Navy Personnel Research and Development Center
San Diego, California 92152-6800 TR-91-4 January 1991

(2)

AD-A231 711

Low Cost Microcomputer Training Systems Project, Computer Based Educational Software System: Final Report

C. Douglas Wetzel
W. H. Wulfeck II

NPRDC-TR-91-4 January 1991

LOW COST MICROCOMPUTER TRAINING SYSTEMS PROJECT, COMPUTER BASED EDUCATIONAL SOFTWARE SYSTEM: FINAL REPORT

C. Douglas Wetzel
W. H. Wulfeck II

Reviewed by
Janet L. Dickieson

Approved and Released by
B. E. Bacon
Captain, U.S. Navy
Commanding Officer
and
R. C. Sorenson
Technical Director (Acting)

Navy Personnel Research and Development Center
San Diego, California 92152-6800

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
January 1991

3. REPORT TYPE AND DATE COVERED
Final FY82-FY90

4. TITLE AND SUBTITLE
Low Cost Microcomputer Training Systems Project,
Computer Based Educational Software System: Final Report

5. FUNDING NUMBERS
Program Element 0603720N
Work Unit Z-1772-ET002

6. AUTHOR(S)
C. Douglas Wetzel and W. H. Wulfeck II

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Navy Personnel Research and Development Center
San Diego, California 92152-6800

8. PERFORMING ORGANIZATION REPORT NUMBER
NPRDC-TR-91-4

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Chief of Naval Operations (OP-11), Navy Department,
Washington, DC 20350-2000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The Low Cost Microcomputer Based Training Systems project addressed the amount of development effort required to create computer-based instruction (CBI) by instructional developers, the proliferation of nontransportable machine-specific CBI software over incompatible hardware systems, and the opportunity to standardize previously successful CBI software using generative techniques. The project conducted assessments of Navy training patterns relative to CBI, developed the Computer Based Educational Software System (CBESS), and fielded CBESS at representative test sites. CBESS standardizes a set of CBI development and delivery tools in five subsystems which include general CBI programs, a management interface, and three specialized subsystems involving fact memorization, technical vocabulary, and equipment simulation. Instruction developed at test sites spanned several content areas, including threat fact memorization with a videodisc, study skills, mathematics, and electricity and was employed as a supplement to instructor resources for remediation, refresher training, self-study, and repetitive practice

14. SUBJECT TERMS
Computer based instruction, authoring, computer aided instruction, computer based training

15. NUMBER OF PAGES
88

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNLIMITED

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

FOREWORD

This work was funded as part of the Advanced Development project entitled Low Cost Micro-computer Training Systems (Program Element Number 0603720N, Work Unit Number Z-1772-ET002). The project was the result of an operational requirement originally promulgated by the Chief of Naval Operations (OP-987H, OP-01B7) and then subsequently supervised by OP-11.

The purpose of this research was to assess Navy requirements for computer-based instruction and develop computer based instruction application software for wide Navy application through tryouts at representative test sites. The results of the project are primarily intended for the Department of the Navy training community.

B.E. BACON
Captain, U.S. Navy
Commanding Officer

RICHARD C. SORENSON
Technical Director (Acting)

SUMMARY

Problem

This project was concerned with Navy training needs for authoring and delivering instruction on low-cost microcomputer-based training systems. The problems addressed were the amount of development effort required to create computer-based instruction (CBI) by instructional developers, the proliferation of nontransportable machine specific CBI software over incompatible hardware systems, and the opportunity to further develop and standardize promising ideas from previous research and development work in sophisticated generative approaches to CBI. When the project began in 1982, CBI often took the form of specialized computer programs with instructional content embedded in the programs themselves. CBI generally required extensive development time and high levels of computer programming expertise that exceeded that of most instructional developers. Incompatible hardware and operating systems created serious transportability problems such that instruction developed for one machine often would not run on others without expensive recoding. At the outset of the project, the risks involved turning ideas from previously developed unstandardized programs into easily usable technology, and risks associated with hardware and software engineering market forces leading to future standardization. Additionally, successive cost reductions in what is still an emerging field offered the technological opportunity to provide the Navy with automated instruction, remediation, and drill and practice to supplement instructor resources.

Purpose

The overall purpose of the project was to provide the Navy with automated tools for developing computer-based instruction. The project sought to standardize a set of computer-based instruction strategies into a system and to reduce the expertise required to produce instruction. This report describes the status of the project after these tools were developed and fielded at representative test sites.

Approach

The overall approach of the project involved three elements: (1) assessment of Navy instructional practices relevant to CBI, consisting of a survey of instructional managers and a tabulation of the frequency of various instructional objectives, (2) development of CBI software for wide Navy application which resulted in the Computer Based Educational Software System (CBESS), and (3) development of demonstration test-beds for the CBESS authoring and delivery systems that actualized Navy specific courseware at various Navy training sites.

Results and Conclusions

One standard system was created from a diverse set of software which had previously been prototyped in various programming languages for different hardware platforms using divergent standards. The resultant difficulty level of the system decreased

relative to the prototypes from which it was derived. This work resulted in formal authoring tools which moved the authoring of computer based instruction from the realm of programmers into that of instructional developers.

The CBESS developed consists of five subsystems: (1) the Computer Based Memorization System (CBMS), which is specialized for repetitive fact training and has been used with large threat databases and a videodisc; (2) the Equipment Problem Solving Trainer (EPST), which is specialized for equipment simulation in the context of locating and replacing faulty parts; (3) the Language Skills Computer Aided Instruction (LSCAI), which is specialized for technical vocabulary training; (4) a General Computer Based Instruction (GCBI) package, which is a flexible general purpose utility for creating unique interfaces and lesson sequences; and (5) instructional management programs, which provide a menu interface linking the lessons from the other four packages. CBMS, EPST, and LSCAI are specialized authoring facilities that reduce development time by assuming certain pre-configured instructional delivery strategies.

Authoring and student programs were designed to separate courseware from executable programs so that the system can be reused to create many new varieties of separate instructional courseware lesson files. The authoring programs use standardized self-contained editors that reduce the effort of instructional developers and now make the availability and capability to produce CBI more widespread. The skill required by the programs generally assumes some prior basic operating knowledge of computers and instructional design. Market forces during the project reduced the number of prominent standard computers, which reduced the need to recode CBI among hardware platforms. The programs were specifically adapted to Navy standard microcomputers and can be reconfigured over a range of hardware options, such as display cards and videodisc players. The system was formally documented in 18 user manuals and the government controls the source code and can update it with desired features in the future.

The CBESS was successfully used by developers in creating deployable instruction that now remains at various Navy training commands as a regularly used instructional media. Four of the CBESS packages were used in substantial development efforts and a catalog of instruction documents these finished products. The incremental development of the system was responsive to user needs through an ongoing program of modifications, updates, and user training. System modifications during field tests resulted in an increase in the ease of using the programs and an increased utility with newly added features. Software development records showed 43% of the modifications traceable to user suggestions, with 70% of those being related to interests in interface features.

The CBESS is applicable across many ratings and for many types of instructional content. The developed lessons and potential applications include: remediation, enhancement, refresher, reviews, initial primary instruction, repetitive drill and practice, self-study, and as a general supplement to instructor resources. The developed lessons generally addressed specific training objectives supplementing larger bodies of regular course material. The system was successful in the intended application environments as indicated by its regular use by students and instructional managers, its contribution to increased performance or reduced attrition, and by the desire of test-bed sites that it be continued or expanded, and supported in the future.

Several evaluations are reported. Surveys showed heavy emphasis in Navy training for fact and procedure type learning objectives, and course managers reported concerns with curriculum stability and student entering skills. Student performance results from several test sites and previous studies included higher progress test scores, fewer retests, less training time, reduced attrition, fewer set backs, and increased usage with material tailored to course quizzes and supplemented by a videodisc. One intensive study of the authoring process with the specialized LSCAI showed reasonable development times and actualized a decision matrix method for selecting courses that would most benefit from computerization.

Recommendations

The following recommendations are for OP-11 and the Navy education and training community:

1. Continuing life cycle management support should be given implementation attention to realize previous development investments in the government-controlled CBESS. The success of the project directly implies specific post-project maintenance to support the continued operational use of the system and developed instruction.

2. Support responsibilities should be assigned and funding should be sought from a broad base appropriate to the wide number of applicable ratings.

3. CBESS can be adopted as a standard to avoid proliferation of incompatible and nontransportable lessons. Exceptions to the use of government-controlled CBI software should be allowed for justified special capabilities. The implementation of CBESS should proceed at sites such as the CNET Model Schools program.

4. User support should be provided for distributing software, manuals, maintaining stock, and consulting that is tailored to the intended instructional development purpose of the software.

5. Routine software life cycle maintenance should be planned to continue the viability of the CBESS on new host devices and to preserve investments in previously developed CBI so that it can continue to be delivered and updated.

6. Existing instruction in the CBMS threat databases should not be allowed to go out of date and should be maintained and updated centrally because of its wide applicability.

7. The Navy should systematically guide computer-based instruction technology by fostering the support infrastructure required by an environment with inherent personnel rotations and loss of trained individuals. Instructors will remain consumers and need resource specialists similar to those that have evolved in civilian school systems or currently exist in audio-visual support specialists.

8. Computer-based instructional development efforts should employ decision criteria that consider student throughput, course stability, course importance, level and type of training objectives employed, potential of remediation to affect attrition or setbacks, management potentials such as supplementing instructor resources, and basing the selection of appropriate software tools on instructional requirements.

9. Computer-based instruction technology continues to evolve and the Navy should adapt the CBESS to new DoD portability standards and enhance these systems with further reductions in user skill requirements. Although authoring systems distance developers from low level programming, developing computer-based instruction still requires more expertise than does developing conventional instruction.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Problem and Background

This project addressed standardization needs for authoring and delivering instruction on low-cost microcomputer-based training systems. The problems addressed included the amount of effort required to create computer-based instruction (CBI) by instructional developers, the proliferation of nontransportable machine specific CBI software over incompatible hardware systems, and the technological opportunity to further develop and standardize promising ideas from previous exploratory work in sophisticated generative approaches to CBI. The original Operational Requirement for this project was established in Chief of Naval Operations Memorandum 102/63-80 of 28 April 1980, and the project began in October 1982.

At the time the project began, CBI generally required extensive development time and high levels of expertise exceeding that of most instructional developers. CBI often took the form of specialized computer programs, which sometimes had instructional content embedded in the programs themselves, and which required programming experience to modify. These "difficulty" issues could be addressed with "authoring" programs that create instruction by organizing many options in higher level interfaces such as menus or special keyword languages. Authoring programs enter instructional content and many presentation options into complex database formats for the user, in effect distancing the user from the tedium of many lower level details.

When the project began, many incompatible hardware systems, operating systems, and programming languages created serious transportability problems. Instruction developed for one machine often would not run on other machines and expensive recoding was required to deliver CBI on the variety of proliferating hardware platforms used at different sites. Transportability issues directly threaten investments in previous instructional development and create the need to adapt programs over hardware platforms while attempting to avoid modifications to the instruction itself. At this point, low-cost personal computers (PCs) had yet to become standardized or widely affordable.

Previous exploratory research work had identified several common types of instructional situations for which CBI programs had been prototyped. This work included general frame-based study management, simulation, and sophisticated generative approaches. Generative CBI involves programs that create instructional presentations at run time by using rules to assemble data-based instances that are then presented with templates. The success in fielding these prototypes indicated a potential usefulness if they were standardized for more widespread availability and transportability. The potential for this use depended upon developing common authoring interfaces systematized for standard hardware platforms.

These technological opportunities to improve the delivery of Navy instruction were further supported by projected successive cost reductions in computer technology. At the outset of the project the risks involved turning these ideas into easily usable technology, and risks associated with hardware and software engineering market forces leading to future standardization The ultimate benefit to the Navy was to provide more readily available automatic delivery of instruction that would supplement management and instructor resources. This benefit would apply to a wide range of occupational ratings with learning objectives appropriate to computerization in the areas of drill and practice, remediation, simulation, and general self-study.

Purpose

The overall purpose of the project was to provide the Navy with automated tools for developing computer-based instruction. The project sought to standardize a set of computer-based instruction strategies into a system and to reduce the expertise required to produce instruction. This report describes the status of the project after these tools were developed and fielded at representative test sites.

APPROACH

The overall approach of the project involved the following three major phases: (1) assessment of Navy instructional practices relevant to CBI, consisting of a survey of instructional managers and a tabulation of the frequency of various instructional objectives, (2) development of CBI software for wide Navy application which resulted in the Computer Based Educational Software System (CBESS), and (3) development of demonstration test-beds for the CBESS authoring and delivery systems that actualized Navy specific courseware at various Navy training sites. The major phases of the approach are further elaborated in Figure 1.

Assessment Surveys

Two assessments of Navy training problems and patterns were conducted early in the project in order to develop profiles allowing generalizations about the applicability of various CBI methods. One assessment used a questionnaire to survey Navy instructional managers and the other tabulated the frequency of actual training objectives in Navy courses. Very brief descriptions of this work are given below (Wetzel, Van Kekerix, & Wulfeck, 1987a, 1987b; Wetzel & Wulfeck, 1986).

On-site structured interviews obtained from senior instructors or course managers of 135 Navy schools were reported in Wetzel et al. (1987a), which documented numerous statistics on the time devoted to various instructional and testing methods. The course managers identified general administrative computer support for themselves as a first ranked priority (97%), reflecting the 1984 survey date when low-cost microcomputers were not generally widespread. About 27% of the courses nominated at least one module as being suitable for CBI. At that time about 12.6% of the sampled courses used some form of CBI (20% in A-schools and 5.6% in C- & F-schools), with most of these being in electronics-related schools (30%). Special problems were identified by the instructional managers with regard to student entering skills (33%), abilities in math (35%) abilities in reading (46%), curriculum stability (39%), and inadequate learning objectives (39%). A severe student "wait time" for access to laboratory equipment was reported by 13% of the

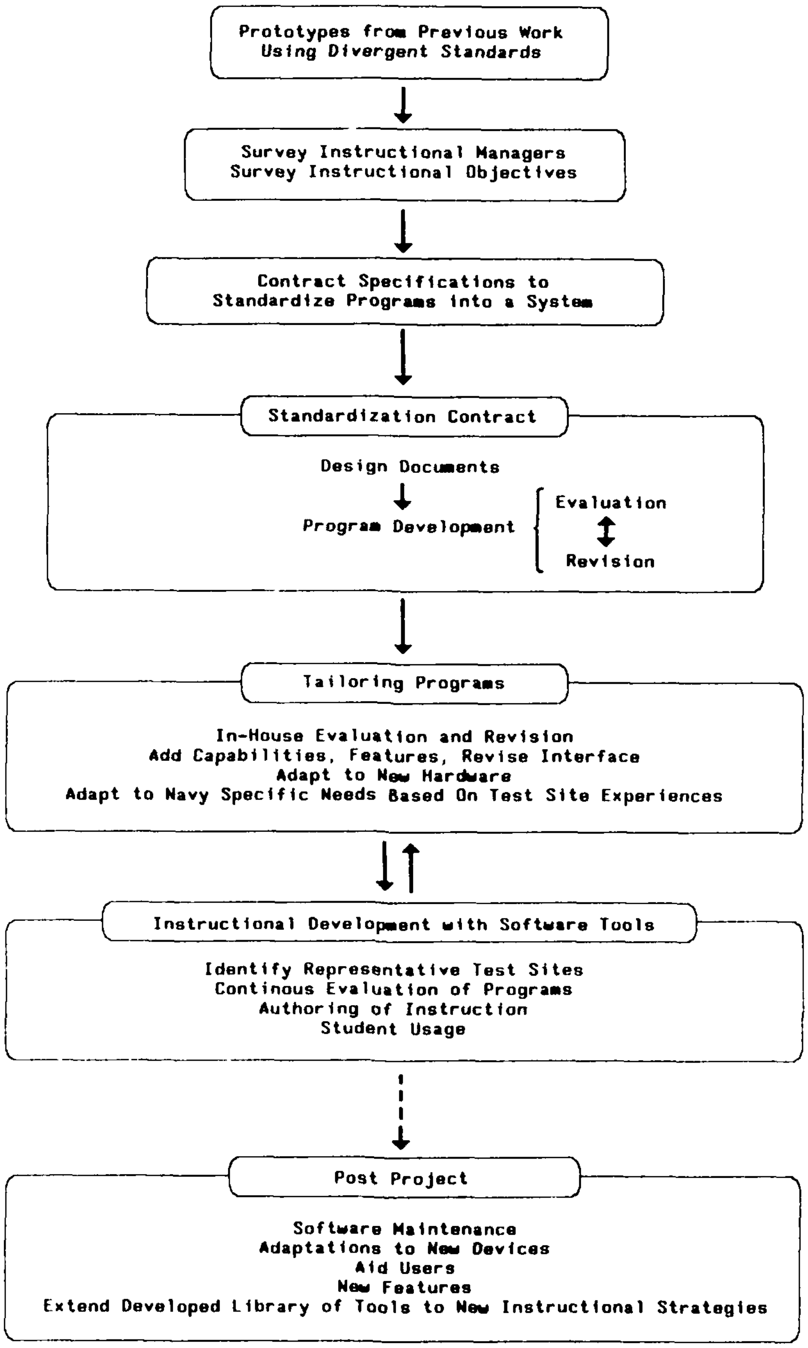

Figure 1. Project phases.

courses, with A-schools reporting a higher severity of this problem (20.3%) than C- & F-schools (7%). About 14% of the students were reported to not reach criterion on the first attempt of a module test, and the managers estimated that fast and slow students differed by about eight days in completing courses.

The relative frequency of different types of training requirements was determined through analysis of actual Navy training objectives in Wetzel et al. (1987b). Curriculum outlines and instructor guides from 246 Navy technical training courses yielded 34,373 training objectives. The Instructional Quality Inventory (IQI) (Ellis, Wulfeck & Fredericks, 1979) was used to classify the objectives according to: (1) what *task* the student must perform (i.e., "Remember" information or "Use" it to do something), (2) the type of information the student must learn (i.e., the type of instructional *content:* Fact, Category, Procedure, Rule & Principle), and (3) whether the objectives were "enabling" or "terminal" objectives. About 10% of the objectives were found to be major terminal objectives that were generally Use-tasks. The remaining 90% were enabling objectives, which prepare a student to acquire the terminal skills, and were most often Remember-tasks. Figure 2 shows that fact and procedure objectives were overwhelmingly the most frequent type used, with principle objectives ranked a much less frequently used third. The fact objectives were Remember-tasks and generally enabling objectives, while the procedure objectives were Use-tasks and most often terminal objectives. The introductory familiarization knowledge characteristic of entry level A-schools was evidenced by more fact objectives than were found in advanced C- & F-schools, while the skilled performance nature of advanced courses was shown by more procedure and principle objectives. Mechanical, operator, and team occupational groups showed predominate emphasis for Use-procedure objectives. Electrical and clerical/administrative groups most frequently employed Use-task objectives for procedures, rules and principles.

These two assessments provided profiles of common problems and training objectives as a background for the subsequent development work. Several of the programs developed later supplemented instructor resources in addressing the large number of enabling fact objectives with drill and practice, reviews of introductory or background material, and applications appropriate to remediation of deficient entering skills. The diversity and highly specific nature of procedures training suggested the need for a general facility with sufficient flexibility. The reports discussed basing CBI on the specific type of training objective, rather than applying it to entire curricula or all types of curricula. Some items in these surveys could be repeated in the future to gauge changes in Navy training practices over time.

Development of Computer-based Instruction Software

The second and largest phase of the project involved the development of CBI software for wide Navy application, which resulted in the Computer Based Educational Software System (CBESS). This phase involved developing specifications for CBI programs estimated to be successful in the past, developing the computer programs on contract, and post-contract enhancement of the programs.

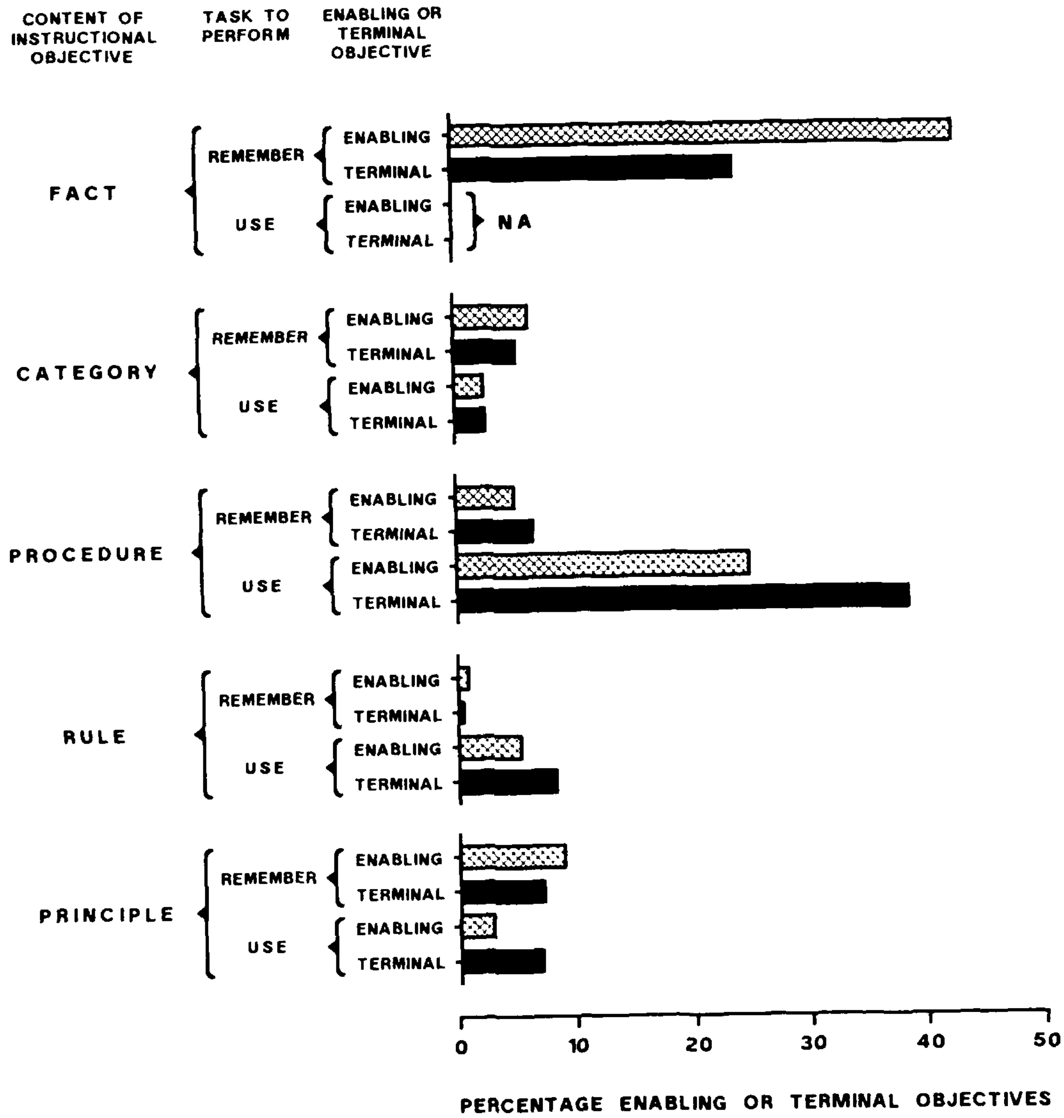

Figure 2. Training objectives by content and task.

Previous Work

The specifications for the CBESS programs grew out of four previous research and development lines of work in exploratory development (6.2) and advanced development (6.3) funding categories. First, fact memorization using threat databases were previously developed and tested at the Fleet Combat Training Center, Pacific with Tactical Action Officer students (Crawford & Hollan, 1983; McCandless, 1981). Second, automated remediation in vocabulary, reading, and language skills was the focus of several efforts concerned with deficient student entering skills (Wisher, 1980; Wisher & O'Hara, 1981; and recently summarized in Wisher, 1986). One implementation site resulting from this work is still in operation at the Operations Specialist (OS) A-school in Dam Neck, Virginia, and will be discussed in more detail later. Third, equipment problem solving and maintenance work had been done on specially configured hardware with software known as the Generalized Maintenance Trainer/Simulator (GMTS) (Rigney, Towne, King, & Moran, 1978; Rigney, Towne, Moran, & Mishler, 1980). Finally, a general study management system known as CAISMS was used in work reported by Van Kekerix, Wulfeck, & Montague (1982a, 1982b).

These previous four applications encompassed different programming languages and graphics standards, and ran under different operating systems on Terak,* Apple II, and specially configured hardware platforms. Only one of these configurations now has even a descendant in widely available products in the market place, and at the time other contemporary software products ran on an even wider set of configurations. These diverse configurations made it difficult to transport either the programs or previously developed instruction among the computers available at the time. A fortunate technological opportunity during the initial development of CBESS was the emergence of the IBM PC as a standard in the market place. Those conventions were later found in the Air Force/Navy "Desktop I/II/III" series of contracts that resulted in growing numbers of microcomputers in Navy commands.

Standardization Contract

The previous application programs noted above were identified for systemization in a standard set of software that was initially developed under contract with the University of Utah. The contract ran from December 1982 to April 1987 and the final cumulative cost with this contractor reached $1.5 million. The government obtained the raw program source code so that future modifications, enhancements, and programs could be built upon those libraries of software tools. The contractor developed an initial set of systematic software design documents in order to integrate the various requirements of the diverse applications included in the CBESS. An issue that emerged during this phase was the trade-off between the need to actually produce the software and the need to conduct additional planning to foresee later inconsistencies or problems. A lesson learned from this design document phase was that careful planning must have a cut-off point in order to permit actual coding to begin. Several such instances of this design/coding trade-off are noted below. During the initial contract, an outside consultant evaluated the

* Identification of specific equipment and software is for documentation only and does not imply endorsement.

programs at a cost of $22K (Halff, 1987a). This evaluation supported ongoing in-house evaluations of the intermediate program versions produced by the contractor.

The CBESS programs were developed with the C programming language because of its low level control and portability. The programs were constructed with embedded compilation statements allowing the production of both MS-DOS and UNIX operating system versions. Only the MS-DOS versions had significant graphic and video capabilities. Redesigning the programs into the CBESS included systemization of many lower level modules and lesson file formats so that they were common to all programs. An example of systematic design trade-offs involved pitting the economical maintenance of a small total number of common system modules against the local variances required in specific programs. Attempts to provide requested new features later in the project showed that conventions used in earlier common modules constrained the programming of new local variances so that changes had to be made cautiously to avoid unwanted side effects in other programs using the common module.

The programs were specially designed with student delivery programs separated from the authoring programs that instructional developers use to enter new instructional material. The courseware databases created were in turn separated from the student and authoring programs so that the programs could be reused again and again for new instances of instruction requiring different data. The authoring programs generally provide authors with menu-based selections of attributes which the program then translates into a compact non-ASCII format lesson file. The various lesson files share many format conventions so that elements may be copied among the authoring programs. The editing programs provide features for graphics, videodisc images, windows, text, windows, and nonduplicative linking capabilities that provide economy in file sizes.

The CBESS programs were designed to be reconfigured over various devices by editing a conventional ASCII file read by the programs on start-up. This feature avoided the need to supply different versions of the programs for each hardware device configuration. During the contract, various hardware standards emerged (e.g., graphics boards) and these were incorporated as potential significant usage became apparent. Such hardware standards will continue to emerge. It is a given feature of software life-cycle maintenance that changes will have to be made in the future to allow the programs and previously developed instruction to operate on new equipment. Ongoing work in the Department of Defense (DoD) Courseware Portability Standards project (PORTCO) offers potential for future standardization (Thomason, Van de Wetering, & Booth, 1990). For example, device drivers can be separated from programs in a standardized way that could reduce RAM memory requirements and provide greater courseware portability and other efficiencies.

The initial user manuals produced on the contract descended from the initial program design documents. These manuals were revised and supplemented during the remainder of the project to reflect new program features. The state of the programs and manuals resulting from the initial contract were judged to require additional development in order to provide finished products to users with greater reliability, enhanced functionality, and user interface refinements. The end of this contract lead to the initiation of other contractual efforts to complete the development.

Post-contract In-house Work

Following the initial conversion contract, three years of additional work was performed in-house to eliminate problematic program bugs, rework many interface conventions, and enhance the programs with new features. During this period, a significant number of users were provided CBESS for field testing. The efforts of NPRDC researchers were supplemented by a local contractor (Systems Engineering Associates) and local contract work-study students. The local contractor costs were $488K for work by professional programmers. Work-study students provided additional support in programming, software testing, and curriculum development (at a cost of $415K during the entire span of the project).

A number of significant new features were added during this period. The object-oriented graphics capability embedded in the CBESS programs was supplemented with an alternative feature allowing the use of scanned bitmapped images. Other additions included various colored text objects, increased program execution speed, enhanced answer analysis syntax, and keywords to allow greater precision in asking questions. A significant amount of effort was devoted to increasing the ease with which the programs could be used and to enhancing the interface of the student and authoring programs. The general CBI package was significantly reworked to allow the ability to create unique interfaces, branching schemes, and variable manipulation. A running record of these in-house software changes was systematically maintained and the results of an analysis of these changes are reported in the evaluation section.

Description of CBESS Programs

This section describes the computer-based instruction programs that were standardized into the current Computer Based Educational Software System (CBESS). Figure 3 is a simplified overall system view of the CBESS which gives many of the authoring and student program names in relation to courseware lesson files and other configuration files. The CBESS currently consists of the following five major elements, with the first three being specialized for certain instructional strategies and the last two being general in nature:

1. Computer Based Memorization System (CBMS)
2. Equipment Problem Solving Trainer (EPST)
3. Language Skills Computer Aided Instruction (LSCAI)
4. General CBI package (GCBI)
5. Instructional management programs

Computer Based Memorization System (CBMS)

The CBMS programs use a semantic network to represent large bodies of facts to be memorized through database browsing and game programs that quiz both facts and picture recognition (see Figures 4 and 5). Figure 4 represents a semantic network, which consists of a tree structure subcategorizing items, the assignment of attribute descriptions to the items, and automatic cross-referencing when one item describes another. CBMS

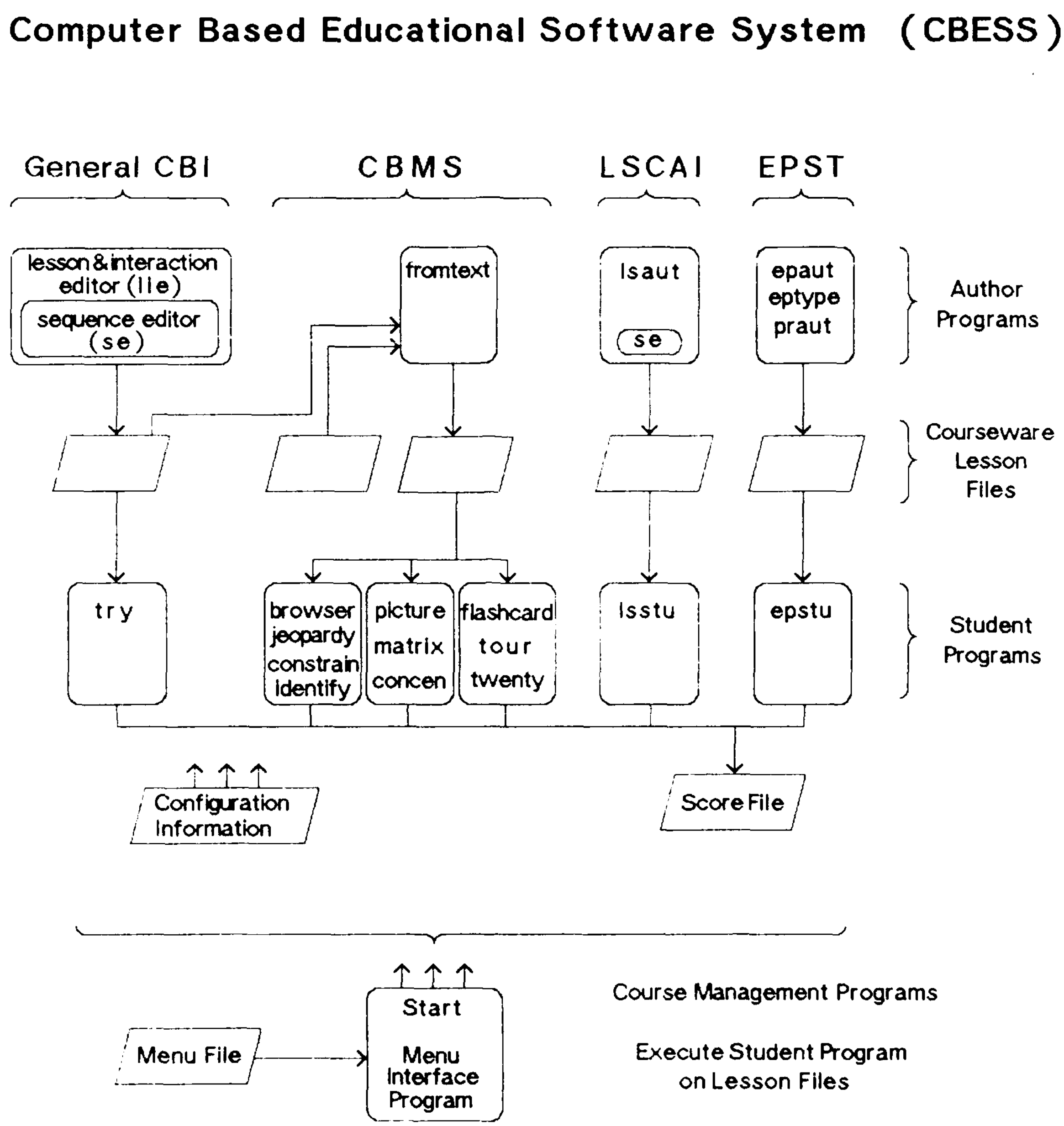

Figure 3. System diagram for the CBESS authoring, student and management programs, and lesson files.

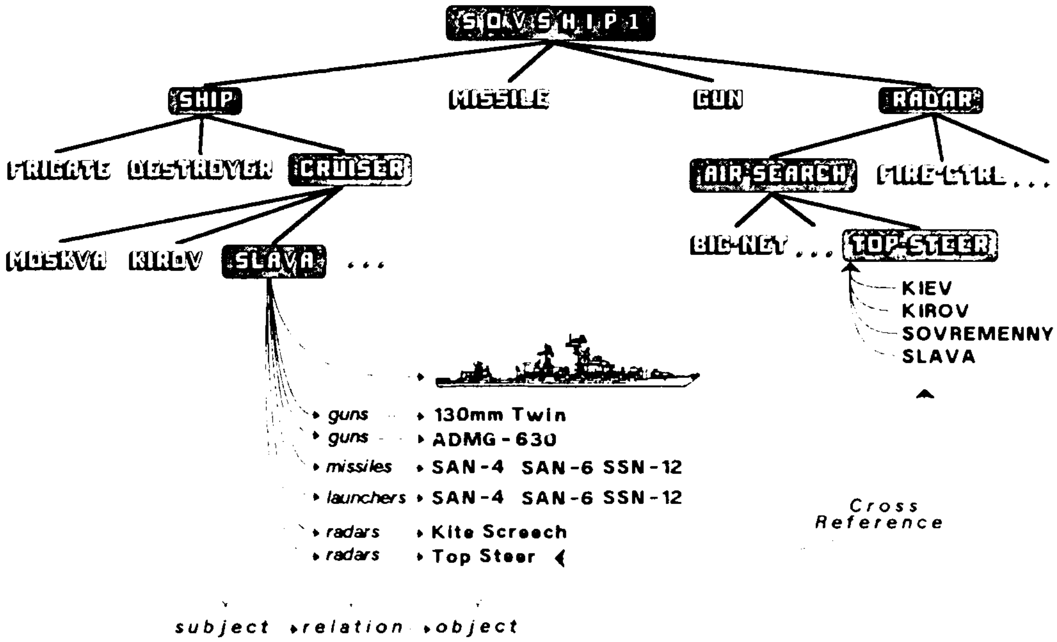

Figure 4. Example of semantic network for CBMS.

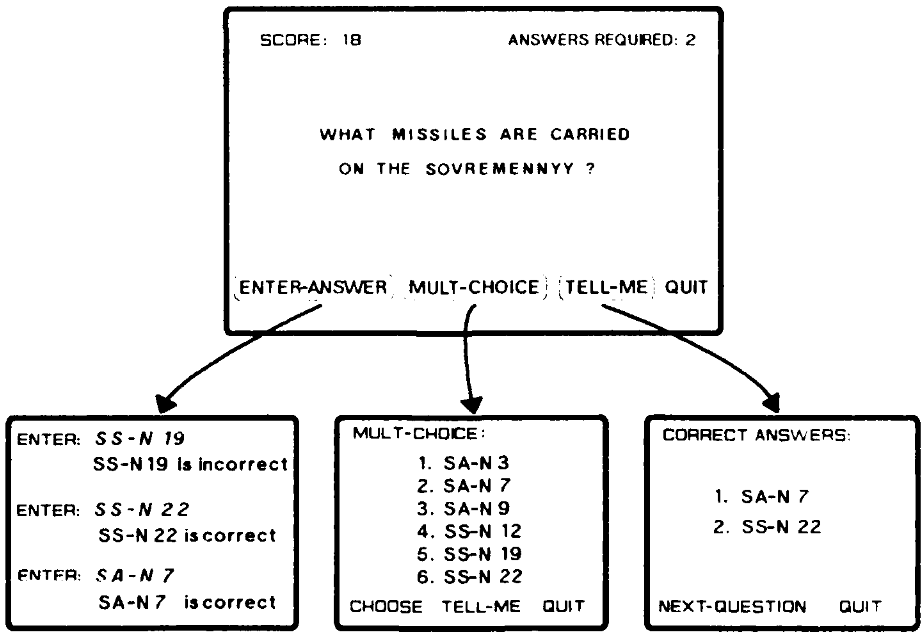

Figure 5. Three instructional strategies of the CBMS Flashcard game.

uses a generative approach in which questions are generated on-the-fly from database assertions as the student programs run. That is, large numbers of question screens do not have to be laboriously made up ahead of time, as would be the approach if commercially available packages were used. The CBMS programs embody several instructional strategies, and many have a built-in scheme in which students can answer questions by one of the following methods: (1) prompted recall where answers are typed in, (2) multiple choice recognition from a list of alternatives, or (3) requesting the program to "tell-me" the answer. Figure 5 illustrates a student's potential selection of these three strategies with the FLASHCARD game. Other student programs use variations in the interface to query information via different game board formats. CBMS consists of 10 student-execute programs and two authoring programs for translating conventional text files into databases, and may be used with one of two management interface programs. Graphics and video editing are accomplished by using an editor from the GCBI package (discussed below).

Equipment Problem Solving Trainer (EPST)

The EPST programs provide a simulator designed to reduce reliance on the use of actual equipment trainers for learning to operate, maintain, and troubleshoot malfunctions. EPST provides simulations of equipment containing problems presented to students in which faulty parts are to be discovered by making tests and by replacing parts until the equipment is functioning. Figure 6 illustrates these functions of EPST. The EPST is primarily a simulator and has minimal tutorial facilities for training procedural steps. EPST consists of three menu-based authoring programs and a student-execute program.

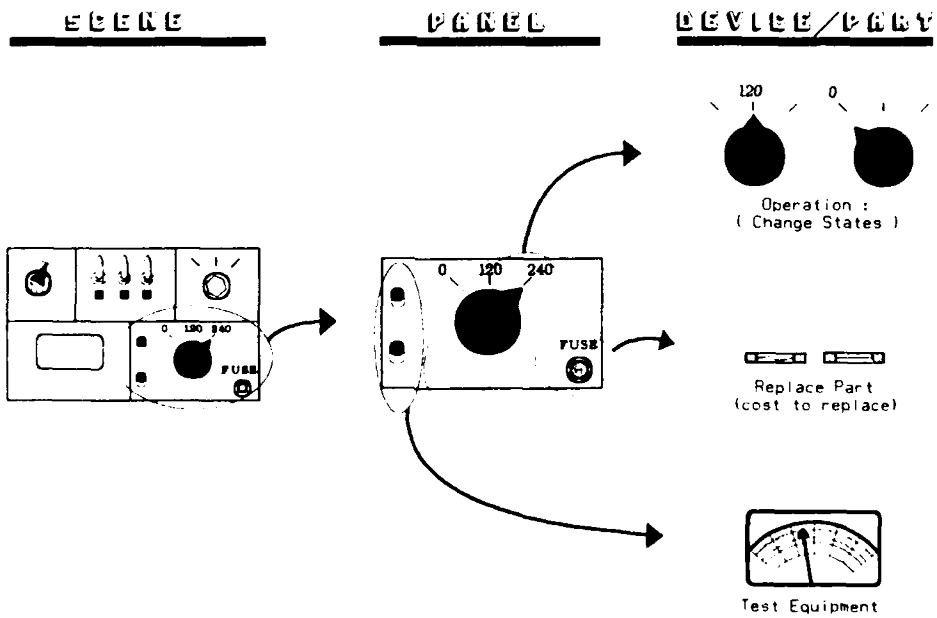

Figure 6. Equipment Problem Solving Trainer (EPST).

Language Skills Computer Aided Instruction (LSCAI)

The LSCAI programs provide training in general and technical vocabulary, and in reading through exercises with words and their definitions. The LSCAI uses a generative approach for many of the student activities in which to-be-learned material is presented from a database consisting of the words and their definitions. Thus, once words and their definitions have been entered by a developer, the student program will automatically construct and present activities such as definition review, multiple choice, true/false, spelling, matching, associated words, and building a definition phrase by phrase (see Figure 7). With input of additional unique information, the programs will also present a linear instructional sequence, feedback for multiple choice and true/false activities, a "cloze paragraph" with missing words to be completed, and a graphic labeling activity for identifying the parts of a drawing or video. The nine types of testing activities available in LSCAI can be selectively activated in various combinations to provide flexibility in the type of instruction delivered. LSCAI consists of a menu-based authoring program and a student-execute program. The left panel of Figure 8 illustrates the word definition database. The middle panel illustrates the naming of words to be included from the database and the manual creation of data for three learning activities that cannot be automatically generated. The right panel shows the learning activities available in the student programs and their correspondence to the exercise and word data.

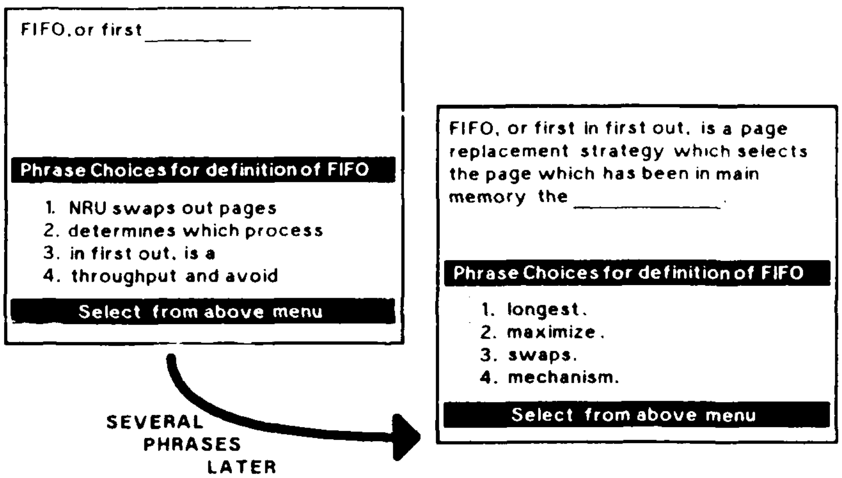

Figure 7. LSCAI definition building activity.

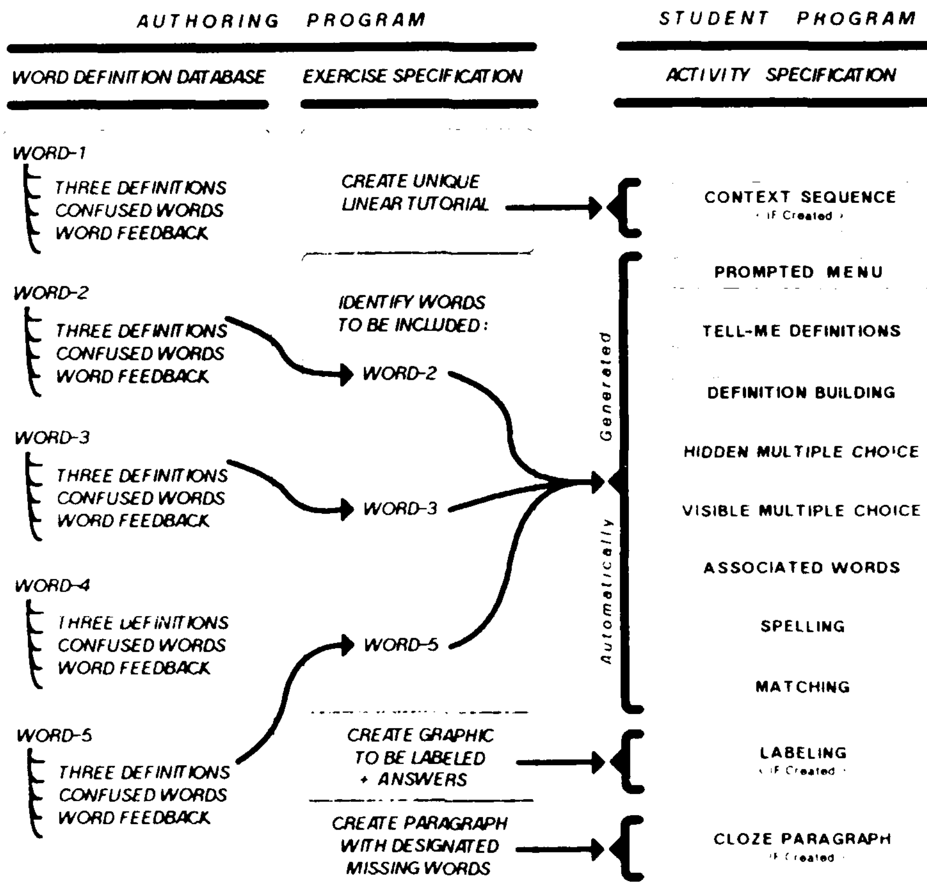

Figure 8. LSCAI authoring and student programs.

General CBI Package (GCBI)

The General CBI package allows presentation of linear displays, asking of questions, and branching and lesson control. The linear displays are known as "sequences" and are screens consisting of two types of graphics, text lines, windows, menus, laser videodisc screens. Questions or "interactions" are asked via 10 templates that provide for different answer input modes such as typing textual answers, selecting from various types of menus, pointing to screen positions with a mouse, and selecting from command lines. The question templates have built-in prefix, feedback, scoring, and answer analysis options. Conditional lesson flow of these screens and questions is provided by a presentation and branching meta-language with a built-in error checking parser (known as "control and computation frames"). These components are incorporated in the Lesson and Interaction Editor (LIE) program illustrated in Figure 9. The GCBI package allows the developer to determine instructional strategies, in contrast to the other packages which have predetermined strategies. The GCBI package is preferred for creating unique instructional strategies requiring greater flexibility in screen design and control over lesson flow. However, such unique presentations require more effort on the part of the developer than the predetermined strategies found in the LSCAI and CBMS. The GCBI package consists of two menu-based authoring programs and a student-execute program. The lower levels of the CBESS GCBI package are component building blocks used in part by the CBMS, LSCAI, and EPST packages. They use graphics, video, answer analysis, and windowing modules, many of which are incorporated in a component known as the Sequence Editor (SE).

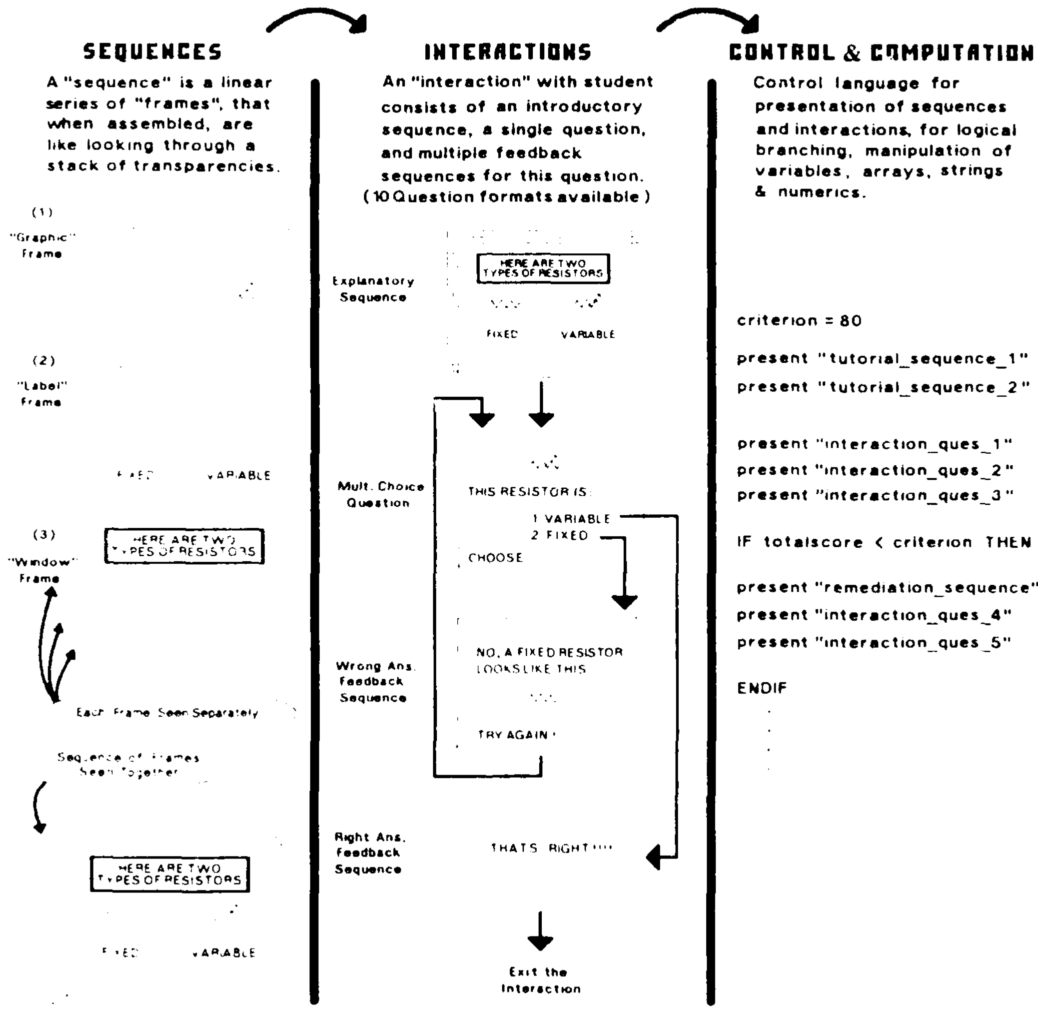

Figure 9. Components of the GCBI Lesson and Interaction Editor (LIE).

Instructional Management Programs

A collection of instructional management programs is used in conjunction with the other four packages. The START program provides a menu interface that allows students to select among many different lessons from the CBMS, EPST, LSCAI, and GCBI programs. The program was a direct result of test site experiences indicating the need for a unified interface that avoided extensive instructions to students on how to execute many different applications. A conventional text editor is used to create menus via a simplified keyword syntax specifying the menu choices and their actions. Another related management program controls student advancement in linear lessons if passing criteria have been achieved. Other miscellaneous programs are various utilities, such as score file management and summary programs.

Target Hardware and User Manuals

The CBESS was targeted for a fully configured Navy Standard Microcomputer such as the Zenith Z-248, which is widespread at many Navy commands. CBESS requires a MS-DOS operating system microcomputer, 640K RAM, an internal hard disk, and at least an EGA resolution video card and monitor. The CBESS may optionally use a mouse pointing device, six different videodisc players, and two alternative video overlay cards to permit the combination of videodisc images on the same screen with text and graphics.

A total of 18 user manuals for the CBESS were developed during the project. The total page count for all the manuals is approximately 1690 pages. These manuals are listed separately in Appendix A.

Generative Features of CBESS

Several of the CBESS programs employ "generative" CBI techniques that distinguish the programs from many conventional CBI authoring packages (cf., Wetzel, 1990). Generative CBI techniques involve new instances of instruction being generated from components not previously assembled in their complete finally delivered form. That is, the generative CBI programs produce output determined at run-time rather than simply presenting completely elaborated previous screens. This is accomplished by using a standardized kind of template which contains "dynamic placeholder" slots in which each new instance is inserted. Figure 10 illustrates the use of sentence templates in the CBMS for transforming database information into presentations to students. Templates for question and answer frames are found in some authoring packages (including CBESS) to save steps in creating instruction with a number of similar frame sequences. These partially completed templates are duplicated over and over in order to fill in *all* pieces of information unique to each. A generative CBI approach advances beyond this elaborative application by generating new instances for a *single* template used again and again. An algorithm accesses instructional content from a database and each new instance is inserted into dynamic placeholder slots in the template, which are prearranged spots for answers, text, or graphics. In some cases, the difficulty of creating such instruction is reduced because the templates, screen design, answer analysis, and presentation algorithm have been pre-defined. To the extent that they are appropriate to the desired instruction, the "rough edges" of the products created by less experienced authors may be reduced. Generative CBI

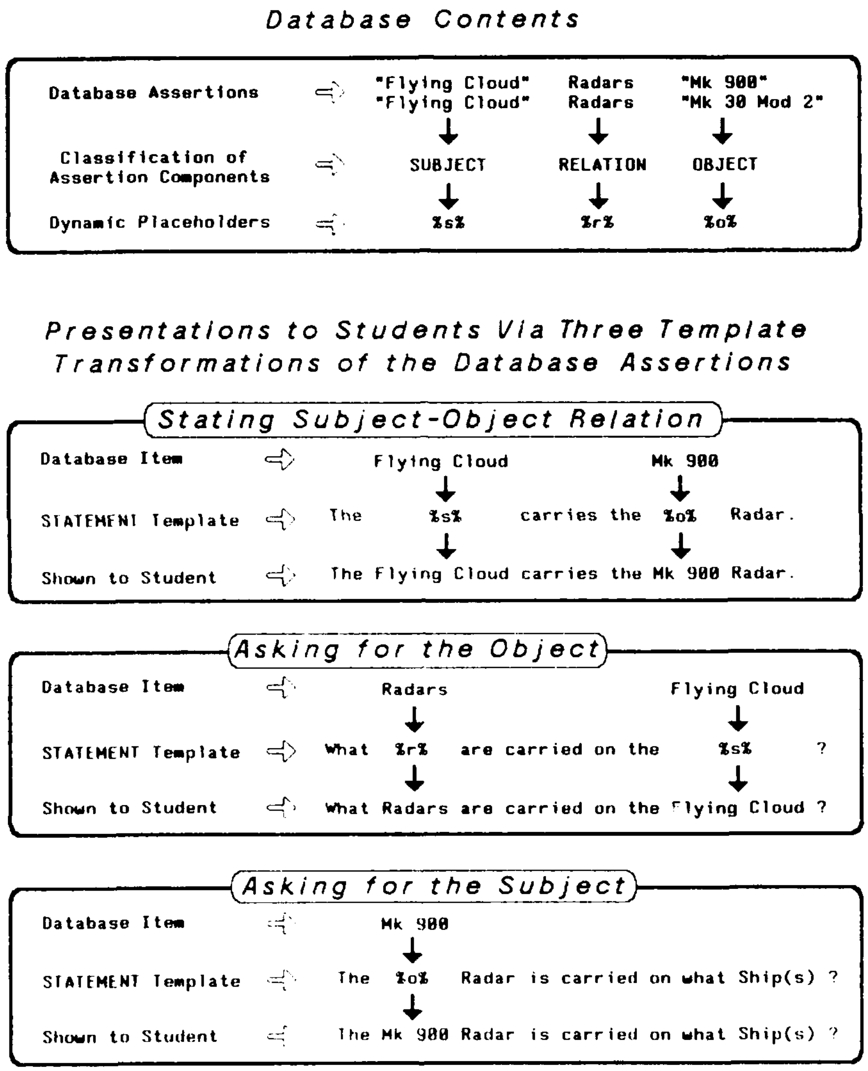

Figure 10. Transformation of CBMS database assertions into presentations to students.

techniques are an intermediate step between conventional frame-based instruction and more sophisticated artificial intelligence techniques (cf. Kearsley, 1987), which may be less manageable for users with little programming expertise.

The generative technique generally depends upon some degree of similarity from instance to instance in the features of the interface for the tutorial, question, answer, and feedback. Thus, generative CBI specializes the programs by standardizing them for common or routine instructional situations. Factors favoring routine standardization are those requiring little author input or overhead and depend upon whether it is possible to have a standardized student interface, question and answer database, feedback, tutorial, and method of process control. Even when these standard conditions exist, the user must still enter the text of a question and the correct answer, and create a tutorial. Factors not favoring standardization and demanding unique creation are the specification of incorrect answers, alternative correct answers, unique feedback, nonlinear process control for branching, and storing and manipulating variables.

Table 1 shows some of these standardization factors for the CBESS programs. The CBMS is almost completely a generative CBI application, and the LSCAI and GCBI packages employ the technique in some situations and conventional frame-by-frame creation techniques in others. CBMS standardizes all of the factors shown in specializing for repetitive quizzing on facts and images, and consequently is more efficient in generating hundreds of questions from a database than is manual elaboration of each question. One CBMS program (TOUR) provides a minimal tutorial in the sense that the facts to be quizzed are listed just prior to the quiz (see Figure 11). The LSCAI also standardizes many of the features, but provides a unique tutorial as a preface to the testing/learning activities, allows general feedback, provides two free-form testing activities, and offers some process control in selecting the combination of activities. The GCBI package provides free-form templates with slots in which any prearranged content could be inserted for the question, answer analysis, and feedback (illustrated in Figure 12). The price for this flexibility is a greater authoring overhead, and standardization is offered only in the sense that a selection of predefined question templates is available. All of the CBESS programs are reusable for new instructional content and they vary to the extent that the content is cast simply as a database for a standard student interface or is configurable in unique presentations. In general practice, a user might employ a combination of the CBESS packages. For example, repetitive practice and limited tutorials can be created *with one package* (e.g., CBMS or LSCAI), unique strategies can be created with the GCBI package, and all these applications can then be linked together as menu choices with the management interface.

Table 1. Degree of Standardization in CBESS Programs

Name of Application	Student Interface	Ques-Ans Database	Tutorial	Feedback	Process Control
CBMS	Standard	Standard	None/ Minimal	Standard	Strict Algorithm
EPST	Standard/ View Equip.	Simulation	None	State Change	Simulation
LSCAI	Standard	Standard	Preface	Minor Variability	Some Control
GCBI	Unique	Unique	Unique	Unique	Unique
START	Standard	Run Other Programs	Not Applicable	Not Applicable	Run Other Programs

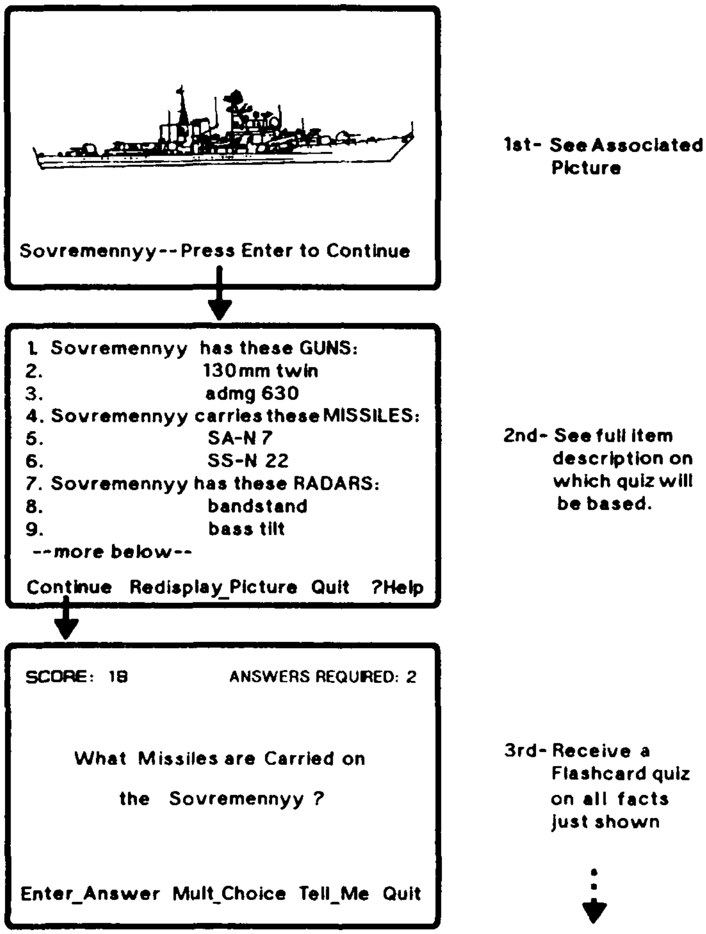

Figure 11. CBMS TOUR game presents information before quizzes.

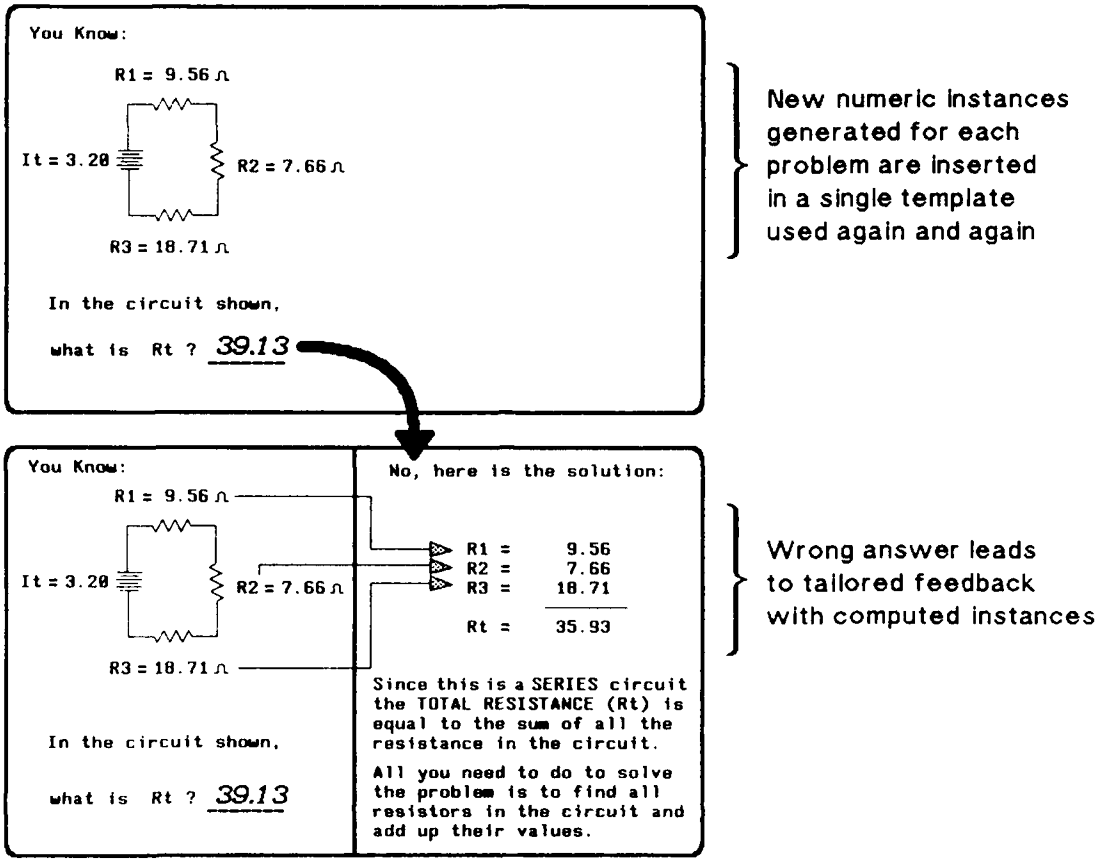

Figure 12. Example of generative CBI with the GCBI program.

Demonstration Test-beds

The third major phase of the project involved the development of demonstration test-beds for the CBESS authoring and delivery systems. Many of these efforts actualized specific Navy courseware at various Navy training sites. The test sites were a diverse set involving varying amounts of effort and actual instructional development. Emphasizing this variability, the sites might be categorized as: (1) major or full sites involving continuing interaction with the school or development site on a regular basis; (2) minor, transitory, or demonstration sites involving little or no instructional development, providing instruction developed at other sites or a demonstration system, or involving a transitory relationship; (3) mailed distribution of previously developed software and courseware; (4) emerging potential sites. The following section describes the details of CBESS activities at these sites.

TEST SITES AND INSTRUCTIONAL DEVELOPMENT

CBESS was fielded at a diverse set of test sites to evaluate the developed programs and to infuse it into actual instructional development efforts. This section describes the activities and the instruction developed at each site as well as several informal sites and potential applications. A later section details relevant evaluation results. The test sites are listed below:

- Navy and Marine Corps Intelligence Training Center (NMITC), Dam Neck, VA.
- Fleet Combat Training Center Pacific (FCTCPAC), San Diego, CA.
- U.S. Army Aviation Research and Development (R & D) Facility, Ft. Rucker, AL.
- Naval Air Station (NAS), Oceana, VA.
- Chief of Naval Technical Training (CNTECHTRA) and Naval Air Technical Training Center (NATTC), Millington, TN.
- Naval Construction Training Centers.
- Electrician's Mate (EM) A-school, Naval Training Center (NTC), Great Lakes, IL.
- Interior Communications Electrician (IC) A-school, San Diego and Naval Technical Training Support Group (NTTSG), Naval Training Center, San Diego.
- Naval Air Technical Training Center (NATTC), Lakehurst, NJ.
- Waterfront Trainers of the Chief of Naval Education and Training (CNET) and Naval Education and Training Program Management Support Activity (NETPMSA).
- Mail distribution, informal sites, and potential applications.

1. NMITC Dam Neck. The CBMS is regularly used at NMITC in support of Naval Intelligence Officer and Enlisted Intelligence Specialist courses. The CBMS is used for threat memorization practice on large numbers of facts about Soviet naval platforms and for recognition practice. NMITC has used the CBMS since 1986, shortly after the school was newly created by consolidating training formerly distributed at four other sites. The CBMS databases initially fielded included only line drawings for recognition purposes. An Army videodisc from Ft. Rucker (showing helicopters, AA-guns, SAMs, and tanks) was installed for an interim period while a Navy videodisc was being prepared at NPRDC. The evolution of this work culminated in the development of an unclassified laser videodisc on Soviet platforms, which was installed in August of 1989. This videodisc contains still and motion pictures, primarily of Soviet ships and their superstructures, radars, guns and launchers; Soviet aircraft; and Soviet submarines and their superstructures. It also contains one still image of each U.S. platform. CBMS shows these images, which allows students to browse through information and either take picture quizzes or receive a large number of factual questions about the subject platforms (e.g., "what missiles [or guns, or radars] does the Slava carry?", or "what ships carry the SS-N-22?"). The delivery systems were initially located in two classrooms, and later relocated to the heavily used library to provide greater access to students from all classes. Student users of CBMS have varying entry levels of knowledge and are self-selected based on their own judgment of a need for adjunct practice to supplement their courses.

The databases currently in use were constructed from unclassified sources to include the platforms being taught at NMITC. Table 2 lists these databases, which serve a varied user community since they are often mailed to individuals at other commands upon request. Instructional development work was performed by NPRDC personnel, with NMITC providing subject matter expertise and deployment support via an Automated Data Processing (ADP) officer and then an officer-in-charge of the library. Numerous changes to the CBMS and the databases resulted from comments of NMITC users. For example, a new CBMS program named TOUR was created to address the needs of users unfamiliar with a new content area (see Figure 11). Recurrent database updates were not assumed by NMITC personnel because of the unavailability of instructional developers. These platform databases reflect a rapidly changing set of facts requiring updates, in contrast to more static instructional content. As a reflection of the success of the CBMS and a concern with the future update of the databases, NMITC sent a letter to OP-092 via CNET in November 1989 requesting maintenance support and wider implementation of the system.

2. FCTCPAC San Diego. The Tactical Action Officer (TAO) course at FCTCPAC is a six-week course taught six times a year. Portions of the TAO course involve memorizing a large number of facts about Soviet and U.S. platforms from printed "threat matrices". The CBMS is regularly used in the FCTCPAC TAO course for threat memorization training as a practice adjunct to the course rather than as the primary source of instruction. At the start of each course, an instructor introduces students to the system as an automated alternative to creating their own flash card notes for rehearsing the large number of facts.

In February 1988, the initial computer-based training was provided for Soviet ships and was regularly used by the course instructor in a separate lab session. In October 1989, the system was upgraded with a videodisc containing motion and still images of the platforms. In December 1989, the system was upgraded to its final classified form on special removable hard disks for all types of platforms. The CBMS databases named in Table 3 were converted to classified form in order to address complaints that they should contain exactly the same material on which students are tested. Instructors participated extensively in database development as subject matter experts, but database development was performed by NPRDC because instructional developers at FCTCPAC were unavailable. These efforts provide a "model" hardware suite for potential export of the system to any other sites teaching this particular classified content. Relevant potential sites are a second TAO course taught at FCTCLANT Dam Neck, VA and the Surface Warfare Officers School (SWOS) at Newport, RI. Since this information can change as often as four times a year, maintenance of these databases for one site could benefit other sites. The overall success of the FCTCPAC application is indicated by its regular incorporation as part of FCTCPAC TAO course and by usage frequency data.

Table 2

Unclassified CBMS Platforms Threat Databases

Database Type & Name		Number of Items in Database			
Graphic or Text	Video Disc	Questions	Answers	Pictures	Database Description
sovship1	sovshipa	247	818	117 *	Soviet cruisers, destroyers & frigates.
sovship2	sovshipb	215	485	87 *	Soviet amphibious, intelligence, minesweeper, support/auxiliary, patrol craft.
sovsub	sovsubv	288	656	59 *	Soviet submarines.
sovair	sovairv	584	939	85	Soviet aircraft.
usship	usshipv	283	665	54	U.S. ships & submarines.
usair	usairv	287	395	23	U.S. aircraft.
natoair	------	181	274	0	NATO aircraft. Fact quiz only.
med	------	378	522	0	Mediterranean nation platforms. Fact quiz only.
pergulf	------	452	578	0	Persian Gulf nation platforms. Fact quiz only.
westpac	------	308	413	0	West Pacific nation platforms. Fact quiz only.
armfact0	armfact	324	720	51	Army helicopter, tank, AA-gun, SAMs of various countries stressing facts.
armpics0	armpics	308	700	50	Army helicopter, tank, AA-gun, SAMs of various countries stressing quick recognition.

* Line drawings of platforms and equipment are found in both versions of these databases.

Table 3

CBMS Platforms Threat Databases for FCTCPAC TAO Course

Database Type & Name		
GRAPHIC or TEXT	VIDEO DISC	Database Description
taoship	taoshipv	Soviet ships: ASuW cruiser, ASW ship, destroyer, ASCM patrol craft.
taoair	taoairv	Soviet aircraft.
taosub	taosubv	Soviet submarine.
taous	taousv	U.S. ships, submarines, aircraft.

3. U.S. Army Aviation R & D Facility, Ft. Rucker AL. The CBMS was used by the Army to develop threat recognition/memorization training for Army helicopter crews. The R&D facility at Ft. Rucker created a videodisc containing helicopters, tanks, anti-aircraft guns, and surface-to-air missiles from various countries (U.S., Soviet, NATO). Different CBMS databases were configured at Ft. Rucker to provide both fact quizzes and picture recognition training. The picture recognition techniques were unique in that the images were made difficult to recognize by timed presentations in which the images advanced from far to near or were deliberately shown in a cluttered terrain scene. The helicopter crew trainees receiving this instruction were located at both Ft. Rucker, AL and Ft. Campbell, KY. Details on the development of this application were reported by Halff (1987b), but student evaluation data were never formally released. These databases were also installed at NMITC and are listed at the bottom of Table 2.

4. NAS Oceana. The CBMS was provided to the Commander Tactical Wings Atlantic (COMTACWINGSLANT) for use by operational intelligence officers and intelligence specialists at the request of the CNET Training Technology Implementation Office. In November 1988, two CBMS equipped machines were installed in the briefing room library of this operational air intelligence unit, and in March 1990, the platform videodisc was added. The previously developed unclassified platform databases shown in Table 2 provided fact memorization and picture recognition threat training. The value of CBMS reported at this site has been for refresher training for air intelligence officers who visit the facility while their carrier is in port.

5. CNTECHTRA and NATTC Millington. A significant contribution to the evaluation and dissemination of CBESS was provided by instructional development personnel at CNTECHTRA, Millington, TN. CNTECHTRA personnel received their initial training in the use of the LSCAI, GCBI, and CBMS packages in July 1987, and then received several subsequent update training sessions. As a result of their use of the CBESS, many application lessons were developed and numerous improvements were added to the programs by NPRDC. The lessons developed were generally for remediation applications and were developed with the CBESS LSCAI and GCBI packages. Table 4 summarizes many of these lessons.

The initial application fostered by CNTECHTRA was for various aviation ratings in the Job Oriented Basic Skills (JOBS) program at NATTC Millington TN. Microcomputers were installed in 1987 to deliver several CNTECHTRA developed CBI applications. First, LSCAI was used to computerize three JOBS vocabulary modules covering 154 definitions of non-technical, electricity, and electronics terms. Effectiveness data (reported below) showed that this structured practice increased test scores and reduced retests. The JOBS instructors used these modules in various ways at different times to provide structured practice outside of class, as primary instruction in lieu of the class, mixed with classroom instruction, and as a tool to free the instructor to devote more time to problem students (a phenomenon of computer laboratories also reported by Schofield, Evans-Rhodes, & Huber, 1990). A second early application developed for JOBS was a set of study skills lessons designed to acquaint students with techniques for studying and reading more effectively. The topics of these lessons are included in Table 4. Portions of these lessons were subsequently modified in variants that tailored them to other ratings. A third set of lessons developed for the JOBS program addressed remedial mathematics training problems. A portion of the JOBS curriculum addresses a range of basic mathematics topics and these lessons were developed to be complementary to the paper-based lesson materials in areas identified as being recurrent problem areas for students.

Several significant contributions to the CBESS effort resulted from the use of CBESS at CNTECHTRA. First, regular use of CBESS provided continuous feedback that formed the basis for instituting numerous changes in the CBESS programs. That feedback served to eliminate program bugs, suggest new features, and reveal common interface issues that only field usage could provide. Second, the work at CNTECHTRA resulted in the development of lessons that were then directly fielded in schools for the benefit of students. Third, the CNTECHTRA instructional development team fostered additional CBESS application sites by either managing them directly or coordinating them with the CNET Model Schools effort. CNTECHTRA personnel initiated the development of a CBESS test site in 1988 at the Naval Construction Training Center, Gulfport, MS. In 1990, this work was exported to a second Naval Construction Training Center, Port Hueneme, CA. CNTECHTRA personnel also delivered CBESS lesson material to the Great Lakes Model EM A-school and to the Data Systems Technician (DS) A-school, Mare Island. Finally, the stability of the CNTECHTRA development team highlights this as an important element in their successful development, maintenance, and expansion of CBI products. By contrast, efforts at some sites were undermined as a result of regular personnel rotations or within command reassignments, which caused the loss of experienced developers, discontinuity, and the need for retraining.

Table 4

CNTECHTRA Instruction: Study Skills and Electrical Terms

JOBS Module 11.1 (Non-technical terms)
JOBS Module 11.2 (Electricity terms)
JOBS Module 11.3 (Electronics terms)
Improving Your Study Skills:
Studying
Improving your memory
Effective listening
Taking good notes
Tips on taking tests
Reading More Effectively:
Finding the main idea
Skimming
Understanding charts and graphs
Using your rate training manual
Signed Numbers:
Add / Subtract (easy)
Add / Subtract (intermediate practice)
Multiply / Divide (easy)
Add / Subtract / Multiply / Divide (difficult)
Fractions:
Quiz on very elementary fraction terminology
Addition with a common (same) denominator
Reduce to lowest terms
Elementary Math Terminology Review

6. Naval Construction Training Centers. In 1988, CNTECHTRA initiated, developed, and coordinated a CBESS test site for SEABEEs at the Naval Construction Training Center in Gulfport, MS. This site was attractive because existing CBI development efforts were underway by a Chief Builder (BUC) who had been developing BASIC language programs on remediation skills. CBESS eliminated the need to program at very low levels and simplified the development of graphics. The Chief Builder used the CBESS LSCAI and GCBI packages to develop lessons on site. NPRDC provided computers to establish a remediation laboratory. Lessons developed at Gulfport were mostly for the Construction Electrician (CE), Equipment Operator (EO), and Utilitiesman (UT) ratings, although some general lessons were also applicable to the Builder (BU), Engineering Aid (EA), Construction Mechanic (CM), and Steelworker (SW) ratings. Examples of general lessons included general study skills and mathematics practice such as converting between feet and inches. Examples of specific lessons were in geometry, electricity, boilers, refrigeration, and a CNTECHTRA developed CBMS PICTURE quiz for recognizing heavy equipment operator hand signals (see Figure 13). The lessons currently used at Gulfport include those listed in Table 5, and several listed in Tables 4 and 6. A description of this work was reported in McCormick, Jones, and Wetzel (1989).

Records for nine months ending June 1990 show that an average of 65.6 students per month used the the math, reading, and study skills lessons during the period prior to the start of formal courses. Students use the other specific content lessons for review and remediation during courses.

The training given at Gulfport duplicates some of that given at the Naval Construction Training Center at Port Hueneme. To capitalize on previous development efforts at Gulfport, CNET designated Port Hueneme as a Model School and efforts began in 1990. The Gulfport program was also designated as part of the Personal Enhancement Program (PEP).

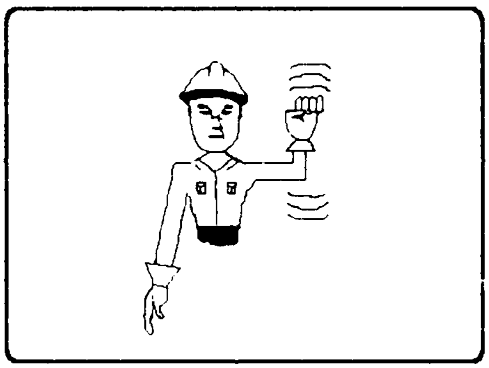

Figure 13. Sample item from heavy equipment operator hand signals quiz.

Table 5

Naval Construction Training Center Instruction

Construction Electrician (CE):
Electrical symbols quiz
Introductory schematic symbols
Basic electricity review
Triangle hypotenuse, altitude, base
Trigonometry sine, cosine, tangent
Utilitiesman (UT):
Boiler introduction, auxiliary equipment, fittings, steam cycle
Plumbing symbol review with labeling quizzes
Refrigeration review quizzes
Equipment Operator (EO) hand signal picture quiz
Practice in converting between feet and inches, with rounding
CNTECHTRA study skills lessons adapted to CM, EA, EO, BU, SW ratings
CNTECHTRA JOBS electrical vocabulary review

7. EM A-school NTC Great Lakes. The Electrician's Mate (EM) A-school NTC Great Lakes was the first Model School designated by CNET for special support in creating an example of infusing new technologies in instruction. The CBESS LSCAI and General CBI packages were used in this effort and initial training was provided in March 1989 to an EM Petty Officer. He developed lessons such as resistor color code practice, a tutorial on ship running lights, and AC motor controller troubleshooting lessons with a previously developed videodisc. NPRDC developed special driver software to allow the use of older model videodisc players already in use at Great Lakes. CNET established a remediation laboratory with computers, which were upgraded by NPRDC to enable the use of CBESS. Subsequent lesson development has contributed to a growing library of Great Lakes lessons such as practice on various series, parallel, and combination circuits. Table 6 lists many of these lessons. Figure 12 illustrates one of the series circuit practice lessons. Several additional CBESS training courses have been given at Great Lakes to EM A-school personnel and to individuals from the Curriculum Instructional Standards Office (CISO), the Training Development Unit (TDU), and the Fire Control School. The success of the model EM A-school resulted from a combination of efforts that includes the allocation of school personnel resources to this special effort, the efforts of the CISO and TDU, and contributions from NETPMSA, NPRDC, and CNTECHTRA. That is, the success of the effort is attributable to special attention and the provision of extra resources. Since numerous other courses are taught at the Great Lakes NTC, various other schools are currently beginning efforts that may also employ CBESS.

Table 6

Electrician's Mate Instruction

Resistor color codes practice
Review of basic matter
Series circuit: selecting the right formula
Series circuit: practice solving for six values
Series circuit: practice solving for selected values
Series/parallel circuit: computation practice
Parallel circuits: computation practice
Parallel circuits: current tutorial and practice
Parallel circuits: resistance tutorial and practice
Parallel circuits: power tutorial and practice
Reactance-time constants: solving for values
Electrical symbols quiz (using CBMS PICTURE game)
Electrical symbols review & quiz (using GCBI)
Navigation lights for Electrician's Mates
AC motor controllers troubleshooting videodisc
Steam cycle for Electrician Mates

8. IC A-school and NTTSG NTC San Diego. The Interior Communications Electrician (IC) A-school in San Diego was designated as a CNET Model School in 1990. Instructional development for this site was provided by CNTECHTRA's Naval Technical Training Support Group (NTTSG), Curriculum Development Team, San Diego. NTTSG developers were given CBESS training at NPRDC in March 1990. NTTSG developed CBESS lessons with the LSCAI and GCBI packages. Table 7 lists some of the lessons developed. As with many of the Model Schools, applicable lessons developed at other sites are also used; e.g., EM A-school lessons and CNTECHTRA study skills lessons.

Table 7

Interior Communications Electrician A-school Lessons

Transistor amplifier operations
Basic diode rectifier circuits
Function & operation of semiconductor & zener diodes
Voltage divider networks
Theory & operation of transistors

9. NATTC Lakehurst. The Commanding Officer at NATTC, Captain J.M. Kaiser, USN, used the CBESS LSCAI package to develop technical vocabulary instruction for an Aviation Boatswain's Mate Fuels (ABM-F) course. The instruction included terminology on different fuel valves, as shown in Table 8. Captain Kaiser learned how to use the software from the user manuals, and his well-constructed lessons included detailed graphics of the fuel valves (see Figure 14). A full account of this work is contained in Kaiser (1989), a Doctoral Dissertation for Nova University which contains lesson development details, an evaluation checksheet of the LSCAI capabilities, and an evaluation of cost and implementation issues. As discussed later, a noteworthy feature of Kaiser's work was the application of a series of selection criteria to determine which course would benefit the most from the effort required to develop the CBI.

Table 8

NATTC Lakehurst ABM-F Course: Fuel Valve Lessons

Introduction & review of types of gauges
Gauge component review (operation, care and safety)
Introduction to valves
Gate valve components & operation review
Gate valve drill #1
Gate valve drill #2
Globe valve review (use, operation & components)
Butterfly valve review (use, operation & components)
Swing check valve (review components with drill & practice)
Eductor review (function, components & operation)

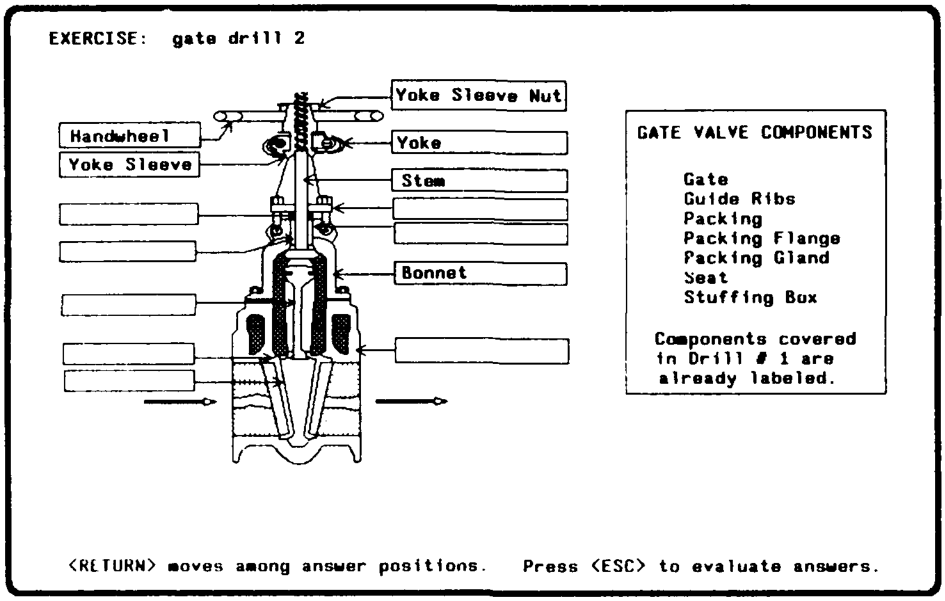

Figure 14. LSCAI graphics labeling activity for fuel valve lessons.

10. Waterfront trainers. CNET maintains waterfront training trailers at the naval operating bases in Norfolk and in Long Beach (formerly San Diego). These trailers are part of a demonstration project providing conveniently located instruction to shipboard personnel. In conjunction with the NETPMSA Instruction Technology Implementation office, the project provided a single videodisc system to each of these sites in 1990. These systems provide threat memorization/recognition training with the CBMS programs and the NPRDC developed platform videodisc as well as selected lessons from the other project test sites.

11. Mail distribution to other than test sites. CBESS was distributed by mail to various DoD commands in response to requests from individuals who had seen CBESS at test sites or had learned of CBESS from other recipients of mailed copies. An average of 21 mailings per year were made in the past three years, with the most frequently requested portion of CBESS being the CBMS and threat databases. These mailings did not include regular updates of new software releases, formal training, or installation visits as with the NPRDC test site activities cited. The phone consultation and assembly time required to support mailed distributions should be an expected implementation element in the future.

12. Informal sites. Several small application sites evolved from the interaction with users of mail distributed copies of CBESS. The Naval Reserve Operational Intelligence Unit 0194 at the Naval Air Station, North Island, CA used the CBMS threat databases for refresher training on drill weekends. Several reservists received training in installing CBMS so that they could install it at other local units or ships. Another reservist provided subject matter expertise in the development of CBMS databases on third world platforms.

Several small applications relating to repetitive recognition practice also evolved from the use of CBESS by several Chief Signalmen of the Fleet Training Center, San Diego, and the USS RANGER. The CBMS threat database line drawings were used for silhouette identification with the PICTURE program. Table 9 shows other databases developed in which this technique was readily adapted to showing other images for recognition practice relevant to the Signalman rating. These same techniques were also used for picture recognition applications of electrical symbols (Table 6) and SEABEE heavy equipment operator hand signals (Figure 13). The technique can be used to test individuals by multiple choice questions or by having students type in answers in several forms. Because of the precise timing requirements for flashing light practice, NPRDC developed a special program to permit students to practice alone without the need for a second person. This program is named MORSE and is illustrated in Figure 15.

Table 9

Signaling Applications

MORSE: A flashing light practice program
International signal flags & pennants
Flags of major maritime nations
Soviet signal flags
Semaphore quiz, by single letters, or by opposites
Semaphore practice on 5 letter sequences

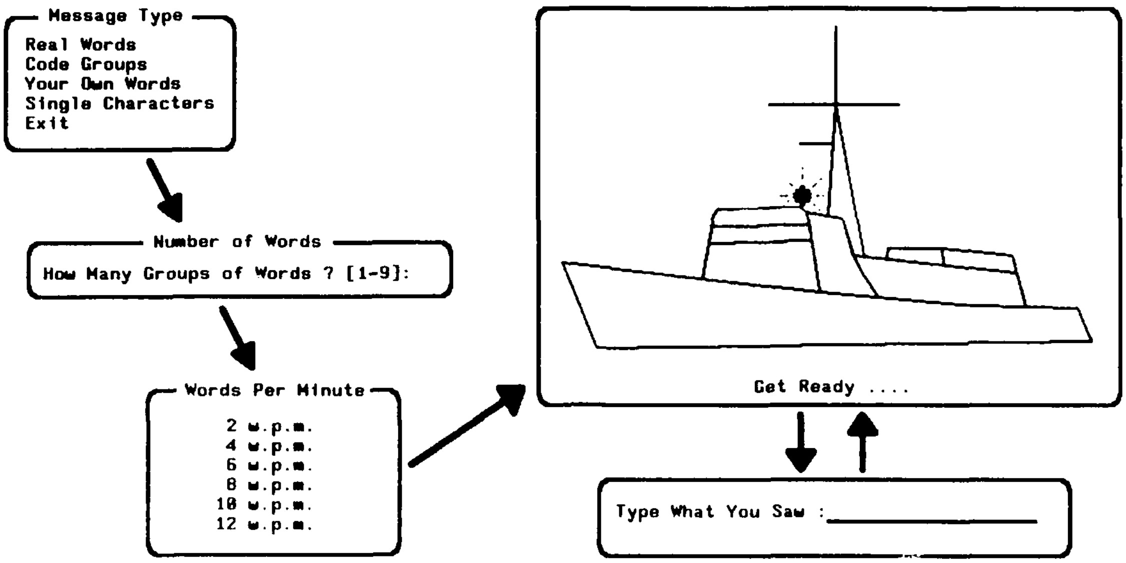

Figure 15. MORSE flashing light practice program.

13. CNET Model Schools and other potential applications: The CNET Model School program has been expanded from its initial site at the EM A-school NTC Great Lakes to several other sites. Table 10 lists all of the existing Model Schools and those which were being considered as potential Model Schools as of this writing. Several of these sites were discussed previously, and the remainder are potential users of CBESS as this program expands.

Several other potential CBESS application sites are worth noting even though no formal plans exist to make them sites at present. First, the CBMS threat databases cover information taught at several other sites which could capitalize on work already completed. The Tactical Action Officer (TAO) course taught at FCTCPAC is the course model manager for a second identical TAO course taught at FCTCLANT Dam Neck VA. Likewise, the Surface Warfare Officers School (SWOS) Newport RI includes portions of this same curriculum. The Fleet Intelligence Training Center, Pacific (FITCPAC) San Diego also teaches portions of this curriculum, although the courses there are somewhat shorter. Second, the Operations Specialist (OS) A-school, at FCTCLANT Dam Neck currently uses the original Apple-II computer version of the LSCAI in a remedial program which has reduced the attrition of low reading grade level students. These students review 450 technical terms from the OS A-school curriculum during a two week period before they start the A-school. As the current hardware is at the end of its useful life, conversion of this successful training material to Navy standard microcomputers running the newer CBESS LSCAI offers a potential future application. This seven-year old LSCAI application provides a useful data point on the expected longevity of the hardware employed in a computer laboratory.

The Naval Surface Reserve Force (COMNAVRESFOR), New Orleans, LA, is a potential user of courseware previously developed at various CBESS test sites. A large number of U.S. Army Electronic Information Delivery System (EIDS) machines were purchased for deployment in reserve centers by COMNAVRESFOR. Because other instructional development efforts might produce lessons of use in reserve centers, COMNAVRESFOR provided NPRDC EIDS machines for testing and development purposes. While initial versions of the EIDS machine were underequipped, recent hardware upgrades and adjustments to CBESS have made them suitable for direct implementation of CBESS courseware. Additionally, the MORSE flashing light practice program developed on this project was previously selected for use by the reserves, and NPRDC provided program modifications to adapt it to the EIDS machines.

Table 10

CNET Model Schools

Electrician's mate (EM) A-school, Great Lakes *
Machinist's mate (MM)/Boiler technician (BT) A-school, Great Lakes
Fire control (FC) A-school, Great Lakes *
Gas turbine system technician (GS) A-school, Great Lakes
Electronics technician (ET), A-school Orlando
Electronics warfare (EW)/Crypotologic technician (CT) Corry Station
Aviation electrician's mate (AE) A-school, NAS Memphis *
Avionics technician (AV) A-school, NAS Memphis
Air traffic controller (AC) A-school, NAS Memphis
Electronics technician (ET) A-school, Groton
Basic electricity rate training (BERT), Groton Submarine School
Construction Training Center, Gulfport *
Construction Training Center, Port Hueneme *
Interior communications electrician (IC) A-school, San Diego *

* Users of CBESS

EVALUATIONS

Evaluation data from the project fell into two broad categories. First, an on-going program of incremental modifications to the authoring and student delivery software was based on in-house and user evaluations in order to improve the efficiency and effectiveness of the system. Second, instruction was evaluated as it accumulated in development efforts and in operationally fielded instruction. The evaluation data included: user comments on usefulness, records of authoring experiences, revision cycles on instruction, records of software modifications completed, interactions with users and requests for new features, frequency of usage, and usage data correlated to student performance. A few evaluations reported here are based on earlier versions of the programs which still reflect the same basic techniques of later versions. This section does not include informal phone, letter, or message evaluations or experiences summarized in the discussion section. The usability of the software products was a primary project focus and overall evaluations of developed instruction in terms of student performance was limited by the availability of resources contributed by remote sites.

Analysis of Software Development Effort

In-house software modifications to CBESS were documented on a regular basis on programmer check-in forms, which were then retrospectively analyzed to determine the distribution of effort. The software check-in process involved submission of a candidate software change, evaluation by two individuals, entry of the change into the master set of code in a manner that allowed reconstruction of the history of changes, then rebuilding the executable CBESS programs on both UNIX and MS-DOS machines, and finally general release to users with versions denoted by the release date.

The time period analyzed encompasses two and a half years from 1988 to 1990, in which the software modifications were divided into 10 quarters. Each of 470 software changes was assigned an estimate of the number of hours needed to make the modification and then characterized according to five independent ratings: (1) was the change a program bug fix, a new feature, or an overhead/maintenance action; (2) was the change initiated by the developer or did it grow out of interactions with the users in response to a user request, suggestion, or complaint; (3) did the change involve an upgrade to the user interface; (4) was the change specific to a CBESS package (e.g., LSCAI, CBMS, etc.) or was it general to all CBESS programs (e.g., a common library function); (5) did the change imply an eventual change to the user manuals or not, or was it an actual check-in of a changed manual. The number of hours estimated for each change included the hours of all persons involved in planning, making, and evaluating the change. The number of hours estimated for each change were then converted into the percentage of the total number of estimated hours for each of the five ways of characterizing the software modification.

The overall percentages of effort hours for each of the five ratings of a software change are shown in Table 11. Overall 56% of the changes were for adding new features; 22% for fixing program bugs; and 22% for overhead items. Examples of overhead/maintenance items are converting to new compilers or host machines, maintaining a reconstructable record of source code changes, or software engineering techniques not apparent to users that optimized the code, its organization, or the creation of executables. Over the course of the rating period, the number of bugs was high at first and then dropped to a constant low level, new features increased with time, and overhead items were relatively constant except during periods of start-up and significant conversion efforts. Forty three percent of the effort was related to interaction with users in addressing requested or suggested changes, and 44% was devoted to changes apparent in the interface. Both of these increased over the rating period. These increases are partially related to an increased distribution and availability of the programs over time, yielding an increased opportunity for receipt of comments. Since CBESS is a system of programs with many libraries common to the individual programs (e.g., graphics, windows, menus, file input/output), 59% of the effort was devoted to changes general to all of the CBESS packages. One half of the changes implied a future change to the user manuals, and 11% of the effort was devoted to actually producing new manuals (Appendix A lists the manuals).

Table 11

Five Ratings of Percent Estimated Software Development Effort

56 %	New Feature Added
22 %	Bug Fix
22 %	Overhead/Maintenance
43 %	User Requested/Suggested/Complaint Change
57 %	Developer Initiated Change
44 %	Interface upgraded
56 %	No effect on interface
59 %	General to all Packages
41 %	Specific to a Package
50 %	User Manual Change Needed
39 %	No change needed to User Manual
11 %	Actual Check-in of User Manual

Note: Each block adds to 100% and each is a separate rating of same data.

Table 12 shows a further breakdown in which user and developer based changes are separated and percentages are recomputed within these categories for two of the other ratings. When conditionally cross classified in this manner, it becomes apparent that users predominately requested or suggested new features (85%) that also generally affected the interface (71%). These changes were by definition of joint interest with the software developers, with the remainder being ones in which no mention by a user could be cited. Thus, by exclusion, changes initiated solely by the software developers reflect less obvious technical issues and showed a more even split among features, bugs, and overhead categories. For example, overhead work was 38% of the effort and 34% were new features not overlapping with user comments. Ongoing internal evaluation of the programs also revealed a greater proportion of effort devoted to program bugs (28%) than was ever apparent to users (14%). Over time, program bugs fell to constant low level of about 14% overall and often reflected intermediate program versions containing provisional new features.

Table 12

Percent Estimated Effort Within User and Developer Initiated Software Changes

43 %	User Requested/Suggested/Complaint Change	
	85 %	New Feature Added
	14 %	Bug Fix
	1 %	Overhead/Maintenance
	71 %	Interface upgraded
57 %	Developer Initiated Change	
	34 %	New Feature Added
	28 %	Bug Fix
	38 %	Overhead/Maintenance
	24 %	Interface upgraded

Note: Indented entries were recalculated after separation into User or Developer initiated changes.

Study of CBI Decision Criteria and LSCAI Authoring

The LSCAI software was used in a exemplary study of the CBI decision, development and initial evaluation process conducted by Captain J.M. Kaiser (1989) at NATTC Lakehurst, NJ. The value of Kaiser's study was that it actualized some of the CBI implementation decision criteria discussed in Wetzel et al. (1987a, 1987b). Those criteria emphasized the selective consideration of only content areas judged to benefit from conversion to CBI rather than conversion of complete courses or all types of course materials.

In the first phase of Kaiser's study, a decision matrix was constructed in order to choose among NATTC courses that would benefit from development with CBI. This matrix incorporated information on student throughput, level of training, course stability, importance of the course, and attrition/setback rates. The top three courses from these combined rankings were then subjected to a detailed analysis of the type of training objectives employed with the Instructional Quality Inventory (IQI) (Ellis et al., 1979). This secondary analysis determined the appropriateness of the course objectives to the specialization of the LSCAI program to drill and practice applications of review, structured self-study, and remediation. The final course selected by the decision process was an Aviation Boatswain's Mate Fuels (ABF) A-school course, which had higher proportions of objectives requiring the remembering of facts. An alternative decision at this point would have been an evaluation among software tools appropriate to the requirements of the instruction. This evaluation was not conducted since Kaiser's original intent was to employ the LSCAI program.

In the second phase of Kaiser's study, the LSCAI was used to develop instruction for the selected course material and a formative evaluation was conducted. The events of the authoring process were recorded along with the development time for the selected course. A total of 5 hours of classroom instruction was covered during 52.2 hours of CBI development time, yielding 10.4 hours of development time per hour of classroom instruction. The most time consuming portion of the development was creating graphic illustrations of fuel values, which took approximately one quarter of the development time (12.3 hours of the 52.2 total hours). These development times were quite reasonable, primarily because the LSCAI programs are specialized for one type of instruction and therefore can reduce its development time. Kearsley (1983) cites a rough rule of thumb of from 100 to 200 hours of development for each hour of computer-based delivered instruction. Development times range widely depending upon detailed features of the instruction (e.g., the complexity of the graphics, the use of video, or the fidelity required to actual equipment). The development of this courseware was conducted after reading user manuals and without formal training. Experience with formal training conducted at NPRDC would estimate about a week's training time in all phases and options of LSCAI.

Following development of the actual instruction, Kaiser performed three ratings with standard forms: (1) a check list of LSCAI program features, (2) a small sample of students completed an evaluation form after trying out the instruction, (3) a evaluation questionnaire was completed by staff members. The study stopped short of collecting effectiveness data relative to student performance. The evaluations reported in this phase were favorable and comments were similar to others received during the project concerning details about the instruction, or suggestions or requests for program modifications.

Kaiser also developed illustrative preliminary estimates in a cost-effectiveness analysis that pitted one-time costs against annual costs. The one-time costs were for hardware, software, laboratory facilities, and staff and for courseware development. The annual costs included only maintenance, with estimated cost savings being based on reduced attrition and setbacks and for reduced instructor classroom time replaced by the automated reviews with the CBI. No costs were estimated for laboratory staff since it was assumed that this function would be performed by watch personnel. This analysis yielded around 12% savings in unamortized first-year annual costs over the one-time costs. This analysis should only be considered as one preliminary estimate relying on various assumptions untested with actual data. The real value of the analysis was in illustrating how others might set up similar site specific analyses and eventually collect substantiating data.

LSCAI in the Academic Remedial Training Program

The current CBESS LSCAI programs evolved from earlier work with an Apple computer version of LSCAI developed by Wisher (Wisher, 1980, 1986; Wisher & O'Hara, 1981). Wisher (1986) summarizes performance data with the Apple computer version of LSCAI with recruit students at NTC San Diego in the Academic Remedial Training (ART) program. A control group of 75 recruits received conventional classroom-based literal comprehension training involving workbooks and interaction with an instructor. These control students received instruction until their score on an exit test exceeded 70%. An experimental group of 75 comparable recruits received a fixed 5 hours of computerized instruction on the same material. Both groups improved approximately one grade level in literal comprehension skills upon finishing the instruction, and did not differ significantly on this measure. The control group required 9.4 hours of instruction to achieve the same exit test performance as the computerized group, which was limited to only 5 hours of instruction. Thus, the computer-based approach was judged to be more efficient than the conventional approach because these students required less training time to achieve the same level of skill on a comprehension test.

Effectiveness of LSCAI in the Memphis JOBS Program

The LSCAI package in CBESS was used by instructional developers at CNTECHTRA in Memphis to provide computerized training on electrical terms for Jobs Oriented Basic Skills (JOBS) students at NATTC Memphis. These students were enrolled in the Electricity/Electronics Strand of the JOBS program from 1987 through 1988. The students did not initially qualify for A-school training and were enrolled in the JOBS curriculum to receive additional training to qualify them for A-schools in aviation ratings.

Three portions of a JOBS module were converted to CBI with the LSCAI programs by CNTECHTRA instructional developers: Module 11.1 consisted of 49 general nontechnical terms such as implosion, consumed, differentiate, conjunction; Module 11.2 consisted of 63 electricity terms such as amplitude, battery, capacitance, inductor, sinusoidal; Module 11.3 consisted of 42 electronics terms such as oscillator, phosphor, regulator, bandwidth, clamper. The students studied the modules in the order in which they are listed (above). The LSCAI instruction consisted of an initial linear instructional sequence, followed by various test-like activities consisting of multiple choice with feedback, true-false with feedback, matching, definition building, filling in blanks in a paragraph, and graphics labeling.

The progress test measures used in this evaluation consisted of: (1) percent correct on 30 progress tests and (2) the number of retests required to pass a progress test. For each of the three modules, students regularly took written paper-and-pencil progress tests, five tests with "definition-type" questions and five tests with "example-type" questions (a total of 30 tests). Definition-type questions involved students matching words with their definitions and example-type questions required students to translate a given example into a decision about which word applied to the stated example. Any student who did not achieve 80% on any test was required to repeat that test later for as many attempts as needed to reach the 80% passing criterion.

Two different LSCAI experimental groups were employed, each of which was compared to one of two matched control groups who received no CBI at all. All students had the same instructor. Students in the control group were matched to students in the experimental groups on two criterion measures obtained before the instruction began: (1) Gates-MacGinnite Level F Reading Grade Level (RGL) percentile and (2) Armed Services Vocational Aptitude Battery (ASVAB) Electrical Information (EI) scores. No between-group differences were significant on either matching criteria. The matching criteria for the four groups and the number (n) of students in each are shown in Table 13.

Table 13

JOBS Experimental and Control Groups

Group	Number of Students	RGL %tile	ASVAB EI
LSCAI Structured Practice in Afternoons	12	46.7	53.0
Control	12	42.7	52.6
LSCAI as Primary Instructional Medium	8	35.3	48.7
Control	8	35.5	46.3

Both control groups were the same, with the exception of their being matched to the students in their respective experimental groups. The control groups received instructor review of the materials in the mornings and were left to their own devices to study the material in the afternoons in the barracks. The 12 students in the "LSCAI Structured Practice in Afternoons" group received instructor training in the mornings and returned to the JOBS school in the afternoons to get a computer-based structured review of the terms. Thus, these students represent a condition in which extra instruction was given in a structured fashion as opposed to unstructured self-study. The eight students in the

"LSCAI as Primary Instructional Medium" group received the LSCAI materials in the morning in lieu of receiving this training from the instructor.

Figure 16 shows the results in terms of the percent correct and number of retests on the written progress tests for the definitions (DEFS) and examples (EXMP) type test questions. For each of the non-technical, electricity, and electronics progress tests, the five progress tests are combined into one data point. Figure 16 shows that there was very little variance among conditions for the non-technical terms, which were therefore excluded from the statistical analyses. A general trend reflecting the increasing difficulty of the material is also apparent: the percent correct decreased and the number of retests increased as students progressed from non-technical to electricity to electronics terms. Another obvious general trend was that the example-type questions were more difficult than the definition-type questions.

The "LSCAI Structured Practice in the Afternoons" group showed significant beneficial effects from having received the CBI. For both definition and examples type questions, the LSCAI group performed better than their matched controls. The LSCAI group had a significantly higher percent correct on the progress tests ($F(1,22) = 15.03$, $p<.01$) and significantly fewer retests ($F(1,22) = 11.86$, $p<.01$). The "LSCAI as Primary Instructional Medium" comparison groups showed the same trend, but the statistical comparisons were not significant (respectively, $F(1,14) = 2.36$, $F(1,14) = 2.19$, $ps>.05$). The "LSCAI Structured Practice in the Afternoons" and the "LSCAI as Primary Instructional Medium" panels of Figure 16 cannot be legitimately compared since the students in these conditions were not matched and they differed in terms of the RGL and EI measures.

These results indicate that structured practice with LSCAI has significant beneficial effects on test results in comparison to students left to their own devices to study the material. No significant improvement was obtained when LSCAI was used as the primary instructional medium in lieu of an instructor.

Operations Specialist (OS) A-school LSCAI Data

The original LSCAI computer program was developed by Robert Wisher as part of the NPRDC project entitled "language skills assessment & enhancement" (Wisher, 1980). This early version of LSCAI ran on an Apple II computer and was substantially enhanced when included in the CBESS. Wisher's version of LSCAI was implemented at the Operations Specialist (OS) A-school at the Fleet Combat Training Center (FCTCLANT) Dam Neck and is still in use there. Previous data collection with this version of LSCAI was reported via personal communication with Mr. Jamie Stewart. Educational Specialist at FCTCLANT Dam Neck, 12 Feb 1987.

From 1982 through 1983, attrition data were evaluated while the original LSCAI program was in use at the FCTCLANT OS A-school. This evaluation grew out of a desire to reduce the attrition rate and the dissatisfaction with a contractor delivered reading course that was not content specific enough, and embodied poor assessment and delivery techniques. The selection technique for the OS A-school involved both the Gates-MacGinitie reading grade level (RGL) and ASVAB scores. At the time the following data were collected, only students with at least a eleventh grade reading level were admitted to the OS A-school, with 12% of these students attriting. Two groups of

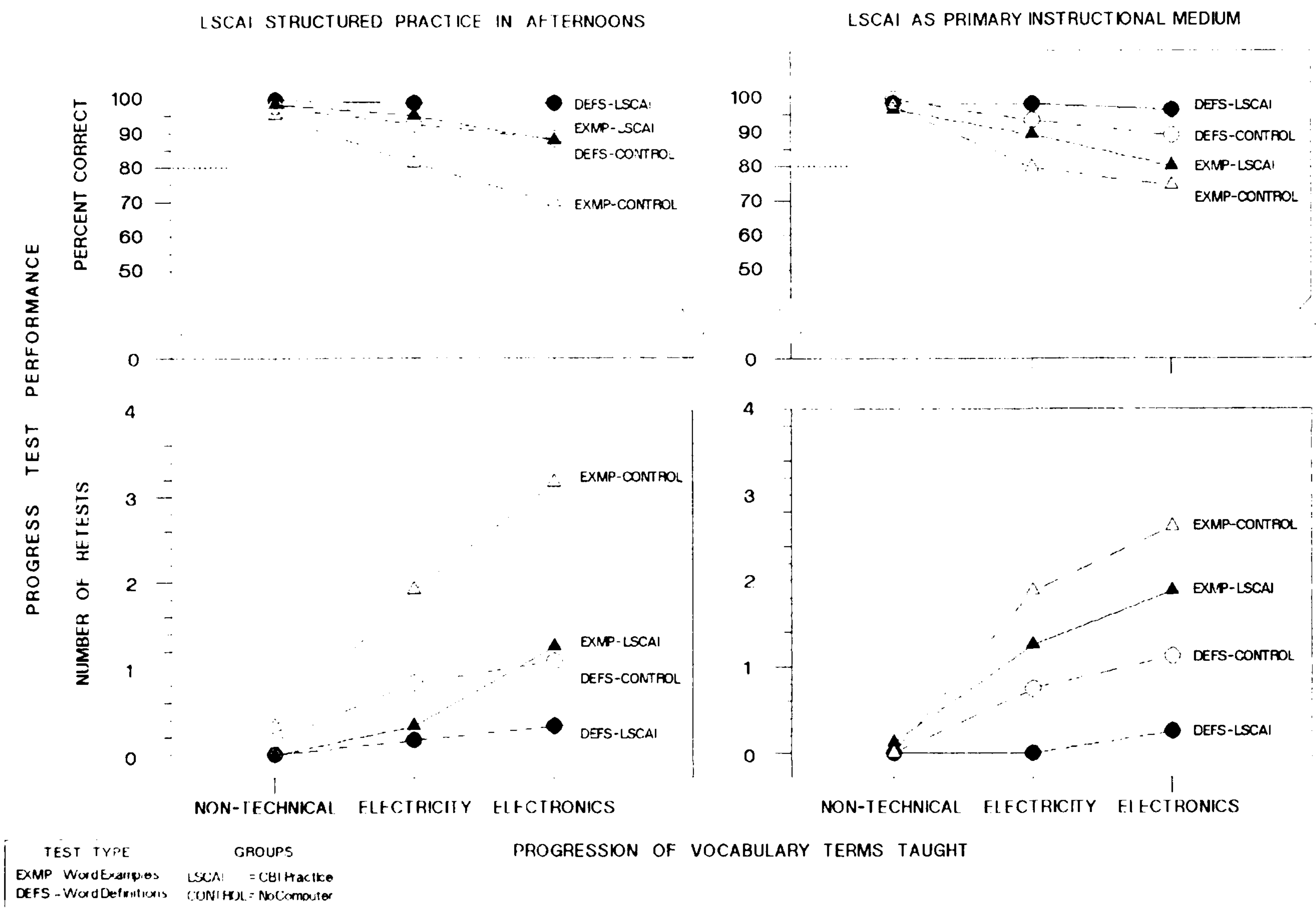

Figure 16. **Progress test performance of JOBS students using the LSCAI vocabulary program or using conventional study methods.**

students with an RGL less than the eleventh grade criterion for OS A-school admission were monitored. The first group was an untreated control group. The second group was the LSCAI treatment group which received a total of 40 hours of CBI for 4 hours a day over a 2-week period. The LSCAI group materials consisted of 450 words taken directly from the OS A-school curriculum which were either general vocabulary words judged to be difficult or technical terms specific to the instruction (e.g., longitude, latitude, meridian). Both the untreated group and the LSCAI treatment group were subsequently allowed to enter OS A-school, and their attrition rate was recorded.

Table 14

OS A-school Comparison Groups

GROUP	RGL Status	Total Attrition Rate
2 week LSCAI treatment	RGL < 11	12.7 %
Untreated Control Group	RGL < 11	18.2 %
Old contractor training	RGL < 11	15.9 %
Regular OS "A" students	RGL > 11	12.0 %

Table 14 summarizes the attrition rates of the comparison groups. The attrition rate of the LSCAI treatment group was 12.7%, which was significantly less than the 18.2% attrition rate of the untreated control group. The attrition rate for the LSCAI treatment group with RGL less than 11 was not significantly more than the 12% attrition rate of the regular students with RGL greater than 11. Thus, the LSCAI treatment group produced attrition results similar to regular students. The contractor training course did not achieve this success level and had an attrition rate of 15.9%.

These attrition statistics are no longer valid because the curriculum and tests have since been revised. A re-evaluation of the attrition statistics for OS A-school students is currently underway, but the data are not yet available. The lesson learned from this instance is that enhancement of skills directly related to subsequent instruction can reduce the attrition rate. The explanation of the effect is that the technical terms taught were operating tools for comprehending other more complex or higher order units taught in the actual course of instruction.

Combined with the descriptions of the CNTECHTRA JOBS implementation, several scenarios are suggested for this type of remediation: (1) enhance skills before the start of regular courses by training content-specific basic skills, (2) provide adjunct remediation when poor student performance is identified, (3) provide the training for more routine drill and practice situations, possibly as a means to allow instructors to devote additional time to poorer students.

Model EM A-school NTC Great Lakes

Evaluations of the Model EM A-school Great Lakes were conducted by CNTECHTRA (McCormick & Jones, 1990) and by the NTC Great Lakes (Service School Command, 1990; Shepard, 1990). The reported data reflect a combination of intervention techniques, with the primary element being a computer laboratory using CBI from various sources, including CBESS lessons.

Data reported by the EM A-school in February 1990 show that the academic attrition rate declined from about 17% to about 9-10% during the period in which the Model School efforts infused new techniques and that the nonacademic attrition rate declined slightly or remained about the same (Service School Command, 1990). The amount of decline in the attrition rate should be qualified because of an accompanying effort with regard to setback students. Shepard (1990) reported that the number of setback students who failed a midway exam but who eventually passed increased from 59% to 85% during the same period.

Table 15 shows the distribution of the types of student users of the computer lessons. Volunteers accounted for half of the users, while the other half used the CBI lessons in the remediation room as a result of a requirement. Utilization of the remedial room after hours was reported to exceed its capacity. The number of computer lessons used over the most recent seven months reported was about 3,000 per month. Questionnaires completed by students were generally favorable on the value of the CBI lessons (74%), and about 50% of the students were favorable toward using the lessons together with another student (Table 16).

Table 15

Reasons for Using Model EM A-school CBI Lessons

Volunteer	50 %
Mandatory Remediation	27 %
Instructor Assigned	8 %
Supervisor Assigned	15 %

Table 16

Results of Questionnaire Administered to EM A-school Students

Type of Question	Positive	NA	Negative
Value of Lessons	74 %	18 %	7 %
Using Computers with Others	52 %	29 %	18 %

NA = not applicable.

McCormick and Jones (1990) compared two groups of 229 EM A-school students from equivalent four month periods one year apart for a before and after comparison of the Model School effort. They found definite decreases in attrition from about 9.5% in the fall of 1988 to less than 3% in the fall of 1989 by examining attrition records from the Navy Integrated Training Resources Administration System (NITRAS) database. Set-back rate fluctuations reflected changes in procedures at the school, but a gradual downward trend was suggested. A second analysis with course test scores showed a decrease in average test scores because more marginal students were being retained in the courses. However, more students passed tests on the first attempt and fewer retests were required in the post-implementation period. The student graduation rate increased from less than 70% to 100% in the two time periods compared. The overall conclusion of this investigation was that the Model School computer laboratory effort had positively affected student performance.

Threat Memorization Training

The generative techniques used in CBMS dynamically assemble database information to create large numbers of questions by funneling database components through question templates. Previous reports relating to the original development of the techniques used in the CBMS threat databases were published in Crawford and Hollan (1983); Halff (1987b); Halff, Hollan, and Hutchins (1986); and McCandless (1981). The current CBMS represents years of refinements which offer an effective tool when large numbers of facts must be memorized. The collection of performance data was problematic because student users were generally self-selected and used the memorization practice as a course supplement instead of as the primary source of instruction. The data summarized below are the only instances of performance reported using the CBMS techniques. Although Halff (1987b) reported on the development of the Army threat databases used at Ft. Rucker, performance evaluation data were never formally released.

The data set analyzed here consisted of 4529 total program execution records obtained between 1988 and 1990 from the FCTCPAC TAO course (36%), the NMITC (58%), and NAS Oceana (6%). Two general types of analyses were possible: (1) patterns of performance based on the entire data set, (2) usage indices from the subset of

TAO data where the possible number of students from a class was known. Class proportions were difficult to determine for NMITC students who used the system in the school library and were self-selected from a wide range of classes. Several selection criteria were applied to the data set to eliminate program executions which were obviously exploratory, were possibly demonstrations, or which used unscored program options. The following selection criteria were applied prior to summarizing the data (below): Trivial program executions in which users were asked less than three questions or attempted less than three answers were excluded. Some analyses also excluded instances in which no questions were asked such as with the BROWSER program and executions using an infrequently used unscored mode where both questions and answers were revealed to the students.

CBMS Program Preference and Database Usage

One use of the CBMS usage data is to judge the relative preference among the 10 different game program approaches to memorization. The practical use of these results is to recommend to future users the most useful games. A result of these empirical observations has been the implementation of menus segregating the most frequently used games from those less frequently used. Additionally, the less frequently games were not included in many mailings in order to reduce the required amount of disk space.

Table 17 has three sections in which each entry is calculated on the basis of each execution of a program or database. The top section of Table 17 lists the 10 CBMS programs in order of their overall percentage of usage. The top four or five games may be considered the recommended reduced set for future users. Excluding the PICTURE game and the unscored BROWSER, the remaining eight programs (64%) are ones that primarily emphasize textual facts as answers. The more frequently used fact games ask direct questions about platform attributes. The less frequently used fact games have somewhat more complicated interfaces, present more information on the screen, and ask students to identify an item based on partial lists of its attributes. The TOUR program is a recently developed variant suggested by NMITC users which combines features of the BROWSER and FLASHCARD programs by reviewing all facts before a quiz is given on just those facts (see Figure 11). The percentages in the table do not add to 100% because the TOUR data were based only on time periods in which it was available. The test sites differed in their usage of the PICTURE game; TAO students executed this program less frequently because they were more interested in fact quizzes containing information more likely to occur on course tests. An average individual session spent with the most popular games was from 20 to 30 minutes in the total sample. The average number of answers are greater than the number of questions because answers were not always correct and because many fact game questions required several answers (e.g., "what radars are carried on the Kirov?"). Were performance perfect, then at one extreme the FLASHCARD game would have required as many as two to three times as many answers per question and, at the other extreme the PICTURE game always would have required one answer per question to identify a single picture.

The two bottom sections of Table 17 show the relative usage among the different CBMS databases. The usage percentages in the middle portion of the table are slightly biased in that some databases were not available throughout the entire time period. For the unfamiliar content given in the Army databases, the proportion correct in the total sample was significantly lower than the relatively similar performance shown with the other databases. The bottom of the table shows usage for three selected TAO classes with access to a stable set of all databases after converting them to classified form. Overall, the Soviet ship and aircraft databases were used most frequently, but some databases were smaller and had fewer questions available (e.g., Soviet submarines). The average time spent in a given session for the TAO students was about 30 minutes and a preference was shown for the nonvideo databases since they required extra display time during fact quizzes. The average percent correct for TAO students was somewhat higher than for the total sample, ranging between 75 and 85 percent.

Table 17

CBMS Program and Database Usage Per Program Execution

CBMS GAMES and DATABASES	Percent Usage	Average Questions	Average Answers	Percent Correct	Average Minutes
PICTURE	35.8	31	41	63	20
FLASHCARD	24.4	56	120	78	26
TOUR	21.2 *	49	89	74	33
JEOPARDY	9.6	17	17	61	17
BROWSER	9.1	-	-	-	8
IDENTIFY	5.3	27	64	60	19
MATRIX	2.5	7	15	54	8
TWENTY	1.3	14	2	89	9
CONCENTRATION	1.2	22	21	71	10
CONSTRAINT	1.1	7	41	72	10
All Soviet Ships	47.4	33	69	73	24
All Soviet Aircraft	19.2	60	89	77	24
All Soviet Submarines	14.2	31	46	67	15
All Army	11.3	19	34	58	15
All US/NATO Platforms	7.6	64	103	74	25
TAO Soviet Ships	40.3	52	122	80	35
TAO Soviet Aircraft	32.0	89	118	85	30
TAO Soviet Submarines	9.4	59	75	85	19
TAO US Platforms	16.6	91	114	75	32
TAO Non-video	69.0	71	120	81	31
TAO Video	29.3	44	76	71	31

* TOUR data based on only time periods in which it existed.
Averages on basis of per program execution with a database.

Repeated Usage by Individuals

Patterns of repeated use by a given individual were determined by selecting all game executions traced to the same person. After eliminating trivial program executions with fewer than three answers or three attempted answers, records in which a person's name appeared more than once were selected. Data for all programs and databases were then summed for each person and sorted by the number of program executions accumulated by that person. This resulted in 417 individuals for analysis (64%), with 234 names (36%) being excluded because they did not reappear (many were random characters or fictitious names).

The data shown in Tables 18 and 19 reflect performance patterns over repeated program executions and not necessarily learning since individual users switched among databases with different instructional content. The percent of people, average time spent, and average executions columns of Table 18 show that about three fourths of the repeat users executed the programs at least three times for at least an hour. A little less than half of the people used the programs at least five times for over two hours. A little less than a quarter of the people used the programs at least 9 times for over almost five hours. The accumulated average total answers attempted shown in Table 18 are broken down into five response method categories in Table 19. These categories reflect the instructional testing strategies offered to CBMS users: a recall method in which answers to a question are typed in, a multiple choice recognition method where answers are selected from lists of right and wrong alternatives, and a "tell-me" method where users can request the correct answer while attempting either recall or recognition. With fewer accumulated executions, the number of correct or true answers given for the recall method are less than for the multiple choice method, but true recall answers become more numerous with greater accumulated executions. False answers are always less frequent for the recall method than for the multiple choice method. These results indicate that repeated users eventually came to prefer the recall response method over the multiple choice method. This preference may be because repeated users had learned more of the content and could answer without looking at the potential answers shown in the multiple choice menus. This interpretation is supported by the consistently greater number of false answers given for multiple choice than for recall. That is, if users had greater confidence in the correctness of an answer in opting for the recall method, then fewer wrong answers would be expected for recall than multiple choice.

Learning Patterns in One Domain

Learning over time was analyzed in a circumscribed domain including only Soviet ship databases for all user records obtained from all sites. These databases were singled out because they were the most frequently used and represented nearly half of all recorded program executions. The records analyzed were segregated on a person by person basis in terms of (1) the PICTURE game and (2) four selected fact games which used similar forms of questioning and answering (FLASHCARD, TOUR, JEOPARDY, and MATRIX). The course of learning for an individual student was preserved by retaining

Table 18

Repeated Usage by the Same Individual

Executions Accumulated	Average Executions	Number of People	Percent People	Average Time (hr/min)	Average Questions Asked	Average Answers Attempted	Average Percent Correct
2	2.0	113	27	35"	46	81	63
3-4	3.4	129	31	1' 6"	90	157	65
5-8	6.2	85	20	2' 12"	221	383	67
9-16	11.6	59	14	4' 43"	438	827	74
17-69	27.9	31	7	11' 47"	1528	2564	79

Each average is for the total accumulated executions for individual people and includes different programs and databases. Percent people column adds to 100% without rounding error.

Table 19

Accumulated Average Number of Attempted Answers by Response Method

Executions Accumulated	Recall		Multiple Choice		Tell-me Answers
	True	False	True	False	
2	13 (16)	5 (7)	38 (47)	18 (22)	6 (8)
3-4	43 (28)	15 (9)	59 (38)	29 (19)	11 (7)
5-8	119 (31)	28 (7)	139 (36)	68 (18)	30 (8)
9-16	338 (41)	68 (8)	272 (33)	119 (14)	31 (4)
17-69	1323 (52)	202 (8)	696 (27)	258 (10)	85 (3)

Percentages within a row are shown in parentheses.

the appropriate execution sequence number for which to compute performance with the Soviet ship databases for the selected programs. False start executions were also eliminated by excluding those with fewer than three questions and three attempts. The learning curves shown in Figure 17 were also constructed so the trial-by-trial performance for each person was included at all applicable points (i.e., a person ending within the 9-16 program execution category is represented in all previous lower categories). For the four fact games, the number of people at each of the 6 points shown on the horizontal axis were 207, 135, 145, 151, 136, and 51. For the PICTURE game, the number of people were 202, 123, 136, 103, 73, and 16.

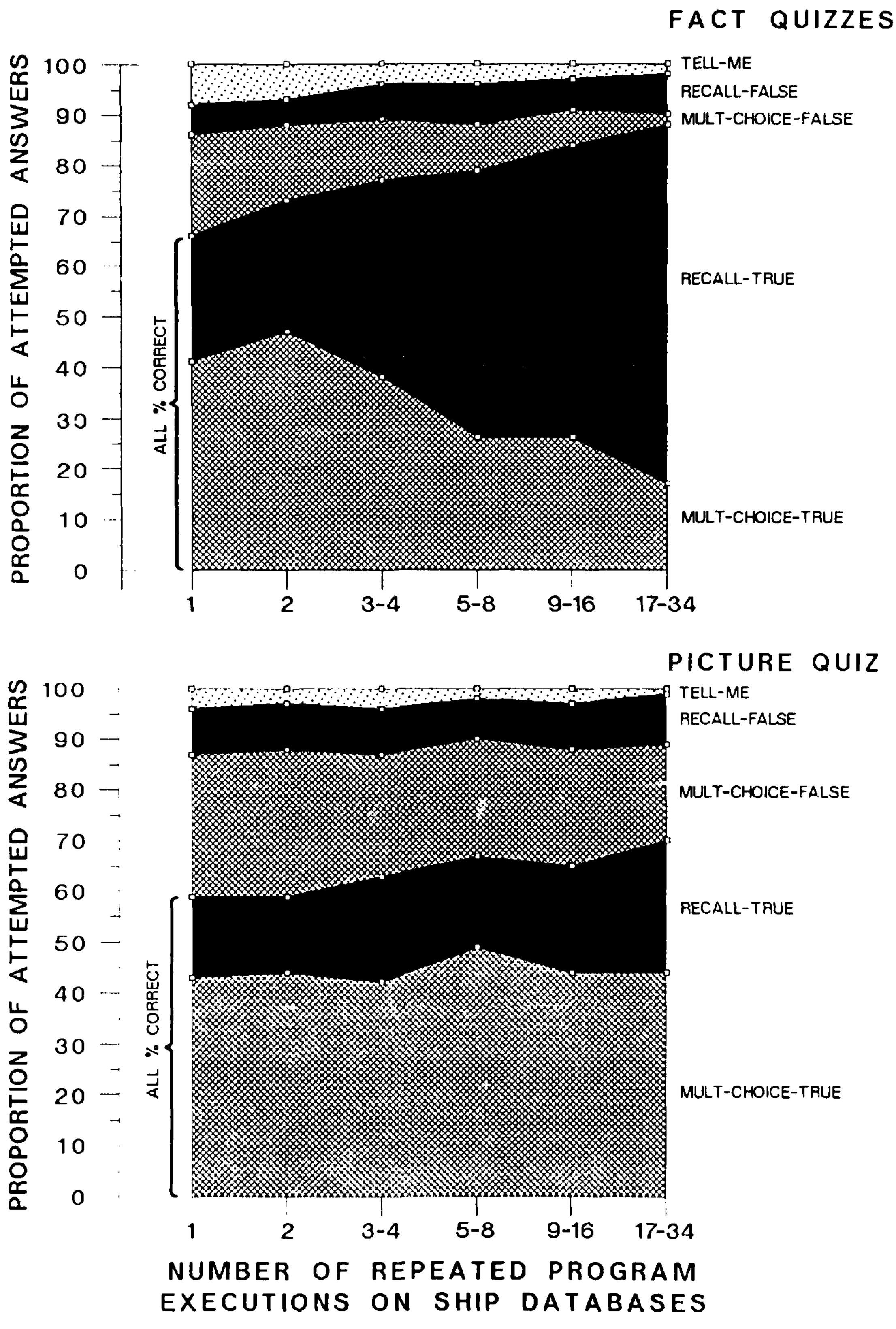

Figure 17. Distribution of response methods and overall percent correct by individuals learning from Soviet ship databases.

The vertical axis in Figure 17 is a percentage that allows two measures to be observed. First, the proportion of all attempted answers is shown in terms of shadded areas for the five methods of responding: true or false multiple choice responses selected from menu lists, true or false recall answers which were typed in by students, and "tell-me" answers in which students asked to be told an answer. Second, the overall total percentage correct may be noted by tracing the upper limit of the recall-true region since the sum of the true recall and multiple choice responses represents all correct answers.

For the fact quizzes, the area representing the overall percent correct is 66% for the first program execution and rises to 88% in the 17-34 program-execution category. The proportion of false and "tell-me" answers decreases correspondingly. The two true answer categories show a pronounced shift over time in which students initially opt for multiple choice responses and then steadily increase their preference for the recall method. True multiple choice responses start at 41% and drop to 16%, while true recall responses start at 24% and rise to 71%. Thus, as students gained familiarity with the content, they could more readily recall answers and they no longer needed to be cued by seeing multiple choice alternatives.

For the PICTURE quiz, the overall percentage correct rises much more gradually from 59% to 70%. The area occupied by the two correct response methods shows little relative change over the executions. In contrast to patterns shown with the fact games, this more gradual increase may indicate a greater familiarity with factual information than with pictorial representations. The true and false multiple choice responses occupied the greatest proportion of responses, suggesting less confidence in responses. This finding is somewhat of a surprise since picture recognition levels would be expected to exceed those of fact learning. Further explanation of this finding is not offered by these data, other than the observation that the PICTURE game was executed less frequently and for shorter durations than the combination of the fact games. The possibility that these differences were due to image quality was examined in a subanalysis showing a small 5% increase in correct performance for later databases enhanced with video images added to the line drawings in earlier databases. However, the increase was in true multiple choice responses at the expense of recall responses, which does not explain an effect of image quality on the selection of a response method.

Tailoring Databases to TAO Course Content

The instruction delivered with the CBMS to TAO students underwent significant changes between 1989 and 1990. The changes were: (1) conversion of TAO course-specific unclassified databases to a classified form that exactly matched the printed "threat matrix" students were given to memorize and from which they were tested at several points in the course; (2) installation of an accompanying videodisc showing actual images of the aircraft, submarines, and ships (including Soviet ship radars, launchers, missiles, guns, and radars); (3) installation of revised CBMS programs and an improved management system that no longer required students to maintain a floppy disk. The most important change apparently was the classification of the databases, as suggested by instructor comments regarding the subsequent increase in system usage.

An examination of system records was conducted to compare two equivalent time periods a year apart, comprising three TAO classes in 1989 (66 students) and three classes in 1990 (59 students). The analysis included only students who were identified in both the computer records and the class roster (51 in 1989, and 49 in 1990). Computer records for a small proportion of student users entering fictitious names were excluded (13%). Figure 18 shows the proportion of TAO student usage in terms of the number of attempted answers recorded by CBMS, broken down in terms of non-usage and trivial, moderate, and heavy usage (up to 50, 51-500, and over 500 attempted answers respectively). Since students were free to use the system according to their own interest, the trivial usage category generally reflects those students who participated in an introductory session conducted by the instructor early in the course. Figure 18 indicates that upgrading the system had a dramatic effect of increasing the percent of heavy users and thereby reducing the number of moderate, trivial and non-users. The 1990 moderate and heavy user categories represented 73% of the students in those classes, where they had represented only 53% in 1989. Table 20 shows the trivial and moderate users had somewhat greater percentages correct in 1990, but heavy users in both years averaged above 80 percent correct. Table 20 also shows that in 1990 the moderate and heavy users substantially increased their hours of use and the number of program executions, questions asked and answers attempted.

TAO Class Standing

As suggested by Crawford and Hollan (1983), the TAO students opting to use the computerized memorization system were students whose class standing was slightly higher than that of students who did not use the system. Non-users had an average class rank of 11.3, while system users scored higher in the class with an average rank of 10.2. Class standing was based upon final measures from the entire class and encompassed much more material than taught with the CBMS. Thus, this finding is interpreted as a characteristic of system users, instead of a causal effect of CBMS on the final class standing. Other analyses were conducted in an attempt to correlate student course performance with computer usage. These were unusable because appropriate quizzes were scored and recorded in different ways over time by different instructors and then discarded because of their classified nature. A special experiment would have to be mounted to obtain such data in which student access was controlled prior to studying specially constructed databases reduced to match the specific content of the quizzes.

CBMS in Groups

Students sometimes out-numbered the available computers at sites using the CBMS threat databases. At these, times a cluster of several students could be observed jointly discussing potential answers. The dynamics of these discussions suggested a different kind of learning occurred than in solitary uses of the games. Conversations often provided embellished explanations or touched on information other than the fact currently being sought by the programs. For example, in discounting a potential answer a debate might include a review of the attributes of a wrong answers. A related observation was reported in Table 16 where 52% of the surveyed Great Lakes NTC students reported positive attitudes in using computers with others. While these group observations were positive, solitary use of the programs may still be of value when students are initially learning material or when they desire to test their own knowledge without potential intervention by a partner.

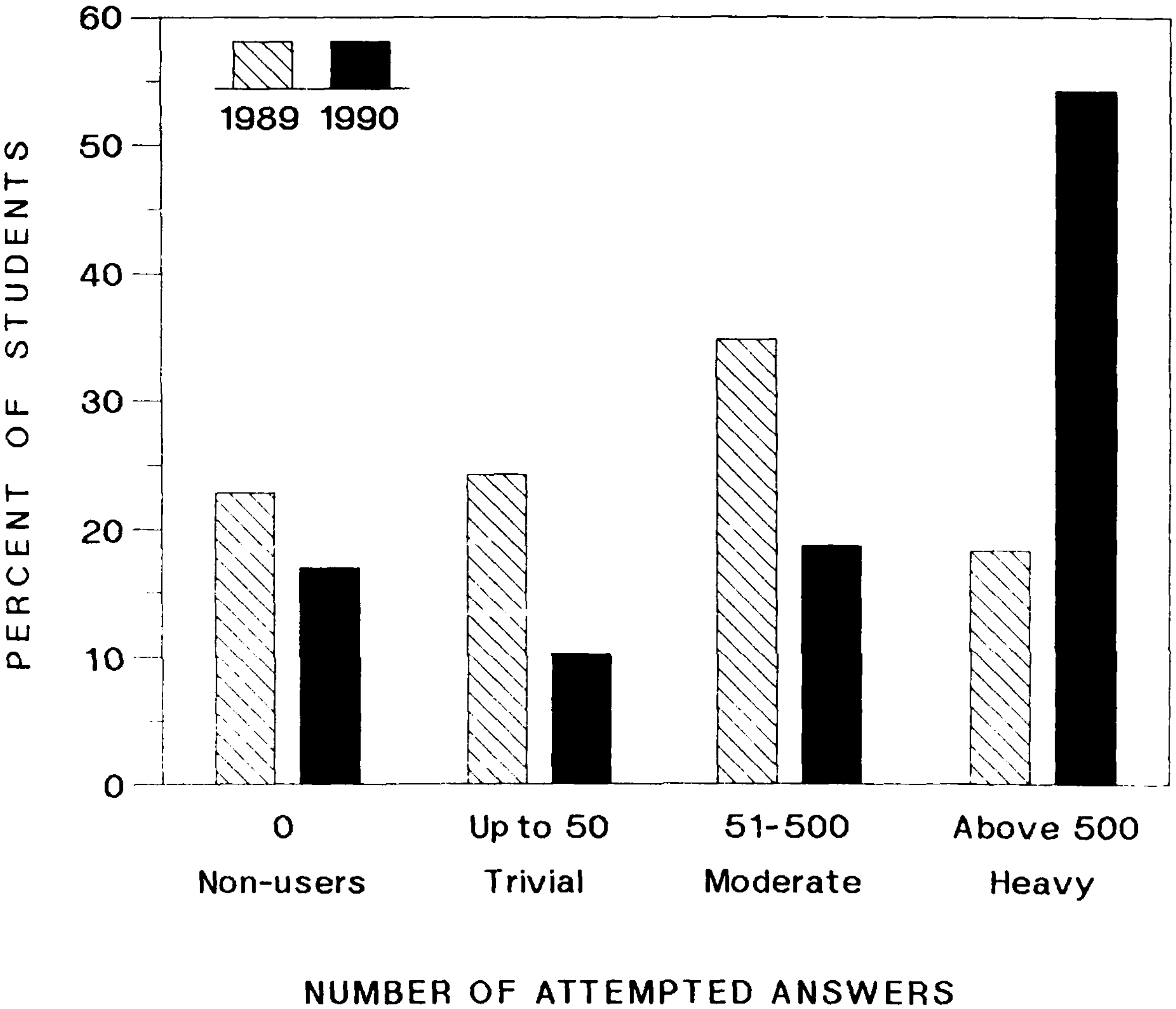

Figure 18. TAO student usage increased in 1990 after conversion of databases to classified form.

Table 20

Selected TAO Classes Before and After Classification of Databases

Measure and Year of TAO Class	User Group *			
	Non-Users	Trivial Users	Moderate Users	Heavy Users
Number of People:				
1989 (3 Classes)	15	16	23	12
1990 (3 Classes)	10	6	11	32
Overall Percent Correct:				
1989		50	74	84
1990		60	81	82
Average (and Max) Hours of Use:				
1989		0.2 (0.4)	1.0 (2.1)	4.5 (7.6)
1990		0.2 (0.4)	2.3 (4.4)	10.4 (40.5)
Average (and Max) Number Program Executions:				
1989		1.2 (2)	3.4 (9)	11.0 (19)
1990		1.7 (3)	6.8 (23)	24.3 (87)
Average (and Max) Number of Questions Asked:				
1989		7 (20)	61 (134)	306 (573)
1990		8 (12)	152 (397)	1392 (7813)
Average (and Max) Number of Answers Attempted:				
1989		20 (50)	186 (387)	1038 (1803)
1990		17 (31)	248 (499)	2279 (12293)

* User Groups defined by the number of attempted answers: trivial (up to 50), moderate (51-500), heavy (above 500).

Equivalence of Computer and Paper-based Threat Memorization Tests

Two studies were conducted to determine the reliability and validity of computerized threat memorization tests given by computer in comparison to paper-based tests of the same information. The reliability was the extent to which the two alternative testing methods were measuring the same semantic knowledge, and the validity was the relation of the scores to an external criterion. The CBMS FLASHCARD game technique was employed by Federico and Ligget (1989) with 75 F-14 and E-2C crewmembers given factual tests on classified information about Soviet surface, subsurface, and air platforms. The computer and paper-based measures were found not to differ significantly in reliability, while the computer test showed somewhat greater validity in discriminating the experience level of the pilots and crewmembers (e.g., flight hours). Student ratings of degree of confidence in their judgments did not differ between the tests. The CBMS PICTURE game technique was used by Federico (1989) with 83 student pilots and radar intercept officers from an F-14 squadron for recognition testing of Soviet and non-Soviet aircraft silhouettes. Computer-based and paper-based measures of recognition test scores were not significantly different in reliability. Discriminate validities were about the same in distinguishing between students who were above or below the mean in their class grades. Student ratings of confidence in their judgments were slightly higher for the paper-and-pencil test than for the computer test. Overall, these two studies indicate that computerized tests of threat facts or recognition are generally no different from conventional paper-based measures.

DISCUSSION

The prevalence of computer-based instruction changed substantially in the Navy's education and training community during the life of the project. Lessons learned emerged from this period of technological change during the development of the CBESS system. This section summarizes the project's accomplishments, test site experiences, observed patterns in the development of CBI, the type and quality of instruction developed, problems encountered, components of successful endeavors, other qualitative descriptions, and implementation issues.

Accomplishments During the Project

The overall accomplishments of the project in addressing the original needs for the effort may be summarized as follows:

1. This effort systematized a diverse set of software into one standard system which had previously been prototyped in various programming languages for different hardware platforms using divergent standards. The resultant difficulty level of the system obviously decreased relative to the prototypes from which it was derived. This work resulted in formal authoring tools which moved the authoring of computer-based instruction from the realm of programmers into that of instructional developers.

2. Market forces during the life of the project resulted in a fewer number of prominent standard computers, which reduced the need to recode CBI among hardware platforms. The programs were specifically adapted to the Navy Standard Z-248 Microcomputer. The programs were also designed to be reconfigurable over a range of hardware options, such as display cards and videodisc players.

3. The authoring and student programs were designed to separate courseware from executable programs so that the system is reusable again and again to create many varieties of separate instructional courseware lesson files.

4. The CBESS is applicable across many occupational ratings and for many types of instructional content. The potential applications include: remediation, enhancement, refresher, initial primary instruction, repetitive practice, self-study, and as a general supplement to instructor resources. In addition to a set of general purpose computer-based instruction programs, three specialized authoring facilities were developed to reduce development time by assuming certain preconfigured instructional delivery strategies. These specialized facilities provide repetitive fact training, technical vocabulary training, and equipment simulation in the context of locating and replacing faulty parts.

5. The authoring programs developed allow instructional developers to create CBI, making this capability more widely available, and reducing development effort. The system has its own self-contained editors and management interface. The system has been formally documented in 18 user manuals, with separate sets of manuals for authors and students. The government controls the source code and can update it with desired features in the future without licensing fees. Wide distribution of CBESS to Navy commands does not involve usage licensing fees for which independently acquired commercial costs can typically be $1800 to $2000 per authoring station and from $75 to $100 per student station. Such costs are for software comparable to the GCBI package, while the other CBESS packages do not generally correspond to packaged commercial offerings.

6. The incremental development of the system was responsive to user needs through an ongoing program of modifications, updates, and user training. Incremental modification to the system during field tests increased the ease of using the programs and increased system utility with newly added features.

7. The system was successfully used by developers in creating deployable instruction. The skill required by the programs generally assume some prior basic operating knowledge of computers and instructional design. The GCBI programs provide an acceptable degree of difficulty relative to other systems with the same degree of features and flexibility in weighing the trade-off between power and ease of use. The LSCAI is a fairly easy system to use in its circumscribed technical vocabulary domain. The CBMS is also a fairly easy system to use for repetitive fact quizzing; its difficulty increases with the size, complexity, and unique tailoring of the domain the user wishes to codify in a database. The EPST is the most difficult and least used CBESS program because it embodies complexities of simulating equipment. The management interface grew out of identified test site needs for a single overall student interface, significantly reducing problems of accessing large numbers of diverse lessons.

8. A variety of instructional development activities produced lessons that remain in use at various Navy training commands. Four of the CBESS packages were used in substantial development efforts and the CBESS is a regularly used instructional media tool at test site commands. A catalog of instruction created with the CBESS documents these finished products (available from the first author). These lessons are applicable across many ratings and most often provide drill and practice, remediation, and review of material generally addressing specific training objectives supplementing larger bodies of regular course material. In the intended application environments, CBESS successfully provided general management support, automated instruction supplementing instructors, and contributed to the reduction of student attrition or increase in student performance. Success was indicated by its regular use by students and instructional managers, and by the desire of test-bed sites that it be continued or expanded, and supported in the future. The potential for developing new lessons is provided by the CBESS, but its support requires attention.

Observed Patterns and Lessons Learned

When is CBI Appropriate to Use

Early in the project, misconceptions of the role of CBI revolved around the idea that CBI might be the primary source of instruction and even eliminate the need for instructors. Such conceptions were quickly lost with more experience in actually using this medium, and replaced by views of using this medium as a supplemental management tool.

The initial decision to use CBI requires a global determination as to whether to computerize the instruction at all, after having examined what portions of the instruction will yield beneficial investments. These are important considerations because CBI development requires additional effort as compared to conventional paper-based instruction. Concern with curriculum conversion costs is accompanied by a determination of whether the conversion would have all the capabilities of existing methods. Thus, a preliminary determination must be made as to *when* CBI is the appropriate media, and *what* should be computerized before steps are taken as to *how* to develop the instruction (Wetzel et al., 1987a, 1987b). The media selection process is needed since there may be no point in mere automation unless some benefit to using CBI can be identified.

CBI might be better thought of to deliver part--rather than the whole--of many training courses and so selected CBI applications should be identified on the basis of whether they improve teaching of specific training objectives when integrated with conventional instruction. Some practical reasons for using CBI might be that CBI offers a learning capability not possible with conventional methods, reduces costs compared to higher fidelity trainers, supplements instructor resources by automating instructional objectives with routine or rote features, provides remediation and review for problem areas in conventional courses, standardizes instruction over many sites, provides remote site and refresher training, and saves time because it is individualized. Reviews of CBI studies have generally shown positive effects for instructional effectiveness, time savings, and positive attitudes toward the use of CBI (Kulik & Kulik, 1987).

Project test sites generally evolved along the lines of these prescriptions, but without explicitly documented decision criteria. The best systematized decision approach to selecting material to be computerized that was encountered during the project was provided by Kaiser (1989). Future efforts should use decision criteria that consider student throughput, level of training, course stability, importance of the course, attrition/setback rates, an analysis of the type of training objectives employed, and an evaluation of software tools appropriate to the requirements of the instruction.

Types of Instruction Developed

The computerized instruction developed at project test sites generally addressed specific objectives or topic areas. Review of the tables in this report that summarize the instructional topics developed shows that often they were for drill and practice applications, review of problem areas, and remediation applications. Thus, the instruction developed was characterized as a selective supplement to existing instruction in which it was judged appropriate for the specific content.

The CBMS games are one instance of drill and practice involving the memorization of large numbers of facts. The flash card technique was already a common paper-based method employed by students and the CBMS automated this tedium. The success of the technique increased to the degree that the databases matched the instruction to be tested. Classifying the TAO CBMS databases increased its value to the students who then used it more frequently. Other prominent instances of drill and practice were with mathematics, electrical circuit calculations, and various picture quizzes such as signal flags, electrical symbols, and even heavy equipment operator hand signals. The LSCAI technical vocabulary lessons are best characterized in terms of their introductory familiarization and remediation applications. These include reviews of basic study skills, electrical vocabulary, operations specialist vocabulary, fuel valves, and various other basic electricity information.

The remediation techniques employed by various sites fell into several types and are worth reviewing as a guide to future potential applications. A first application technique was in courses with a primary emphasis on remediation. In these instances, classes were convened based upon previous identification of deficient skills, and students were given the opportunity to qualify for other courses (e.g., the JOBS program in Memphis and the OS A-school in Dam Neck). In these applications, the remediation was given prior to beginning another course of study. A second technique was providing CBI as a form of structured practice in lieu of self-study with paper-based materials that students might or might not use effectively. This technique was shown to reduce retests and improve test performance in the Memphis JOBS program. A third technique involved providing CBI for selected topics as an entire remediation module in lieu of an instructor. This application is appropriate for relatively rote reviews of definitions and concepts that require little interaction with an instructor. The fourth and most frequent remediation technique was providing CBI as an adjunct to a regular course of study. These applications generally employed a separate computer laboratory serving several courses, and sometimes operating at night for self-study. Students who attended either volunteered because they recognized their own deficiencies or were assigned to attend by an instructor to remediate specific problems and to avoid the possibility of a setback. The various CNET Model Schools illustrate this application; for example, positive findings on attrition and setbacks were obtained at the EM A-school Great Lakes.

Taken together, these uses of CBI for remediation emphasize the idea that a major use of CBI is as a management tool to supplement instructor resources. Since the intent of much of this instruction was for specific secondary purposes in the context of the schools where it was developed, use of the material may not be applicable for export as a complete course of primary instruction. Thus, the exportability of CBI developed in schoolhouse courses for remote-site or shipboard training, or for reserve centers should be considered in light of the original intent of supplementing larger courses with CBI addressing selected objectives.

Quality of Instruction Developed

A noticeable variability in the quality of the instruction developed at test sites was observed. The quality observed could be categorized in terms of the human computer interfaces created, the efficiency of the approach used, and the general instructional quality irrespective of whether the material had been computerized or not. These observations generally lead to the conclusion that the variability reflected the instructional development experience of the developers and the length of time a user had been using the programs. This conclusion directly implies that achieving high quality CBI requires instructional developers who have been dedicated to the task and who have some degree of training or experience in CBI.

Variability in the observed quality of the developed CBI echoes many observations on the quality of instruction developed for conventional paper-based instruction. The developed instruction often included common errors in spelling, in clarity of exposition, and in the use of previously undefined terms. Exposition clarity becomes somewhat more important in CBI when limitations in the screen size control how much information can be presented on a given screen. A consideration unique to CBI is when ideas overlap between screens and earlier material must be reviewed. Potential solutions are providing menus from which the shortened lesson segment can be reviewed multiple times or providing a backup facility, which is sometimes confusing in long lesson segments. Another common shortcoming was that instructional objectives were not stated at relevant points in the instruction. When objectives were stated, they ranged from simple labels of lesson topics to statements that clearly identified required prior enabling knowledge and the terminal objective or skill to be acquired. In some instances, the test items did not match the stated objectives or did not provide a sufficient range of practice to cover the desired terminal skill. At the end of longer lesson segments, summary statements of the important points were often needed, particularly when a quiz followed. Graphics often required revisions to increase the clarity with which they conveyed the intended objectives. In a few cases, reviews consisted of presenting material that might just as well have been read in a book.

Judgments on instructional quality take on a new flavor when instruction is computerized. Computerized instruction allows new features not possible with conventional instruction and, at the same time, brings additional new development concerns. Most of those concerns revolve around human interface issues. For example, feedback to a student who has given an incorrect response can be uninformative and not corrective enough to change future behavior. This is of great concern in computational problems, where problem steps should be elaborated in worked out form. As with the need to review previous screens because only so much information fits on a screen, there were often needs for the availability of relevant reference materials, such as tables of data. Observations of instruction created by subject matter experts (SMEs) often revealed certain rough edges with regard to the general instructional quality issues noted above, and in qualities specific to CBI. For example, some instruction developed seemed overly didactic or punitive in the sense that long lesson segments were created in which students were not allowed to exit or return to earlier menus without having to abort the lesson or turn off the computer.

The lack of elegance in some of these lessons could be solved by building in features to force or guide the development of easier interfaces that allow students greater control to exit or review material. Differences observed among the developers were most obvious when developers had greater control over the interface and were minimized to the extent that the programs predetermined the interfaces. For example, with the LSCAI and CBMS programs, developers provide information in database form which is then dynamically assembled in standardized interfaces as the programs run. The general computer based instruction (GCBI) programs allow users more flexibility in designing the interface, which make differences among developers more obvious. This variability can be reduced to the extent that "authoring" programs continue to evolve effective templates that standardize student interfaces that remove access from the developer. The nature of more complex lessons will still require lesson design skill of developers where their experience can reflect variability in the quality of the lessons.

A final observation concerns the learning curve associated with acquiring software skills. As developers continued to use the software, the efficiency of their techniques increased. These techniques included reusing previously developed components, such as graphics and question templates, which needed only small changes to make them into new instances. Even more advanced generative techniques evolved as developers recognized that a practice situation could be optimized by reusing a single template, such as with math problems which need only new numerical values for the question and answer each time the question is presented.

Revision cycles were a positive feature in achieving instructional quality. Like all instruction, a revision cycle is required after both knowledgeable experts and students have tried out the instruction. Revision cycles are more of a concern with CBI because subtle interface features may only come to light after trials have exposed the many combinations of answers attempted by students. In addition to providing acceptable alternative answers for unforeseen student answers, other features often requiring fine tuning include process control, feedback, and help screens.

Role of User Experience in Creating CBI

Several elements of expertise are needed to achieve quality in the development of either conventional paper-based or computer-based instruction. Compared to conventional instruction, CBI development requires an understandably greater amount of effort and expertise. The increased technical demands become more noticeable when attempted by inexperienced developers. While programmers are equipped from a technical standpoint, they may lack the background of an expert in the specific content of the instruction to be developed, and neither may have experience in instructional matters. Thus, good CBI generally requires either a team of individuals or an individual with a combination of subject matter, computer, and educational skills.

The three experience factors in developing CBI form a matrix with degrees of difficulty of the instructional content itself, the difficulty of the desired type of instructional delivery, and the user's experience level. First, the degree of technical familiarity with the content domain to be taught has always given rise to the need for an SME. Second, the difficulty of developing CBI varies with the complexity of the instructional material and the sophistication of the desired lesson techniques. The challenge presented to the CBI developer varies with the instructional material, which can range from complex simulated equipment requiring high fidelity to simpler verbal tutorials embellished with graphics and quizzes. Commonplace requirements in CBI can include the following: tutorial techniques allowing repeated review, elementary quizzing, generating multiple choice foils, accepting alternative typed answers, general or very specific feedback to the student, homogeneous instruction to permit a database sampling approach, the complexity of the required graphics, complexity in arranging menus and branching schemes, the desired scoring scheme, and the extra steps in preparing for videodisc production. Third, user experience varies widely from those without any basic computer experience at all to those with some experience in operating programs from prompts, manipulating files, knowing about file system organizations, changing directories, and word processing or text editing experience. Some users had some elementary control or programming experience with system script files or even the common BASIC programming language or the like. Very few users had prior experience with a CBI authoring system, and user experience was a significant factor in learning CBESS. Authoring systems generally involve several computer skills such as text editing, graphics editing, and general lesson assembly editing. The average training session was about a week for a given CBESS package, with the skill in one transferring to learning a second. The value of CBI authoring systems is in the attempt to compensate for the diverse set of skills required with a specialized instructional development interface where the most prominent missing skill is the content expert. The LSCAI and CBMS take this specialization one step further in the well circumscribed fact and definition learning domains.

In summary, training users to create CBI lessons showed this medium to add computer skills to the conventional equation requiring both subject matter and instructional development expertise. In general, both instructional developers and SMEs needed to acquire some form of basic computer operating skills. However, subject matter expertise played the same role it did before CBI entered the skill equation since instructional development expertise is still needed. Concerns with required skills are common to most software and this triangle of experience was as common to CBESS as to commercial authoring systems with comparable features.

Problems and Components of Successful Endeavors

Despite numerous differences among project test sites, generalizations emerged about the components leading to a successful endeavor. The best overall characterization is that success was related to having devoted resources to the development and maintenance of the CBI effort. Larger operations were generally more successful because more existing resources were available to reallocate.

Hardware was a much valued resource early in the project because microcomputers were then scarce and only beginning to become widespread through standard contracts. Late in the project, the growing bulk of instruction gave rise to the needs for larger hard disks and concerns with the applicability of networks.

One of the most important components of success was the availability of sufficient staff personnel resources. These individuals were needed to actually develop instruction, to provide subject matter expertise, to identify specific instructional objectives that would benefit students, to review the products, and to update it as needed. Prior computer and instructional development experience predicted success. At smaller commands, single individuals sometimes could make or break the effort.

The ultimate delivery of the instruction to students requires management of a computer laboratory. This management involves computer and facility maintenance as well as direct interface with students. Whether a computer laboratory manager or an actual instructor, student guidance is needed to direct students within the laboratory and to initially direct them to the laboratory when their need is apparent. While evaluation of the effect and usefulness of the computer laboratory were often desired, it was often lost in the midst of day-to-day operations at a site. Remote evaluation of CBESS was often difficult to achieve when it was mixed with CBI from other sources (cf., McCormick & Jones, 1990).

All of these factors of success were prominent with the CNETCHTRA instructional development team in Memphis and at the Great Lakes Model School effort. The stability of development or delivery teams was important in maintaining continuity and bringing accumulated experience to bear on new endeavors. The success of the CNET Model School effort resulted from higher levels of attention given to schools in terms of: (1) providing hardware, (2) providing expertise in guiding instructional development, (3) funding or reallocating personnel resources to perform the work, and (4) identifying applications selected for the greatest investment benefit. The resultant programs developed could not have occurred without such institutional backing.

Resource and Personnel Availability. The identified elements of success predictably have counterparts in identifying the problems encountered. While the project provided hardware to many project test sites, notable problems encountered included the availability of other resources. The most prominent recurring problem was the availability of instructional developers and the difficulty of replacing previously trained individuals who had left. While not an uncommon organizational problem, the detrimental effect to a CBI effort is that new untrained individuals need to master one or another of the components of subject matter, instructional development, and computer expertise. Updates to the CBMS databases were a problem because local personnel were available as subject matter experts and not as instructional developers, leaving them dependent upon project personnel. Computer laboratory support was only a problem to the extent that it involved new development and skill acquisition. In some cases, the computer laboratory was seen as the effort of another division, so the involvement of instructors with the operation was problematic. In such cases, instructors become involved to the extent that the CBI directly supplemented their own instructional needs.

Curriculum Stability. Most instruction eventually needs life cycle updating and revision. While some periodic updates are measured in years, others may be required much more frequently. Curriculum stability was identified as a problem area by 39% of the instructional managers surveyed by Wetzel et al. (1987a). For example, curriculum updates arise when equipment used in the fleet changes. Curriculum stability becomes a problem when personnel to update the instruction are not available at a command, when they are not the original developers, or when new local personnel must learn the development system. Even though authoring systems now distance developers from low level programming, curriculum updates to CBI still require more expertise than conventional instruction.

The most problematic instance of curriculum stability encounted was with updating intelligence training applications. Threat parameter information in the CBMS databases needed to be kept current so that students will continue to find the memorization practice of value for their assigned studies and course tests. The threat parameters in the TAO course databases need updating as frequently as two to three times a year. The CBMS threat databases are used by the TAO course, by intelligence officers and specialists at NMITC, and by various other commands receiving mailed copies. During the project, these commands were unable to provide instructional developers to update this CBI. The existence of multiple sites for this type of training suggests that its updates might be most efficiently maintained by one central location.

CBI is Still an Emerging Technology

Although CBI became more widespread and easier to use during the period of the project, the technology still continues to evolve. Management of the technology will also still need attention as it becomes more common place among other existing media tools.

General ease-of-use issues will continue to be of concern while computer technology is still an unfamiliar domain to many individuals. Management support infrastructure resources become needed for effective organizational programs, lesson development, training, and establishing or maintaining computerized learning laboratories. As the technique becomes widespread, large bodies of instruction must be managed in a systematic way in terms of development, control, librarian functions, and distribution to regular users or responding to new requests. The availability of up-to-date software tools involves continued development whether it is with the CBESS or other software. While microcomputers were once a scarce coveted commodity, the rapidly growing number of them used as training platforms drives our attention to maintain an edge on technological opportunity.

Beyond the practical reasons growing out of widespread availability, CBI technology itself is also changing in many ways. Evolving hardware technology leads to program modifications to adapt previous software investments to new computers and their associated hardware. For example, during the project the IBM-PC emerged as a hardware standard, which gave way to the Navy Standard Zenith Z-248, and recently to the Desktop III contract's UNISYS computer. Each of these hardware evolutions used a newer graphics adapter card. New computer standards will also continue to add newer image technologies, devices for video overlay, networking, and optical mass storage. Standards for coping with such hardware evolutions are being developed as part of the Office of the Secretary of Defense (OSD) Portable Courseware project (PORTCO) (cf., Thomason et al., 1990). Such work will offer relief for small computers with increasing large programs accommodating many current and future device drivers as they accumulate over time. This work also addresses desires for special purpose interfaces that often lead to the creation of special purpose versions that reduce program sizes (e.g., touch screens, voice technology, and calculators). The often specialized hardware evolving from video and optical technology is of continuing concern, with the development of new high definition television (HDTV) standards promising new generations of technologies yielding higher resolution training applications. The potential also exists for other operating systems to emerge as widespread standards that may reduce some current microcomputer limitations and may cause program adaptations. Since CBESS currently has a UNIX operating system variant, the evolution of training platforms to this operating system may be made be possible with a short conversion effort adapting CBESS to the host graphics device.

The development of the CBESS yielded several lessons learned. The initial design of a comprehensive system wrestles with the opposing needs to produce products and to develop designs robust enough to survive future needs which are difficult to foresee. The systematic planning and design document phase of the present project required action to transition the effort to production by establishing a cut-off point for design planning. The difficulty of some subsequent modifications addressing user needs were related to original design decisions. Analysis of the software development records from the second in-house development phase indicated a constant proportion of effort in general software engineering overhead, which reflects the systematic nature of the CBESS. While a larger undertaking, integration in an interrelated system benefits from standardized modules reused in many programs. The effort of developing a comprehensive system is offset by the government control of the CBESS source code in adapting to future needs. The benefits of controlling the direction of the system include the ability to maintain backward compatibility so that previously developed lessons continue to be operable. Test site trials allowed incremental system development responsive to the needs of users actually creating Navy instruction. Among the new features of interest to users, interface issues were prominent. These point to future evolutions in interfaces and sophisticated routines encapsulating expertise that can continue to reduce the entering skill level that developers need.

Infrastructure for Computer-based Instruction

As an emerging technology, CBI is still in the process of evolving a support infrastructure. This need is being prodded by the increasingly widespread number of available low cost microcomputer training platforms. By way of analogy, audio-visual technology has been widespread for years in the Navy with formally designated existing laboratory facilities and personnel staffing. This already evolved support structure provides training commands with local support from established specialists that supplement the training mission with skills not possessed by instructional developers or instructors. The Navy has no counterpart specialty to whom instructors can turn for assistance with CBI. The most common resource is from an ADP group, where a background in instructional matters is lacking. This lack of CBI specialists reflects an evolving technology based in the research and development (R&D) world or the realm of specialists obtained as consultants or on contract. An initial seed for a CBI support infrastructure might be to create a regional staffing arrangement for commands that cannot support individual specialists.

A similar situation has existed in the civilian school systems, where the trend has been toward the evolution of a computer resource teacher to assist the rest of the staff (cf., Schofield & Verban, 1988; Walker, 1986). This specialist provides support in media selection, in management and maintenance of the computer lab, and in guiding those without computer experience. This position has evolved because many teachers have little time to devote to acquiring general computer experience and may be technically unprepared to manage the resource and troubleshoot problems. They also have little experience in sources for selection and evaluation of potential commercially available instruction, let alone the development of the computerized instruction itself. Thus, specialized technologies have created a need that has led to the evolution of a specialist on whom the other staff can depend.

Implementation and Life Cycle Maintenance

Implementation attention and continuing life cycle management support are required to realize the benefit of previous investments in the development of CBESS. Software engineering efforts routinely result in post-project requirements for software maintenance updates. Likewise, CBI courseware requires life cycle updates, just as does conventional paper-based instruction. These "software age" responsibilities are a natural consequence accompanying the infusion of CBI into Navy schools. In joining the trend of computerizing instruction, subsequent life cycle maintenance may not have been initially foreseen by the naval education and training community. Costs of updating electronic media are necessary for software and courseware whether its source is the government-controlled CBESS or other commercial systems. The components of life cycle maintenance (discussed below) are general to computerizing instruction and are directly implied for the post-project success of the developed CBESS.

Software Maintenance

Software life cycle maintenance is routinely required to maintain the viability of programs on new host devices and to maintain the investment in previously developed instruction so that it can continue to be run or updated. The frequency of software maintenance updates is related to the rapidity with which new hardware or operating system standards are widely adopted. During the life of the project, NPRDC maintained the master archival copy of the software source code, updated the programs to extend them to new equipment employed by users, added requested features, and updated the user manuals. The software maintenance function involves high level software engineering skills that dictate specialized R&D work. Such work should not be conducted in the absence of direct contact with users. In addition to the current users at project sites, various other sites use the software and represent an installed user base for which maintenance and support is needed. In addition to the maintenance of CBESS, potential increments to the system include enhanced management features for recording usage at schools, further increases in the ease of interfaces, and the adoption of new pending standards from the courseware portability initiatives.

User support

User support, at a minimum, consists of distributing software, manuals, and maintaining them in stock. Interaction with users is required in responding to requests and in providing consulting accompanying distribution. The use of any product also implies some form of consulting which is appropriately tailored to the intended instructional development purpose of the software. Training sessions at a central location or through site visits are a common mechanism of reducing individual learning curves. In an environment with routine personnel rotations, training support is an increased need, particularly when instructional development expertise is a component skill. Many instructors will remain consumers and have needs for resource specialists in this technology. As updates to a software system are accomplished, users must also receive updated manuals and new release information. CBESS distribution, training, and user support were provided from project resources and will end unless a transition effort takes over.

Courseware Maintenance and Threat Database Updates

Two concerns with the maintenance of developed CBI courseware must be addressed as the project ends. One is the specific case of the rapidly changing CBMS threat databases and the other is the general systemic case of accumulating courseware needing maintenance.

Large bodies of instructional courseware need to be managed in systematic ways. Updating courseware to maintain its currency will be required just as with conventional paper-based material, but the resources needed to do so may be somewhat greater. Courseware may be used in more circumstances than in a single schoolhouse, such as in the recurrent concern with distributed on-board and refresher training. Reusing courseware is cost effective, but it leads to centralized librarian functions and distribution. Bodies of courseware will certainly accumulate over time and their systematic maintenance should be addressed in advance. Similar maintenance and revision control functions evolved during the project as CBESS test site instruction accumulated.

The problem of periodic updates required of most instruction is exacerbated in the specific instance of the CBMS threat databases developed and maintained by the project. These databases contain threat parameters on Soviet, U.S., and other nations' platforms. They are currently in regular use at several sites, most notably at the NMITC Dam Neck, NAS Oceana, and the FCTCPAC TAO course in San Diego. The TAO databases were specially constructed to match the tests given to TAO students and converting them to classified form increased system usage substantially. This application could be exported to the second TAO course at FCTCLANT and to SWOS Newport, where the same or very similar content is taught. This instructional domain will continue to be of interest and should be recognized as deserving special attention. The continued interest in this application is suggested by the fact that the unclassified CBMS databases were requested most frequently for mailed distributions, with interest in future classified versions being a common comment. After developing the programs, databases, and a videodisc and having installed them at operating sites, it would be unfortunate to allow the databases to go out of date and the systems to fall into disrepair. The requirement for updates arises from the rapidly changing database content and the lack of local test site instructional development expertise. At a minimum, funding is needed for an instructional developer to update facts in the databases. A central maintenance and distribution scheme is the most efficient means to update rapidly changing content with wide multi-site interest.

New Instructional Development

Development of new instruction grows out of user requirements and funding. The CBESS should be required for new development efforts which would further justify its life cycle support. Standardization and sharing of developed instruction would result from requirements for its use in efforts such as the CNET Model Schools program. Previous general support for the Model Schools, the CBMS threat databases, and other CBI applications ends with the completion of the project. Hardware for these applications continues to become more reasonably priced. The major expense to be incurred is the development and maintenance of instruction, support of the programs, and adequate computer and instructional development support expertise.

Post-project Maintenance Mechanisms

The ultimate success of the project directly depends upon specific post-project maintenance to support the continued operational use of the system. Funding must be identified and responsibilities must be assigned to an appropriate maintenance organization to support users, train developers, provide distribution, maintain previous courseware, and conduct normal software life-cycle maintenance such as adding desired features and adapting to new hardware. The essential capabilities for maintenance involve both computer science and instructional technologies:

1. Instructional development expertise for user support and training, and for distribution of the software.
2. High level software engineering support to update the programs to new hardware, to enhance the programs with desired new features, and to maintain backward compatibility for existing instruction.

These two specialized skills could be split among organizations at the risk of diluted efforts. It is clear that an activity without software engineering expertise would falter if assigned both tasks. An organization with only computer expertise would likewise fail users with instructional development needs properly connected to the training community. Arguments could be made for originating activities to maintain their software products with the original resident development expertise to avoid unfamiliarity of new system maintainers.

Many support efforts accomplished during the R&D project now become tasks for transitioning to other implementation resources. A broad sponsor base is appropriate because of the widespread applicability to all occupational training missions. A central organizational activity should be created or an existing activity assigned this responsibility. It should be funded from and for the widest base in the Navy. Achieving this funding base requires efforts to cull funds from higher warfare sponsor levels. Piecemeal reimbursable funding may lack the stability needed to systematically maintain the system. Software tools developed within the Navy for its own use may not be appropriately supported via market analogies of direct support from small users within the government. Many target users of an educational software system are schools with limited resources whose level of notice in competition for sponsor attention may be too low to derive funding. The push of technology should be identified as the rationale for taxing larger sponsors for the benefit of widespread smaller users. One approach to diffusing the burden of implementation resources is establishing a requirement for use of the system in new instructional development efforts, with exceptions allowed for justified special capabilities. This approach addresses recurrent problems of independent efforts leading to unstandardized, unexportable or duplicated local products. The costs for maintenance are offset by the direct Navy control of the system, in achieving standardization over sites and the realization of the investment in years of development.

The implementation of CBESS should proceed at sites such as the CNET Model Schools program. The implementation of the CBESS draws attention to Navy needs for a computer-based instruction technology infrastructure that reaches beyond this particular system. The ultimate benefit of establishing an integrated life cycle management structure is that the missions of the widely dispersed training community will be standardized and guided systematically with a supported system.

RECOMMENDATIONS

The following recommendations are for OP-11 and the Navy education and training community:

1. Continuing life cycle management support should be given implementation attention to realize previous development investments in the government-controlled CBESS. The success of the project directly implies specific post-project maintenance to support the continued operational use of the system and developed instruction.

2. Support responsibilities should be assigned and funding should be sought from a broad base appropriate to the wide number of applicable ratings.

3. CBESS can be adopted as a standard to avoid proliferation of incompatible and nontransportable lessons. Exceptions to the use of government-controlled CBI software should be allowed for justified special capabilities. The implementation of CBESS should proceed at sites such as the CNET Model Schools program.

4. User support should be provided for distributing software, manuals, maintaining stock, and consulting that is tailored to the intended instructional development purpose of the software.

5. Routine software life cycle maintenance should be planned to continue the viability of the CBESS on new host devices and to preserve investments in previously developed CBI so that it can continue to be delivered and updated.

6. Existing instruction in the CBMS threat databases should not be allowed to go out of date and should be maintained and updated centrally because of its wide applicability.

7. The Navy should systematically guide computer-based instruction technology by fostering the support infrastructure required by an environment with inherent personnel rotations and loss of trained individuals. Instructors will remain consumers and need resource specialists similar to those that have evolved in civilian school systems or currently exist in audio-visual support specialists.

8. Computer-based instructional development efforts should employ decision criteria that consider student throughput, course stability, course importance, level and type of training objectives employed, potential of remediation to affect attrition or setbacks, management potentials such as supplementing instructor resources, and basing the selection of appropriate software tools on instructional requirements.

9. Computer-based instruction technology continues to evolve and the Navy should adapt the CBESS to new DoD portability standards and enhance these systems with further reductions in user skill requirements. Although authoring systems distance developers from low level programming, developing computer-based instruction still requires more expertise than does developing conventional instruction.

REFERENCES

CNO Memorandum 102/63-80 of 28 Apr 1980, from Research Development and Studies Branch to Director, Research and development Plans Division (OP-987). Subject: Draft Operational Requirements (OR), with enclosure entitled "Draft OR, Low Cost Microcomputer Training System."

Crawford, A.M., & Hollan, J.D. (1983, January). *Development of a computer-based tactical training system* (NPRDC-SR-83-13). San Diego, CA: Navy Personnel Research and Development Center. (AD-A149 713)

Ellis, J. A., Wulfeck, W. H., & Fredericks, P. S. (1979, August). *The instructional quality inventory II. User's manual* (NPRDC-SR-79-24). San Diego, CA: Navy Personnel Research and Development Center. (AD-A083 678)

Federico, P.A. (1989, March). *Computer-based and paper-based measurement of recognition performance* (NPRDC-TR-89-07). San Diego, CA: Navy Personnel Research and Development Center. (AD-A206 630)

Federico, P.A., & Ligget, N.L. (1989, January). *Computer-based and paper-based measurement of semantic knowledge* (NPRDC-TR-89-4). San Diego, CA: Navy Personnel Research and Development Center. (AD-A202 152)

Halff, H.M. (1987a, May). *An evaluation of the CBESS*. Contractor Technical Report TCN 86-575 for Navy Personnel Research and Development Center and U.S. Army Research Institute.

Halff, H.M. (1987b, August). *Using the computer-based memorization system for threat training*. Contractor Technical Report TCN 86-269 for U.S. Army Research Institute.

Halff, H.M., Hollan, J.D., & Hutchins, E.L. (1986). Cognitive science and military training. *American Psychologist, 41*, 1131-1139.

Kaiser, J.M. (1989). *A microcomputer-based CAI system designed to complement traditional instruction in Navy technical training*. Unpublished Doctoral Dissertation, Nova University, Fort Lauderdale, FL.

Kearsley, G. (1983). *Computer-based training: A guide to selection and implementation*. Reading, MA: Addison-Wesley.

Kearsley, G. (Ed.) (1987). *Artificial intelligence and instruction: Applications and methods*. Reading, MA: Addison-Wesley.

Kulik, J.A., & Kulik, C.C. (1987). Review of recent research literature on computer-based instruction. *Contemporary Educational Psychology, 12*, 222-230.

McCandless, T. (1981, March). *Computer-based tactical memorization system* (NPRDC-TN-81-8). San Diego, CA: Navy Personnel Research and Development Center.

McCormick, D.L. & Jones, P.L. (1990, April). *An evaluation of Model School Program efforts: Electrician's Mate "A" School NTC Great Lakes* (CNTECHTRA Report No. 90-1). Memphis, TN: Chief of Naval Technical Training.

McCormick, D.L., Jones, P.L., & Wetzel, C.D. (1989). PEP: A small training command's answer to a large instructional problem. *Proceedings of 11th Annual Conference: Interactive Videodisc in Education and Training*. Paper presented to Society for Applied Learning Technology, 24 August 1989.

Rigney, J.W., Towne, D.M., King, C.A., & Moran, P.J. (1978, October). *Field evaluation of the Generalized Maintenance Trainer-Simulator: I. Fleet communications system* (Technical Report 89). Los Angeles: University of of Southern California, Behavioral Technology Laboratories.

Rigney, J.W., Towne, D.M., Moran, P.J., & Mishler, R.A. (1980, July). *Field evaluation of the Generalized Maintenance Trainer-Simulator: II. AN/SPA-66 radar repeater* (Technical Report 80-30). San Diego, CA: Navy Personnel Research and Development Center.

Schofield, J.W., Evans-Rhodes, D. & Huber, B. (1990). Artificial intelligence in the classroom. *Social Science Computer Review, 8*(1). Also reported in *Golem, Newsletter of Technology and Education, 1*, 1990.

Schofield, J.W., & Verban, D. (1988, September). *Barriers and incentives to computer usage in teaching* (Technical Report No. 1 for Office of Naval Research). Pittsburgh, PA: University of Pittsburgh, Learning Research and Development Center.

Service School Command (1990, February). *EM "A" Model School progress report*. Great Lakes, Il: Author.

Shepard, D. (1990). Military update: CBT techniques help more students pass Navy electrician's mate school. *Instruction Delivery Systems, 4*, 19-20.

Thomason, B., Van de Wetering, B., & Booth, R. (1990, March). *A portable courseware architecture* (NPRDC-TN-90-11). San Diego, CA: Navy Personnel Research and Development Center. (AD-A222 797)

Van Kekerix, D.L., Wulfeck, W.H., & Montague, W.E. (1982a, February). *Development and validation of instructional materials for the study management system (SMS) in a Navy "C" school* (NPRDC-TN-82-9). San Diego, CA: Navy Personnel Research and Development Center.

Van Kekerix, D.L., Wulfeck, W.H., & Montague, W.E. (1982b, February). *Development, test, and evaluation of the computer-assisted study management system in a Navy "C" school: Summary report* (NPRDC-TR-82-33). San Diego, CA: Navy Personnel Research and Development Center. (AD-B062 846)

Walker, D.F. (1986). Computers and the curriculum. In J.A. Culbertson, & L.L. Cunningham (Eds.), *Microcomputers and Education* (pp. 22-39). Chicago: The University of Chicago Press.

Wetzel, C.D. (1990, June). Computer based instruction technology. Paper presented at 1990 Meeting of the International Personnel Management Association (IPMA), San Diego, CA.

Wetzel, C.D., Van Kekerix, D.L., & Wulfeck, W.H. (1987a, May). *Characteristics of Navy training courses and potential for computer support* (NPRDC-TR-87-25). San Diego, CA: Navy Personnel Research and Development Center. (AD-A180 609)

Wetzel, C.D., Van Kekerix, D.L., & Wulfeck, W.H. (1987b, October). *Analysis of Navy technical school training objectives for microcomputer based training systems* (NPRDC-TR-88-3). San Diego, CA: Navy Personnel Research and Development Center. (AD-A187 666)

Wetzel, C.D., & Wulfeck, W.H. (1986, April). Low cost microcomputer training applications for Navy training. Paper presented at 1986 Annual Meeting of the American Educational Research Association, San Francisco, CA, 16 April 1986. In R.F. Dillon (Ed.), *Advances in Testing and Training* (Technical Report No. 86-9). Carbondale, IL: Department of Educational Psychology, Southern Illinois University.

Wisher, R. A. (1980, April). *Computer-assisted literacy instruction in Phonics* (NPRDC-TR-80-21). San Diego, CA: Navy Personnel Research and Development Center. (AD-A084 270)

Wisher, R. A. (1986). Generative approaches to computer-assisted reading instruction. In T.G. Sticht, F.R. Chang, & S. Wood (Eds.), *Advances in reading/language research: Vol. 4. Cognitive science & human resources management* (pp 97-106). Greenwich, CT: JAI Press.

Wisher, R. A., & O'Hara, J.W. (1981, May). *Computer-based approach to the Navy's academic remedial training, project PREST: A cost effectiveness evaluation* (NPRDC-SR-81-18). San Diego, CA: Navy Personnel Research and Development Center.

APPENDIX A

CBESS User Manuals

CBESS User Manuals

Installation and operation overview for the Computer Based Educational Software System (CBESS) on a personal computer. By C.D. Wetzel.

Configuring CBESS using the CBESS.CFG file. By D. Wetzel, K. Newbury & M. Lyon.

Miscellaneous CBESS Utility Programs. By C.D. Wetzel & D. Christinaz.

Interacting with CBESS. By R.C. Brandt.

Sequence editor (SE). By R.C. Brandt.

Lesson and interaction editor (LIE). By R.C. Brandt.

Try (Student execute program for computer based instruction created with LIE). By R.C. Brandt.

Instructional management manual. By R.C. Brandt & B.N. Davis.

START: A menu interface to other programs. By C.D. Wetzel.

Computer based memorization system (CBMS): Author and instructor manual. By R.C. Brandt, L.S. Gay, B. Othmer, & H.M. Halff.

Computer based memorization system (CBMS): Student manual. By R.C. Brandt & R.B. Paulsen.

MEM: Student program to administer CBMS games. By D. Wetzel, R. Goodwin, & R. Lawson.

Language skills computer-aided instruction (LSCAI): Author and instructor manual. By R.C. Brandt.

Language skills computer-aided instruction (LSCAI): Student manual. By R.C. Brandt & B.N. Davis.

LSCAI Lesson preparation techniques. By C.D. Wetzel, D. Christinaz, & J. Baldasaro.

Equipment problem solving trainer (EPST): Author manual. By R.C. Brandt.

Equipment problem solving trainer (EPST): Student manual. By R.C. Brandt.

Introductory tutorial for equipment problem solving techniques (EPST). By C.D. Wetzel & D. Croonenberghs.

Catalog of computer based instruction lessons for the computer based educational software system (CBESS). By D. Wetzel & D. Christinaz.

DISTRIBUTION LIST

Distribution:
Chief of Naval Operations (MP&T) (OP-01), (OP-01B2), (OP-11)
Chief of Naval Education and Training (Code 00) (2)
Chief of Naval Technical Training (Code 00) (2)
Defense Technical Information Center (DTIC) (2)

Copy to:
Chief of Naval Operations (OP-059), (OP-091), (OP-092), (OP-11B1), (OP-16), (OP-29), (OP-39)
Chief of Naval Research (ONT-222), (ONT-20)
Chief of Naval Education and Training (Code 02), (Code N-1), (Code N-2), (Code N-3), (Code N-34), (Code N-5), (Code N-6), (Code N-11), (Code N-51), (Code N-54), (L01)
Office of the Assistant Secretary of Defense (FM&P)
Director, Acquisition Education, Training and Career Development Policy, OUSD(A)
Director, Defense Activity for Non-Trad Education Support, Pensacola, FL
Director, Shipyard Training (NAVSEA-072)
Director, Aviation Training Systems Program Coordinator (PMA-205)
Director, Microcomputer Support Department, NARDAC (Code 30)
Commanding Officer, Navy and Marine Intelligence Training Center
Commanding Officer, Fleet Intelligence Training Center, Pacific
Commanding Officer, Fleet Combat Training Center, Atlantic, Virginia Beach, VA
Commanding Officer, Naval Air Technical Training Center, Lakehurst, NJ
Commanding Officer, Naval Education and Training Program Management Support Activity (Code 03) (2), (Code 04) (2)
Commanding Officer, Naval Education and Training Support Center, Atlantic (Navy Campus for Achievement)
Commanding Officer, Naval Education and Training Support Center, Pacific (Navy Campus for Achievement)
Commanding Officer, Naval Air Technical Training Center, Millington, TN (AW-A1)
Commanding Officer, Fleet Anti-Submarine Warfare Training Center, Atlantic (N-12)
Commanding Officer, Fleet Anti-Submarine Warfare Training Center, Pacific
Commanding Officer, Fleet Combat Training Center, Pacific
Commanding Officer, Fleet Training Center, San Diego, CA
Commanding Officer, Fleet Training Center, Naval Station, San Diego, CA (Code 021.2)
Commanding Officer, Naval Damage Control Training Center, Philadelphia, PA
Commanding Officer, Naval Technical Training Center, Corry Station, Pensacola, FL
Commanding Officer, Naval Construction Training Center, Gulfport, MI
Commanding Officer, Recruit Training Command, San Diego, CA (Unit 2441)
Commanding Officer, Service School Command, San Diego, CA (Code 3200)
Commanding Officer, Service School Command, Great Lakes, IL
Commanding Officer, Service School Command, Naval Training Center, Orlando, FL
Commanding Officer, Service School Command, San Diego, CA (Code 3W34)
Commanding Officer, Naval Health Sciences Education and Training Command, Bethesda, MD
Commanding Officer, Naval Training Systems Center, Technical Library (5), (Code 10), (Code N-1), (Code 7)

Curriculum and Instructional Standards Office, Fleet Training Center, Norfolk, VA
Cognitive and Decision Science (OCNR-1142CS)
Fleet Combat Training Center, Pacific (Code 32)
Aviation R&D Facility, PERI-IR
Training Research Specialist, 3400 TCHTW/TTOZ
Naval Education and Training Support Center, Atlantic
Chief of Naval Technical Training, Millington, TN (Code N6), (Code N6221) (2)
Education Specialist, SWOSCOLCOM (Code 012A)
Trident Facility, Curriculum Instruction Standards Office (CISO (Code 02)
Commander, Naval Training Center, Orlando, FL
Commander, Training Center, San Diego, CA
Commander, Naval Training Center, Great Lakes, IL
Commander, Naval Training Center, Great Lakes, IL (Code 019)
Commander, Training Command, U.S. Pacific Fleet, San Diego, CA
Commander, U.S. Army Training Support Center, Ft. Eustis, VA
Commander, U.S. Army Training Support Center, Ft. Eustis, VA (ATIC-ETS)
Commander, Naval Surface Reserve Force, New Orleans, LA (Code 332)
Commander, Naval Reserve Force, New Orleans, LA
Commander, Training Command, U.S. Atlantic Fleet
Commander, Training Command, U.S. Atlantic Fleet (Code 01A)
Commander, Training Command, U.S. Pacific Fleet (Code N-31)
Commander, Fleet Training Group, U.S. Atlantic Fleet
Commander, Fleet Training Group, U.S. Pacific Fleet
COMTACWINGSLANT (Code 60)
NETPMSA, Pensacola, FL (Code 047)
COMTRAPAC, San Diego, CA (Code N212)
Naval Technical Intelligence Center, Suitland, MD
Naval Training Systems Center, Orlando, FL (2)
Naval Training Systems Center (Code 711)
Halff Resources, Inc., Arlington, VA
FAA Academy (AAC934D), Oklahoma City, OK
Institute for Defense Analyses, Alexandria, VA (2)
Research and Development Center (MCRDAC), Quantico, VA
Combat Development Command (MCCDC)
Deputy CG (PM Training Systems)
Commander, U.S. ARI, Behavioral and Social Sciences, Alexandria, VA (PERI-POT-I)
Technical Director, U.S. ARI, Behavioral and Social Sciences, Alexandria, VA (PERI-ZT)
Chief, U.S. ARI Field Unit, Steele Hall, Ft. Knox, KY
Commander, Air Force Human Resources Laboratory, Brooks Air Force Base, TX
Department of Air Force (AFSC), Air Force Human Resources Laboratory, Brooks Air Force Base, TX (AFHRL/ID)
Air Force Human Resources Laboratory (AFHRL), AFHRL/IDC, Brooks Air Force Base, TX
Scientific and Technical Information (STINFO) Office
TSRL/Technical Library (FL 2870)
Director, Training Systems Development, Randolph Air Force Base, TX (Hq ATC/XPRS)
Library, Coast Guard Headquarters

Commanding Officer, U.S. Coast Guard Research and Development Center, Avery Point, Groton, CT
Director of Training, Office of Civilian Personnel Management
Superintendent, Naval Postgraduate School
Director of Research, U.S. Naval Academy
Center for Naval Analyses, Acquisitions Unit
Center for Naval Analyses