the operational software system that is being taught; the two software systems (the CAI and the operational system) are both running concurrently and the student is able to interact with each via "toggling," pop-up or pull-down menus, and/or split screens.

Courseware - Application software designed to teach students how to use the computer system.

Courseware Authoring System - Software development package that allows the rapid development of CBI without the use of standard programming.

Criterion Screen - A screen which presents a question to the student. The student's response is then used to judge the student's level of understanding of the topic.

Cursor - An indicator on a computer screen that shows where the next character is to be typed by the user. The marker is usually a blinking square, line, or arrow.

Data - Any information used by a program or stored on a disk.

Data Bank - A collection of related databases.

Database - A structured collection of related data.

Declarative Knowledge - General knowledge to be used for reasoning, information about a subject rather than "how to perform a task.

Design - The activity of transforming a statement of what is required to be accomplished into a plan for implementing that requirement.

Development - The act or result of creating or refining a process or procedure.

Disk - Media, usually magnetic, used for storing data and/or computer programs.

Drill and Practice - Screens in the NALDA CAI that appear at the end of appropriate lessons to provide hands-on experience in applying the concepts taught within the lesson.

Electronic Page Turner - Instructional package which presents information but does not allow for interactivity and does not accommodate individual differences in students.

Electronic Publishing System (EPS) - An authoring package marketed by Intellisance Corporation that includes a proprietary programming language, XPL, which is used to create a series of programs that may be executed to display instructional material.

Embedded Training - Training that is an integral part of a computer-based system, thus allowing a user to move easily between performing work and receiving training on the system.

Expert Module - Component of the intelligent computer-assisted instruction (ICAI) system that contains the declarative, procedural, and heuristic knowledge of the domain expert.

Expert Programming Language (XPL) - An Intellisance Corporation proprietary programming language which is used to create a series of programs that may be executed to display instructional material.

Expert System - A computer program that is designed to emulate the reasoning and decision making processes of a human expert in some procedural domain.

Feedback - Messages generated in response to user input.

Function Keys - Special keyboard keys that perform predetermined functions rather than display characters on the screen. Their function is defined by a program and can be changed as desired. For example, students can use function keys to get help or to back up to a previous display.

Graphics Mode - A state of the video part of a computer/video system in which characters received from the computer part represent graphics commands or special characters to be displayed on the screen rather than standard text.

Hardware - The physical components (monitor, keyboard, disk drive, etc.) that make up a computer system.

Hypermedia - Allows creation of graphics, digital sound, and animation, in addition to text and data components, from within a development structure.

Hypertext - An automated mechanism for locating reference information based on user-driven associations. Hypertext implies a non-linear arrangement of information sources unconstrained by physical containers (pages, books, files).

Instructional Systems Development (ISD) - A conceptual framework for producing instruction by providing steps for designers to follow.

Intelligent Computer-Assisted Instruction (ICAI) - Computer based instructional systems which emphasize intelligent learning environments and intelligent tutoring systems and combine modules of simulation, expertise, teaching, and diagnostic-student modeling.

Intelligent Tutoring System (ITS) - An acronym for Intelligent Computer-Assisted Instruction (ICAI).

Interactivity - A two-way electronic transfer that involves responses, or interaction, from the user.

Lesson - A subset of a course unit that usually addresses only one or two instructional objectives. (Compare with unit.)

Lesson Logic - The control of the lesson flow in a CAI program through the use of computer programming.

Mainframe - A large computer with extensive memory, disk storage and many facilities; a centralized collection of computing resources (hardware and software) that supports multiple users.

Main Memory - Computer circuitry that is used to store data and/or programs being accessed by the user. Main memory is also referred to as Random Access Memory or RAM.

Menu - A screen display designed to present students with a number of fixed options and allow them to choose the option they desire.

Meta-Analysis - A technique for summarizing research by estimating the size of the differences, called the effect size (ES), between treatment groups.

Module - A portion of a software system that performs a specific function. In the CAI system, there are instructional modules, testing modules, reporting modules, etc.

Narrative - Non-quantitative report of trends in a series of studies.

On-Line Documentation - Reference information which is available on the computer while using a software package. Designed to help the user learn to use the system.

Operating System - A collection of programs to control and manage the resources that make up a computer system.

Portability - The ability to run a computer program on another computer hardware system or to transfer data and/or application programs to a new software release or operating system.

Prerequisite - An instructional unit whose objectives must be mastered before students are allowed to study the lesson in question.

Procedural Knowledge - The application of content knowledge, reasoning or "how to" perform a task rather than declarative information about a subject.

Program - As a noun, a set of instructions that direct a computer to perform some meaningful task. Such instructions are written in programming languages such as BASIC, COBOL, Tutor, etc. As a verb, the act of writing such instructions and storing them on a computer system.

Program Generator - A tool for automating the process of creating a computer program.

Programming - The act of creating or modifying a computer program; translating an idea or concept into a set of computer instructions.

Proof of Concept - Non-quantitative studies designed to demonstrate the effectiveness of a method by answering a series of questions.

Prototyping - Application development method composed of an iterative system design that involves the user in the design process. Functional systems are developed rapidly and presented to target application users who then suggest modifications to the system. These changes are incorporated into the next version of the prototype.

Response - The activity that results from stimulation.

Simulation - The imitative representation of one system's functions by means of the functioning of another system.

Software - The programs that make a computer do something.

Stimulus - Something that arouses or incites to activity.

Symbol - A visual entity that carries some specific meaning to students.

Syntax - The way in which words or terms are arranged to form a command.

Technology Transfer - The transition of expertise, usually from an individual or organization who designed and/or developed a process or tool, to someone who will use that process or tool.

Transportability - The ease with which a program that runs on one system can be made to run on another system of a different type.

Tutorial - A computerized instructional system which presents instruction before providing drill and practice.

Unit - A section of a course that usually addresses a small set of related instructional objectives.

Word Processor - A computer program used to enter and/or modify text.

Workstation - An intelligent terminal connected to a data processing or word processing network.

END

DATE FILMED

02 / 19 / 91