ARI Research Note 93-03

Performance Support Technology To Assess Training Effectiveness: Functional and Test-Bed Requirements

Angelo Mirabella, Douglas Dressel, and George Lawton
U.S. Army Research Institute

Automated Instructional Systems Technical Area
Robert J. Seidel, Chief

Training Systems Research Division
Jack H. Hiller, Director

October 1992

United States Army
Research Institute for the Behavioral and Social Sciences

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

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

EDGAR M. JOHNSON
Technical Director

MICHAEL D. SHALER
COL, AR
Commanding

Technical review by

Richard Gebhard

NOTICES

DISTRIBUTION: This report has been cleared for release to the Defense Technical Information Center (DTIC) to comply with regulatory requirements. It has been given no primary distribution other than to DTIC and will be available only through DTIC or the National Technical Information Service (NTIS).

FINAL DISPOSITION: This report may be destroyed when it is no longer needed. Please do not return it to the U.S. Army Research Institute for the Behavioral and Social Sciences.

NOTE: The views, opinions, and findings in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other authorized documents.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
1992, October

3. REPORT TYPE AND DATES COVERED
Interim Jul 90 - Jun 91

4. TITLE AND SUBTITLE
Performance Support Technology To Assess Training Effectiveness: Functional and Test-Bed Requirements

5. FUNDING NUMBERS
63007A
795
3310
H1

6. AUTHOR(S)
Mirabella, Angelo; Dressel, Douglas; and Lawton, George

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Institute for the Behavioral and Social Sciences
ATTN: PERI-II
5001 Eisenhower Avenue
Alexandria, VA 22333-5600

8. PERFORMING ORGANIZATION REPORT NUMBER
ARI Research Note 93-03

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
--

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)*

For this report, we analyzed Training and Document Command (TRADOC) documents to determine what portions might be included in a performance support technology system for training development. The focus of this review was on the performance evaluation of the design phase of the Systems Approach to Training (SAT) and its automated version (ASAT). The review showed that a combination of procedural guide (i.e., job aid), tutorial, and test production tools could help implement TRADOC policy. A preliminary review of the job aid literature, however, indicated many gaps in our knowledge about how to design and combine performance support tools. Nonetheless, it provided some concrete ideas of how to design effective performance support technologies (PST). We used these ideas to build a demonstration, computerized job aid/tutorial for constructing military occupational specialty Self-Development Tests (SDTs). Content for the demonstration product was derived from the SDT Writer's Workshop at the Quartermaster School. A hard copy version of the demonstration is included as Appendix J.

14. SUBJECT TERMS

	Testing
Job aids	Training automation
Instructional design	Performance support technology
Training development	Criterion-referenced testing evaluation

15. NUMBER OF PAGES
265

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

FOREWORD

The U.S. Army Research Institute for the Behavioral and Social Sciences (ARI), in cooperation with the U.S. Army Training and Doctrine Command (TRADOC), conducts research on how to design effective training products and develop them efficiently. Since 1984 we have worked with the Quartermaster School in a Training Technology Transfer Program (TTTP) to test, in operational settings, ideas and prototype technologies emerging from technical research. The emphasis to date has been on training designs for logistics training in the school setting. Many successful projects have been completed and implemented at Fort Lee.

This report represents an emphasis on developing training products efficiently with the aid of computer-based performance support technology (e.g., job aids, job-embedded tutorials, and training production tools). It also represents a shift in focus to products for export to units. The report provides a foundation for conducting research on performance support technology issues by examining TRADOC's needs for such technology and designing a test bed for studying these issues. The results of the report were briefed to personnel of the Directorate of Training Developments in October 1991. At that time, we delivered a demonstration performance support technology program for producing Self-Development Tests.

Preceding Page Blank

ACKNOWLEDGMENTS

We would like to thank our contractors, Statcom, Inc., and Planning Systems, Inc. (PSI), for providing outstanding support on this project. Ambrose Goichochea of Statcom managed the contractor efforts and provided helpful information, advice, and reports. We are especially indebted to Fu Li of Statcom and Tracy A. Dunlap of PSI. They did a superb job of analyzing candidate computer programs for inclusion in the demonstration performance support technology (PST) system. The system was a major outcome of the project. They made excellent recommendations for designing and constructing the PST. Tracy Dunlap programmed the tutorial and job aid portions of the PST demonstrator. She linked these with a test production tool, creating an integrated package. We have incorporated their detailed analyses in the appendixes to this report.

We would also like to thank the staff of the Directorate of Training Developments at the Quartermaster School, Fort Lee. In particular, Thurman Walters was instrumental in providing school and Training and Doctrine Command documents that were the basis for a tutorial on self-development test construction. This tutorial was a vital component of the demonstration PST system.

PERFORMANCE SUPPORT TECHNOLOGY TO ASSESS TRAINING EFFECTIVENESS: FUNCTIONAL AND TEST-BED REQUIREMENTS

EXECUTIVE SUMMARY

Requirement:

This research was performed to determine functional requirements and specifications for a performance support technology (PST) system for the performance evaluation stage, design phase of Training and Doctrine Command's (TRADOC's) System Approach to Training (SAT).

Method:

We reviewed TRADOC and Government Accounting Office (GAO) documents on policy and methods for training development and management. From this review, we derived issues and features to include in an experimental PST system. We also did a preliminary review of PST tools and research on how to design and integrate such tools into a system. Finally we identified issues that needed to be resolved in designing a test bed for evaluating alternative approaches to job aiding.

Findings:

The literature on how to design and combine effective performance support tools is incomplete and inconclusive. However, the combination of procedural guide (i.e., job aid), tutorial, and test production tool could help implement TRADOC policy on criterion-referenced performance assessment. We showed in the analysis where and how these tools might fit into the SAT process.

Available PST tools can be combined into an experimental system for pilot and field research. We provide preliminary specifications for the system. Its centerpiece will be a computer-based decision aid developed by the U.S. Army Research Institute for the Behavioral and Social Sciences (ARI) on how to produce criterion-referenced tests per TRADOC regulation 350-7. A second component will be a tutorial based on a QMS workshop for military occupational specialty subject matter experts who write performance tests for distribution to units (self-development tests). A third component will be a commercially available test production tool. We produced a demonstration PST to illustrate how these components would be combined.

Organizational as well as job design issues need to be considered in a comprehensive evaluation of job aiding approaches.

Utilization of Findings:

The results provide a baseline and rationale for producing and evaluating an experimental PST system. They can support further development of the Army's Automated System Approach to Training (ASAT), the Navy's Automated Instructional Management System (AIMS), and the Joint Service Automated Training System (JSATS).

PERFORMANCE SUPPORT TECHNOLOGY TO ASSESS TRAINING EFFECTIVENESS: FUNCTIONAL AND TEST-BED REQUIREMENTS

CONTENTS

CONTENTS (Continued)

PERFORMANCE SUPPORT TECHNOLOGY TO ASSESS TRAINING EFFECTIVENESS: FUNCTIONAL AND TEST-BED REQUIREMENTS

1.0 INTRODUCTION

1.1 Background

a. This report summarizes ground breaking research on performance support technology (PST) for training development (Mirabella, 1991). Research results can help the U.S. Training and Doctrine Command (TRADOC) automate its Systems Approach to Training (SAT). They can also help the Navy refine its Automated Instructional Management System (AIMS). The Navy is particularly interested in the embedded tutorial, a PST concept that we explored in our project (Wulfeck, Dickieson, Vogt, and Cody, 1991). Finally, the results can contribute to ARI'S new thrust into Brigade training and evaluation research.

b. What is Performance Support Technology? Puterbaugh (1990) notes that the traditional job aid (i.e., decision flow chart) is only one of many ways to make workers more productive. Tutorials, automation techniques, templates, easily available data bases, and expert systems provide other ways. The expression PST includes all these terms.

c. Why do we need PST? Army training development is labor intensive, inefficient, and poorly standardized. It demands skills in military specialties and in instructional design. Yet, the subject matter experts (SMEs) who produce instructional materials vary in ability, aptitude, interests, and motivation. Moreover, they lack training in instructional design. Consequently, TRADOC is evolving the Automated System Approach to Training (ASAT) to promote computer-based development and management.

d. To support the ASAT program, we focused on PST research relevant to the Design Phase, Performance Evaluation Stage of ASAT. Our goal was to field test and revise PST emerging from training research and development (R & D). This report summarizes functional requirements for designing an experimental PST system and for developing a test bed. Sections 1.0 - 4.0 deal with PST requirements. Section 5.0 addresses test-bed design issues.

1.2 Overview of the Approach

a. Our first step was to define how PST R & D might support TRADOC and its schools. We also examined how that R & D might serve the evolving Joint Services Automated Training System (JSATS). We did so by reviewing Army, Government Accounting Office (GAO), and JSAT documents on performance evaluation requirements and problems.

b. From these reviews, we extracted issues that PST could help resolve. We also compiled research literature for additional issues and for technical base solutions. Next, we identified available PST as a starting point for building and field testing a prototype system. From these reviews we drafted functional specifications for a demonstration/experimental PST system.

c. Finally we outlined a research program to extend our knowledge about how to better design and construct PST systems for training development.

1.3 Summary of Section 1.0. The Army's resources for developing training are severely constrained. The Army must rely on SMEs with little formal training in instructional design. Performance Support Technology (PST) can compensate for the constraints. But we need further knowledge about how to design PST to meet the Army's needs. We undertook this research to acquire the knowledge. Its results can help the other services as well. To begin, we focused on the Design Phase, Performance Evaluation Stage of ASAT.

2.0 METHOD FOR ANALYZING PST FUNCTIONAL REQUIREMENTS

2.1 Review of Army and General Accounting Office (GAO) Documents.

a. TRADOC Reg 350-7 is the "capstone document governing" SAT. It prescribes "minimal essential requirements" for each SAT phase. Other sources about SAT are Pam 351-13, Reg 350-32, Pam 25-33, and the TRADOC memorandum on stand-alone interactive courseware (ICW) policy.

b. Several documents describe how ASAT is being developed. A video tape, (Media Communications Corporation, 1990) explains the functions of ASAT. The tape then demonstrates menus and templates for entering task analysis data. A formative evaluation of the prototype ASAT has documented user reactions and problems (Science Applications, Inc, 15 June 1990). Follow-up evaluations are in progress. In addition TRADOC is doing a needs analysis to support further development or revision of ASAT tools.

c. GAO (1990) cites problems with training evaluation across the services. The Office of the Secretary of Defense (OSD) agreed to review or modify service evaluation practices. PST is potentially useful for helping to improve these practices and to document OSD compliance with a Congressional mandate.

d. GAO (1989) focuses on performance evaluation deficiencies in General Support Maintenance. The OSD agreed that evaluation methods are unsatisfactory. Here again, PST is potentially useful for helping to improve measurement practices and document quality control.

e. A variety of documents describe potential test beds for evaluating PST alternatives. These documents indicate how the Army currently measures student progress and training effectiveness. From them, we can extract candidate issues and methodologies for a PST system.

2.2 Analysis of Army and GAO Documents

We prepared a table that listed training evaluation issues or requirements, stated or implied (Appendix A). These we referenced to specific pages and paragraphs in the documents under review. In an adjacent column, we identified PST design features that might help meet the requirements.

2.3 Review of Research Literature

In the foregoing analyses we examined Army requirements and outlined solutions. To help meet those requirements, we reviewed the following sources of research on job aiding and performance measurement:

a. Military Testing Association (MTA) conference papers. MTA is the major inter-service platform for exchanging information on SAT-related research, especially research dealing with learning analysis and performance assessment. We reviewed MTA proceedings to identify issues and methodologies and then to get detailed information on research accomplishments.

b. Journal articles, research reports, and texts on criterion-referenced testing and design of performance support technology.

2.4 Review of PST Tools to Meet TRADOC Performance Evaluation Needs

We reviewed a set of PST tools to determine how they might contribute to an experimental PST system. The results of this analysis will be summarized in a later report which will also detail the experimental system.

a. Products from ARI Research

(1) Guidebook for Developing Criterion-Referenced Tests, (Swezey and Pearlstein, 1975). This was developed under sponsorship of the Directorate of Training Developments, Ft. Knox, Kentucky. Though 15 years old, this clearly written document is still an excellent source of well illustrated principles and procedures of criterion-referenced testing. The authors provide many practical rules of thumb to help the reader implement test development procedures and interpret test data. Each chapter includes a pull-out flow-chart on the chapter's procedures and exercises to test the reader's understanding. The Guidebook would be invaluable in developing an experimental PST for test construction.

(2) 4-volume report on work sample testing (Guion, 1979a, 1979b, 1979c, 1979d). Guion has compiled a comprehensive, clearly written theory of work sample testing. In Volume I he explains how different test purposes and methods influence test construction and test characteristics. In Volume II, he compares classical norm referenced testing with modern alternatives: content referenced testing, latent trait (item response) theory, and generalizability theory. Volume III deals with test construction. Volume IV details the principles of generalizability theory.

(3) ARI/HumRRO Decision Aid on Performance Assessment. The Decision Aid is a menu-driven, computerized flow chart for constructing criterion-referenced tests. It also contains help screens and worksheets (Guthrie, 1990b). The aid was derived from the original Tri-Service Instructional Systems Design (ISD) model.

(4) Field Performance Test Battery Report (Cormier, Dressel, and Mirabella, 1991). This a study of the characteristics of an MOS test (76C MOS) that could be used to assess the impact of training and to diagnose training deficiencies. It presents a general model for constructing and content-validating such tests. It was the basis for the next item in this list of ARI products.

(5) Test Development and Analysis Program (TDAP). TDAP is an experimental program to study alternative designs for computerized, test construction job aiding (Appendix B; Mirabella, 1990). It provides a spreadsheet for outlining a test plan, e.g., a matrix of tasks and subtasks to be covered by the test. The cells of the spreadsheet are linked to a data base for storing test items. After an examination is administered, results can be displayed on the same spreadsheet to show the test administrator which tasks and subtasks are giving students the most trouble.

b. Commercially available, computer-based products:

(1) EXAMINER and Microcat (Appendices C and D) are tools for writing, storing, and administering test items. Both contain basic item analysis routines. Microcat also contains routines for doing latent trait (i.e. item response) analysis.

(2) Course of Action, QUEST, and TENCORE are instructional authoring programs (Appendices E, F, and G). Course of Action is a Macintosh Program. But its courseware can be run on DOS computers. QUEST AND TENCORE are DOS programs.

(3) Hypercard and Hypertalk (Appendix H) are Macintosh versions of hypertext. Hypertext is potentially valuable for job aiding technology because of the ease with which the user can explore a large data base of information.

2.5 Summary of Section 2.0

The first step of our approach was to define how PST research could help TRADOC and its schools. From this definition and a review of the literature on job aiding and criterion-referenced testing, we extracted research issues. We also designed and produced a demonstration/experimental PST system for addressing those issues. Finally we outlined a research design and test bed.

3.0 RESULTS OF ANALYSIS OF PST FUNCTIONAL REQUIREMENTS

3.1 Analysis of Army and GAO Documents. We analyzed the documents referenced in Paragraph 2.1. Appendix A shows the data. The left side of the table cites issues and requirements by page and paragraph. The right side outlines how technology can help address the requirement. We abstracted the following requirements:

a. General Compliance with TRADOC policy.

(1) TRADOC has defined the minimum requirements for training development in PAM 350-7. It is now computerizing the PAM in the Automated Systems Approach to Training (ASAT) so the schools can implement its policies more easily. Recent audits by GAO make this development timely if not imperative. The ASAT can help address the GAO complaints and make it easier to cope with future Congressional inquiries.

(2) Performance support technology can make ASAT even more effective for performance evaluation than it might otherwise be. For example, a decision aid can show the developer the steps to take in test construction. With supporting book marking, electronic check marking, and disk notes, a decision aid would provide a clear record of compliance. Tutorials and hypertext help screens could support development activity. But in an initial formative evaluation of ASAT, users were dissatisfied with the tutorials and help screens, so the design of these need to be revisited (SAI, 1990).

(3) A decision aid menu can permit the developer to branch to job aids and tutorials for specialized evaluation requirements within the Performance Evaluation Stage. One branch might deal with development of End-of-Annex and End-of-Course tests. A second branch might lead to guidelines producing Self Development Tests (SDTs). Section 16.0, Appendix J illustrates such a menu.

b. Staff Training.

(1) PAM 350-7 directs TRADOC HQ, the Army Training Support Center (ATSC), and the schools to teach their personnel to execute SAT policy and procedures. This requirement holds for Performance Assessment as well as other stages of SAT. PAM 350-7 prescribes "Assignment-oriented SAT training", and "standardized SAT training for senior managers, middle managers, and trainers".

(2) PST tutorials can cover both types of training, if matched to the needs of specific audiences. PAM 350-7 specifies training developers about to enter Skill Codes 7Q, SQI 6, or SQI 2 and instructors entering 5KSQI 8, or SQI 6.

c. Actions, Procedures, Outputs For Performance Evaluation.

(1) Specific requirements for the initial focus of this project appear in Pam 350-7, Appendix Items A12-A15 and Items A22-A24. A15. "Develop test items" is the centerpiece. But we need to include the preceding task and learning analyses to insure that SMEs produce valid tests for instruction and certification. We also have to include test delivery and data management (A22-A24) to have a complete system for performance assessment.

(2) The HumRRO/ARI Decision Aid is a candidate component for an experimental PST. The aid already covers A12-A15 and A22-A24. But we need to examine more closely how well it does so. We also need to detail how well AIMS and ASAT cover PST requirements. We can then revise the Decision Aid and specify supporting tutorials and automation techniques.

d. Knowledge and Skill Requirements for Performance Assessment Tasks. These are specified in the following items from Pam 350-7, Appendix A.

(1) A12 Perform individual task analysis. The output of this action would be available to the test developer in the ASAT data bank. A prototoye PST system would have to indicate what sources of information to consult for task analysis data.

(2) A13 Develop objectives. Since the procedure is to translate task performance specifications into terminal learning objectives (TLOs), the user needs to understand these terms and their distinction. A13 is a candidate for a tutorial.

(3) A14 Perform learning analysis. The SME is directed to analyze TLOs into Enabling Objectives (EOs), knowledges, and skills. A tutorial can help him write test items for each.

(4) A15 Develop test items. PAM 350-7 prescribes criterion-referenced testing (CRT). But the meaning of CRT and other evaluation concepts are not clear from the TRADOC documentation. PAM 25-33, for example, contains inaccurate or misleading definitions. A tutorial on basic concepts is indicated. Since A15 is the centerpiece of this initial effort, we need to discuss its components as follows.

(a) Determine the types of tests needed for specific objectives. A tutorial can explain different purposes for assessing performance and what they mean for test type and format. Minimally, we need to identify four purposes: check for understanding, drill and practice, diagnosis, and certification. We then need to explain how to match test design with purpose. Test types include part vs whole task assessment, performance-oriented written tests, performance-oriented hands on tests, and knowledge assessments.

(b) Prepare several test items. A tutorial plus a variety of automation techniques would support this requirement. An ideal PST would automate construction of items as much as possible, and rely very little on teaching the SME to build them. The test plan feature of TDAP is minimum since the TRADOC Automated Instructional Management System (TAIMS) includes it. Beyond this a good editor is essential. Spelling and expert-system editors along with item and scenario generators are desirable. The easiest way to construct a system is to provide 'hooks' from the Decision Aid to stand-alone automation programs.

(c) Construct the tests. Presumably what this means is to compile the test from the items prepared earlier. Item banking provided by TDAP, Microcat, and Examiner would expedite this task. Each of these programs automatically compiles tests from the users specifications.

(d) Validate tests. We need a good tutorial to instruct the SME in how to deal with a variety of test quality issues. We also need to provide related statistical analysis techniques, especially criterion-referenced methods. The methods in AIMS, TAIMS and ASAT are traditional, norm-referenced item analysis procedures. The statistical programs in Microcat and Examiner should be examined to determine whether they support a criterion-referenced philosophy.

(5) A22-A23 Plan for and conduct training for staff, faculty, and cadre. Initially, we will focus on embedded tutorials for those SMEs directly responsible for producing and administering performance assessment instruments. Later we should consider the needs of supporting staff.

(6) A24 Conduct training. For the initial task focus this means administering performance assessment instruments, processing test data, and making training management decisions. The PST should deliver tests by computer and hard copy. It should also store, organize, and process results so the SME can clearly answer the questions raised in A15(1) about test purpose.

e. Quality Control (QC). QC pervades SAT/ASAT. It includes test validation. It also includes use of performance assessment instruments to check the quality of training products throughout the SAT process.

f. Inter Agency Cooperative Efforts.

(1) PAM 350-7 prescribes cooperation among TRADOC agencies in developing and managing training. For example, proponent schools and the ATSC will cooperate in managing Individual Training Development (IDT) testing (the substitute for SQT). ATSC will compile and distribute tests from materials provided by proponents.

(2) A PST system can be designed to improve productivity of such cooperative efforts. It can speed-up communication and avoid work duplications. Sophisticated groupware or networking procedures are beyond the scope of the initial focus. However, some simple procedures are possible, e.g., use of floppy disks or even E-mail to transfer test materials.

3.2 Analysis of Research Literature. We compiled a data base of research literature on job aiding and criterion-referenced testing. We organized the testing studies into eight major categories and reviewed the studies. The paragraphs below highlight some of the review findings.

a. Job Aiding. TRADOC has decreed that computer-based job aiding will replace current paper-based methods. Research on job aid design can provide useful guidelines to evaluate and improve computer-based programs.

(1) Among recent efforts are pilot studies by Guthrie et al. (1990a,b). They present a model for designing effective procedural job aids. The authors reviewed the literature on how people use alternative forms of job aids. They concluded that effective job aids use four sources or kinds of information. Each source requires a different type of job aid design (i.e., format). They specified three categories of design variance: types or sources of information (x), job aid formats (y), and the interaction of x and y (xy). Their review sheds light on these sources and what job aid formats are most effective.

(a) Information about outcomes, milestones, or how things work. It's useful to provide up-front information about the outcomes of task behavior or a task overview (Guthrie et al., 1990a, 1990b). For example, subjects react faster to "Turn the left knob when the alpha meter reads 20 degrees" than to "When the alpha meter reads 20 degrees, turn the left knob". They also react faster to "to turn light Y on, press button B" than to "Press button B, to turn the light on." Kieras and Bovair (1984) showed that information about how equipment works is most useful when it's embedded in instructions for how to operate the equipment.

(b) Step-by-step procedures. Overviews don't provide enough information. Step by step information may be necessary. The issue here is how many steps should you read and memorize before turning away to do part of the task. The research indicates that the job aid user should chunk the information. He should read a few steps, perform, then read a few more steps. But it does not specify the optimal chunk size. The ASAT formative evaluations have indicated some dissatisfaction with amount of guidance for some parts of ASAT. But how much is enough? These questions provide a basis for specifying researchable issues.

(c) Self testing and feedback. Guthrie et al (1989) hypothesize that self testing and feedback enhance the effectiveness of job aids. None of the studies they cite, however, deal with self testing in the use of job aids. The value of self testing needs experimental study.

(d) Self-correction. This isn't a separate category, though Guthrie et al treat it as such. It's part of self testing and feedback. Guthrie et al., again state an hypothesis: self correction enhances the effectiveness of job aids.

(2) They also conclude that a critical interaction occurs between job aid format and stimulus-response characteristics of the behavior required by a task. The authors have rediscovered what behaviorists used to call stimulus-response compatibility. Compatibility implies a clear and simple connection between the directions in the job aid and the task steps. For example, written directions are easier than a map for getting from Point A to Point B. Written directions specify the exact behavior. For example, "go one mile on Route 3, then turn left on Route 6". With only a map, you'd have to find Points A and B. Then you'd have to find the connecting roads. Finally, you'd have to translate the information into specific steps.

b. Criterion-Referenced Testing. We've compiled an initial data base of 58 studies on criterion-referenced evaluation into eight categories. The studies are summarized in Appendix I and Warrick-Bell (1991). Their implications for SDT development are outlined below.

(1). General and Basic Concepts of Performance Evaluation.

(a) TRADOC doctrine for collective as well as individual training is based on criterion referenced testing (CRT). So it's critical to define CRT clearly. The need for clear definition is highlighted by a misleading definition in TRADOC Pam 25-33 (Army Training Glossary, Page 16). It's also highlighted by GAO's misunderstanding of CRT in its review of DOD training evaluation practices. Many references agree on what CRT means and how it differs from norm referenced testing. They would take issue with the definition in Pam 25-33.

> "Criterion referenced test
>
> A test that establishes whether or not a unit or soldier performs the task to a preset standard."

(b) The definition confuses the ruler and the use to which that ruler is put. A CRT is a ruler designed to measure some well defined dimension of task or job behavior. All the references we found agree on this. The word 'criterion' in criterion-referenced test refers to the dimension of behavior, not to a cutoff score. Let's say this another way. A criterion-referenced test is one that accurately measures task or job performance. Whether that performance meets a pre-set standard is an important but separate question. A test can have a pre-set standard and still not be criterion-referenced if it does not accurately measure the task or job performance. On the other hand, a test can lack a pre-set standard and still be a good criterion-referenced test.

(2) Domain of Content Specification, Front-End Analysis.

(a) Specification depends on where the test developer is in the SAT pipeline and on organizational constraints. For example, SMEs who write items for Self Development tests must use subject areas, tasks, and performance measures already defined by soldiers manuals.

(b) But specification also depends on test purpose. Purposes include: certification, diagnosis, drill and practice, and tests for understanding. These are not explicit in TRADOC guidance on performance evaluation, except for Skill Qualification testing (SQT) and Self-Development testing. But they are prevalent in the Army's management of training. Therefore, they are candidates for inclusion in a PST system. The system would need to define them and show how they affect the development and use of performance assessment instruments. Guion (1979a) identifies additional purposes: program evaluation and organizational trouble shooting. We won't focus on these, but the emerging PST should be easily extendable to these purposes.

(c) A general approach to specifying content has been developed by Cormier, Dressel, and Mirabella (1990). Fritz (1990) computerized it. And, the Army has incorporated it into the TAIMS. The approach is to lay out a two-dimensional table for tasks and subtasks, and then tie specific test items to each cell in the table. The table could be used after a test as a diagnostic tool. It would show the distribution of test errors across tasks and subtasks. Thus, it would indicate to the instructor or training developer where students were having problems.

(3) Item Writing. Research in item writing has generated principles for constructing multiple choice items. TRADOC has communicated these principles in guidelines to SDT developers. The following rules summarize the principles.

(a) Write performance oriented questions which estimate task ability, not rote recall of isolated facts.

(b) Be fair, no trick questions.

(c) Don't say in 20 words what you can say in 10.

(d) Keep the writing at about a 9th grade level.

Appendix J explains and illustrates these rules.

(4) Test Construction and Administration.

(a) Dressel and Mirabella, (1991) and Warrick-Bell (1991) deal with many issues under this category. But test length and item selection are among the more important issues. Comier et al (1991) indicate that MOS multiple choice tests require about 60 items for adequate reliability. This, in fact, is the number suggested by TRADOC guidelines for SDTs. But CRT relevant statistical methods are available for defining test length. Future R & D on job aiding needs to address which of these methods are most useful for the military services. A PST system should then incorporate the methods and instruction on how to apply them.

(b) Item selection, i.e. item sampling, depends on test purpose. The TRADOC SDT guidelines properly require a broad coverage of each tested MOS with a one or two items per task. Tests developed for use during training, e.g., for feedback or diagnosis would cover each task more thoroughly.

(5) Item Analysis.

(a) Item analysis is used for different purposes in criterion-referenced and norm referenced testing (NRT). In NRT, item analysis is used mainly to generate large variance among individual scores. In CRT, item analysis is used to "red flag" poorly constructed or invalid items. Both judgement and statistics are used to do this. But in using item statistics, the test developer needs to avoid inadvertently creating norm-referenced items. For example, in NRT, an item answered correctly by 10% of test takers would be discarded or re-written.

(b) In CRT, such an item should be discarded or revised only if reviewers decide it's poorly written; e.g., unclear or unrelated to the objective under test. TRADOC guidance to discard SDT items with pass rates other than 51% to 89% is questionable. The resulting SDT may not be effective for either of its two purposes - training evaluation or promotion.

(c) But item statistics <u>can</u> be used in a more routinely quantitative way to reduce biases, improve reliability, and discriminate masters from non masters. A well designed PST for test development should include instruction on how to use such statistics. It should also provide computer programs for applying them.

(6) Standard Setting. Research has shown that standard setting can be used in two ways (Mirabella, Macpherson, & Patterson, 1989). Training managers can reduce the rate of skill

decay by increasing standards (e.g., from the conventional 70% pass grade to 85%). Standards can also identify students who have mastered training objectives. Standard setting procedures are detailed in Dressel and Mirabella (1991) and Warrick-Bell (1991).

(7) Reliability and Validity.

(a) The issue of how a PST system should deal with CRT test reliability and validity is a technical mine field (Dressel and Mirabella, 1991; Warrick-Bell, 1991; Swezey and Pearlstein, 1975). NRT methods are not appropriate because they require high variance among test takers. CRT methods may not be dependable. They may result in biased estimates, rest on questionable assumptions, or require arbitrary definitions of mastery.

(b) Here's a practical solution. Write good, performance oriented items or scenarios. Then carefully sample items across an MOS or component program of instruction. Statistical solutions may be more trouble than they are worth. In any case, carefully examine current service practices and policies. Accommodate them in future R & D on PST. But question them where appropriate, as we did in Paragraph 3.2 b.(5) for item analysis.

3.3 Summary of Section 3.0. From a review of Army and GAO documents we identified Army needs that PST could satisfy. These included compliance with TRADOC policy, staff training, actions and outputs for performance evaluation, skills for assessing performance, quality control, and inter-agency cooperation. Most important PST could help insure that developers comply with training development policy and doctrine. From the literature on testing, we identified Army-relevant issues that PST needs to address. For example, PST must provide guidelines on how to write job-valid, performance-oriented test items and scenarios.

4.0 PRELIMINARY SPECIFICATIONS FOR A PST SYSTEM

4.1 The AT&T Training Test Consultant.

a. AT&T has a very promising test development PST. It is proprietary and not-for-sale. Nonetheless, it provides a candidate model for an Army system (Puterbaugh, 1990). It has the following components:

(1) INFOBASE. Hypertext files on the following topics:

(a) Overview of testing

(b) test selection, design, item construction, and administration.

(c) Reliability and validity

(d) Decision making

(e) Security.

(2) TUTOR. Drill and practice tutorials on:

(a) Criterion-referenced vs. norm-referenced tests

(b) Selecting type of test item.

(c) Constructing multiple choice, matching, fill-in, and open-ended items

(d) Interpreting item analysis data

(3) ADVISOR. A combined expert system and set of statistical programs. The expert system is limited to selecting test type and matching items to objectives. The statistical programs deal with reliability, validity, cut-off scores, and item analysis.

b. Puterbaugh doesn't detail how the system is 'glued' together. But he says the user can move easily from tool to tool. He also reports efficiency increases of up to 50%. The AT&T program suggests features for an Army system, e.g., help screens, drill and practice, and statistical routines.

4.2 An Army System.

a. A production system would address the requirements identified in Sections 2.0 and 3.0. The WordPerfect Tutorial provides a good model for it. The Tutorial integrates self paced, menu driven instruction with a production tool, the WP word processor, in the same screens.

b. A preliminary, experimental version for pilot research and prototype evaluation might be configured as follows. The ARI Decision Aid would be the centerpiece; i.e., the home base program. The aid would link to help screens, an embedded tutorial, and a production tool, either EXAMINER,or MICROCAT. But it would have a 'breadboard' configuration. For example, the decision aid, help screens, and tutorial might appear on one work station. The production tool might appear on an adjacent work station. Integrating the tools into one file, on one station, is beyond the scope of the initial research effort.

c. As the user navigates the home program, he would have easy access to screen-sensitive tools. The tutorial would be a sub program designed for the initial test bed - Self Development Testing. And we would incorporate materials from a workshop given by the Quartermaster School (QMS) to SDT item writers. How we would weave in the tutorial, remains to be determined by research outlined in Section 5.0.

d. Demonstration PST.

(1) We produced a demonstration PST program (DEMPRO) to illustrate the combined use of job aiding, embedded instruction, and a production tool (Appendix J). The DEMPRO is a 'breadboard' since the PST components are 'hooked' together. They are not integrated into a single file like the Word Perfect tutorial. An expanded version, with more detailed instruction and practice exercises, would be suitable for research.

(2) DEMPRO is based on portions of the Harliss instructional model (Mirabella, MacPherson, and Patterson, 1989). It's also consistent with principles of job aiding discussed in Guthrie et al. (1990a,b). First, it tells the SME how to set up a plan for an MOS Self Development Test. It illustrates what a plan looks like and then lets the user create a sample plan using a production tool (Examiner). Next, it explains and illustrates principles for writing multiple choice items. It checks for understanding and then lets the user practice writing items with the help of a computer-based test development program (EXAMINER).

(3) We examined three authoring systems for the decision aid/tutorial portion of the PST (Course of Action, Ten-Core, and QUEST). We also examined three computer-based test development programs (TDAP, Microcat, and EXAMINER). Our goal was to select one program from each category and integrate the programs. With contractor support, we analyzed the programs for ease of use and compatibility with DOS. We also examined the ease with which programs from the two categories could be linked. We chose QUEST for the tutorial/job aid. We chose EXAMINER as the test production tool.

(4) Development of the decision aid/tutorial portion of the Demonstration Program. The basis for DEMPRO was a set of TRADOC documents provided by the Quartermaster School, Ft. Lee.

(a) Lesson Plan for Self Development Test Item Writing Workshop. This outlines a lecture/conference period of 155 minutes (description of 15 slides) and a practical exercise session of 140 minutes.

(b) Memorandum for Distribution, from Deputy Chief of Staff, TRADOC (DCST), "Interim Policy Guidance for the Self-Development Test (SDT)", 11 Dec 1990. This defines policy for SDT development, administration, and management.

(c) Quartermaster School Publication QMSP III-8-A, "NCO Self-Development Test (SDT) Development". This details test development and administration procedures.

In producing DEMPRO, we adhered to the guidelines and procedures specified in the documents above. We did, however modify some details to conform to standard principles of instructional system design.

4.3 Summary of Section 4.0. We constructed a demonstration program (DEMOPRO) to illustrate the major features of a PST system. DEMPRO combines a tutorial and job aid for developing tests with a production tool for constructing the tests. The tutorial and job aid were based on a QMS test-development workshop and TRADOC guidelines. We used the authoring system Quest to construct the tutorial and job aid. We linked these to the production tool Examiner.

5.0 TEST-BED DEVELOPMENT FOR EVALUATION OF TRAINING DEVELOPMENT AUTOMATION

5.1 Introduction

Completion of a PST R & D program requires Army test beds to measure the effects of candidate tools on training development efficiency. Uhlaner (1970) suggested that such test beds should be designed to assess all the variables which could affect performance on a military job. Investigators could then study how individual and organizational factors constrain the design and use of training development technology. Simon (1973) concurred with and quantified this view. He showed that 15 years of piecemeal human factors research on weapon design was worthless. That work, on average, dealt with two variables per experiment and accounted for 16 % of the variance.

5.2 Research Issues: Overview

a. The top level issue is how to design and integrate performance support tools for training development. The tools should fit TRADOC policy and constraints of operating units. Each type of tool, (i.e. decision aid, tutorial, authoring system) has its own set of behavioral issues. For decision aiding, some of these have been identified and examined in recent ARI sponsored research (Guthrie et al, 1990a,b; Swezey, Llaneroz, Perez, and Gittleman, 1991). They are complicated by uncertainties about whether and how to interleave multiple tools into a system.

b. The research literature on integration of PST tools is fragmentary, incomplete, or inconclusive (Curley 1990; Guthrie et al., 1990a,b; Swezey et al., 1991; Stone and Hutson (1984); Porter and Hutson, 1984).

(1) Guthrie et al (1990a,b), for example, hypothesize that decision aides are more effective if they provide feedback on correctness of decisions. But they offer no direct evidence. Stone and Hudson (1984) offer only indirect evidence. Their subjects, given a choice of sources of information, frequently asked for graphic information on what interim products should look like. Stone and Hudson also suggest that hypertext can improve the effectiveness of procedural job aiding. But they did not compare job aiding with and without hypertext. Puterbaugh (1990) describes 'successful' use of an AT&T proprietary test production system which includes drill and practice in writing test items. But he offers no evidence for the success of the system or its embedded tutorials.

(2) Majchrzak (1990) suggests that work place automation does not reach full potential where the need for related training is underestimated. But again, the evidence is not conclusive. Furthermore Majchrzak doesn't show how to provide the training.

5.3 Variables in the Training Development Test Bed.

a. Measures of Effectiveness (MOEs). These are the measures which tell us whether any particular technology is justified.

(1) Development cost. How much will automation cost and how much it will save? If automation reduces demands on training staff or increases the speed of training development, it is partly justified. MOEs include time and dollar expense to produce and administer training. But opportunity costs are also important. For example, NCOs in training development cadres are lost to active units. Automation can reduce the loss. It can reduce the number of people and time to develop training packages.

(2) Effects of automation on behavior of its users. Goicoechea, Stakhiv, and Fu Li (1991) measured agreement among subjects using alternative engineering decision aids. They found significant differences in the reliability of the aids. Swezey et al. (1991) measured task accuracy in electronic troubleshooting. But they also measured how frequently subjects asked for various kinds of information (e.g., verbal vs graphical). Swezey et al. distinguished information seeking behavior of experts from that of novices. A critical effect would be the impact of the job-aided product on training effectiveness.

(3). Consistency or quality of training products. TRADOC has produced extensive checklists for assessing the quality of multiple choice test items. Guthrie et al. (1990b) used an anchored rating form to assess how well test items met standards of instructional design. Their test development job aid did not improve item quality. But their explanations for the null findings provide useful issues and hypotheses for further research. Majchrzak (1988) and Majchrzak & Klein (1987) have shown that PST may have potent effects on productivity or quality, but may act indirectly, through effects on the organization. Their work suggests the need for caution in measuring product effects directly, especially in a laboratory setting. That work also shows that PST evaluations should not be limited to measures of product quality.

(4). Usefulness in helping to meet TRADOC policy on technology and standardization in SAT. To asses this, the research staff would compare policy statements with candidate development tools. TRADOC staff could then validate or revise the assessment.

(5) 'Customer' satisfaction. Research products are effective only if their consumers are satisfied with the products. So, we will study the reactions of training developers and their organizations to automation. Results will help us recommend how to use training development tools most effectively.

b. Independent Measures. Uhlaner (1970) suggested the study of personnel variables, (intelligence, aptitudes, abilities) and work experience, in addition to training variables. He also felt that human and organizational factors should be examined in training research.

(1) Training Development Variables.

(a) We want to learn how the quality of SAT products vary with different levels of development technology. The SAT includes analysis, design, development, implementation and evaluation. Many types of technology can be used in each phase. Computer programs can do parts of the training developer's job. Examples are forms production, test development, and test administration. Computer-based training can enhance the skill of training developers. On-line job-aids and information sources can guide the developer through her tasks.

(b) ARI has developed computer programs which play one or more of these roles. These tools include the Training Diagnosis and Analysis Program (TDAP), the Automated Knowledge Acquisition Tool (AKAT), and the AISTA/HumRRO Decision Aid. The programs may be useful in the Evaluation, Analysis, and Development phases of the SAT. Similar, commercial products may also help automate the SAT. We'll assess the usefulness of the in-house and commercial tools for our test beds studies. We will then evaluate the effectiveness of some combination of these tools in the test beds.

(2) Training Variables.

(a) The technology we evaluate will include training on how to use the methods of SAT. We'll need to ask training design as well as research questions. For example, should we embed tutorials in decision flow charts and, if so, how?

(b) A second set of training issues deals with effects on soldiers' performance of the products resulting from use of development technology. Even if these products meet high standards of instructional design, they may not improve the final product. They have to increase training effectiveness or reduce costs without reducing effectiveness.

(3) Organizational Factors.

(a) Majchrzak and Klein (1987) and Majchrzak (1988) concluded that automation changes how factories and offices work. It changes task and job structures, the personnel system, structure of organization, and informal communication networks. Automation increases work pace, information demands of the job, need for cooperation and coordination between workers, and worker discretion and flexibility. Would similar effects occur at TRADOC sites?

(b) For example, automation may change the numbers and qualifications of people required to do a job. They note that computer-aided design, because of its three dimensional capability, is different from paper-based design. Automation then could increase demands on personnel recruiting, selection, or training. The need for programming, maintenance, and repair can create new jobs while other jobs may disappear.

(c) Automation hardware and software impose more formal organization and operating procedures. On the other hand, they may either centralize or decentralize decision making authority (Majchrzak, 1987).

(d) Finally, automation may change informal organizational structure. It may change the communication networks through new media (e.g., electronic mail), new conversation topics or issues, and the spread of critical knowledge. Similarly, changes in skill concentrations may informally redistribute power (Majchrzak, 1987). While some of these effects will be anticipated, others may not.

(4) Personnel Variables. Individual differences in ability and motivation influence whether people use new technology. These differences also affect how well people use the technology. We need to document the characteristics of training developers. We'll co-vary some of the characteristics in our evaluations of development tools.

5.4 Test Bed: Self-Development Testing (SDT) Program.

a. ARI Mission Focus. Under the laboratory reorganization, ARI will focus on unit training research. A major and increasingly important thrust here is distributed training. This is training developed by proponent TRADOC schools for export to units. SDT is a distributed training product. Its development currently is not supported by advanced job aiding technology.

b. Importance of evaluation. Evaluation is essential at each stage of SAT. Without it, the Army cannot control the quality of training development. Though we will focus on the design phase, our research on job design issues should be applicable to other phases.

5.5 Research Designs.

a. Strong designs let you measure performance resulting from use of the products of training development. But we may be limited to rating the quality of the development tools and their products. For example, instructional developers would be trained to use an experimental PST. They would then apply the tool and rate its usefulness and ease of use.

b. They would also provide information on personal background, education, general job information, and job experience. A well designed questionnaire could yield considerable information about the effectiveness of an automated tool.

c. The research can be strengthened by measuring the user errors of commission and omission. Errors of commission can be recorded by an on-line log. Errors of omission include incomplete task performance or long pauses indicating confusion. Quality and quantity of output with and without automated tools provide additional measures. We can still conduct the trade-off studies recommended by Uhlaner (1970). To do so we would use sequential research designs (Simon, 1973). These allow us to examine many independent variables in applied test beds.

e. Charles Simon (1973) offers a compelling alternative to the prohibitive, complete factorial design. First identify all the variables you think apply. A questionnaire can help you do so. Then do pilot studies to screen unimportant variables. Fractional factorial designs will minimize data collection. Repeat the studies several times, excluding additional variables each time, but pooling data on remaining variables. Blocking techniques make the observations comparable across stages of the experiment.

5.6 Summary of Section 5.0.

(a) Design of a test bed begins with a top-level issue. How do you design and integrate performance support tools that support Army training development needs. Then interacting variables must be accommodated. We identified organizational and personnel factors. The effectiveness of PST design is constrained by the characteristics of the PST user and his or her organizational climate.

(b) We must then consider what MOEs to use in evaluating PST effectiveness. We identified five classes of measurement. These are cost, impact on the user performance, consistency of the quality of training products, usefulness in meeting TRADOC requirements, and last, but certainly not least, customer satisfaction.

(c) We described several research designs for exercising the above sources of independent and dependent variance. Highly controlled designs are preferable. These assess the performance of the PST user or effects of the training products on the trainee. However, the most feasible designs use ratings of training development tools, observations of PST users, or ratings of training development products.

(d) In any case, sequential testing procedures provide a model for dealing with the large number of variables we expect will impact on PST.

6.0 CONCLUSIONS

1. The literature on how to design and combine effective performance support tools is fragmentary, incomplete, or inconclusive.

2. However, the combination of a procedural guide (i.e. job aid), tutorial, and test production tool could help implement TRADOC policy on criterion-referenced performance assessment.

3. Available PST tools can be combined into an experimental system for pilot and field research.

4. A minimally effective PST combines a tutorial, job aiding, and test production capability.

5. Organizational as well as job design issues need to be considered in a comprehensive evaluation of job aiding approaches.

6. The results of this research provide a baseline and rationale for producing and evaluating an experimental PST system.

7.0 REFERENCES

Cormier, S.M., Dressel, D.J., and Mirabella, A. (1991). The Development of Content-Valid Performance Measures for the 76C Course and Field Assessment (ARI Technical Report 919). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences. (AD A232 741)

Curley, J.C. (1990). Job Aids: Literature Review. (Unpublished ARI Report). Alexandria, VA.: U.S. Army Research Institute for the Behavioral and Social Sciences.

Department of the Army. (1988, 26 February). Training: Systems Approach to Training. (TRADOC Reg 350-7), Ft. Monroe, VA.: Training and Doctrine Command.

Department of the Army. (1989, 28 April). Army Training Glossary. (TRADOC Pamphlet 25-33), Ft. Monroe, VA.: Training and Doctrine Command

Department of the Army. (1990, 26 March). The TRADOC Training Effectiveness Analysis (TEA) System. (TRADOC Reg 350-32), Ft. Monroe, VA.: Training and Doctrine Command.

Department of the Army. (1990, 17 September). Systems Approach to Training - Analysis. (TRADOC Pam 351-13), Ft. Monroe, VA.: Training and Doctrine Command

Department of the Army. (1990, October). Training Development Policy. (TRADOC Memorandum ATTG-CD-350), Ft. Monroe, VA.: Training and Doctrine Command.

Dressel, J.D. and Mirabella, A. (1991). Criterion-Referenced Testing: A User's Resource (ARI Research Product 92-65). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences. (AD A254 290)

Goicoechea, A., Stakhiv, E.Z., and Fu Li. (1991). Experimental Evaluation of Multiple Criteria Decision Support Systems for Applications to Water Resources Planning. Water Resources Bulletin. 15 January Issue.

Guion, R.M. (1979a). Principles of Work Sample Testing: I. A Non-Empirical Taxonomy of Test Uses. (ARI Technical Report TR-79-A8), Alexandria, VA.: U.S. Army Research Institute for the Behavioral and Social Sciences. (AD A075 381)

Guion, R.M. (1979b). Principles of Work Sample Testing: II. Evaluation of Personnel Testing Programs. (ARI Technical Report TR-79-A9), Alexandria, VA.: U.S. Army Research Institute for the Behavioral and Social Sciences. (AD A072 447)

Guion, R.M. (1979c). Principles of Work Sample Testing: III Construction and Evaluation of Work Sample Tests. (ARI Technical Report TR-79-A10), Alexandria, VA.: U.S. Army Research Institute for the Behavioral and Social Sciences. (AD A072 448)

Guion, R.M. (1979d). Principles of Work Sample Testing: IV.Generalizability. (ARI Technical Report TR-79-A11), Alexandria, VA.: U.S. Army Research Institute for the Behavioral and Social Sciences. (AD A071 083)

Guthrie, J.T., Bennett, S., Weber, S. (1990a). Processing Procedural Documents: A Transformational Model. (Technical Report 90-1), College Park, Md.: University of Maryland.

Guthrie, J.T., Bennett, S., Weber, S., and Wang, X. (1990b). An Evaluation of Components of an Automated Job Aid. (Technical Report 90-2), College Park, Md.: University of Maryland.

Majchrzak, A. and Klein, K. J. (1987, Fall). Things Are Always More Complicated Than You Think: An Open Systems Approach To the Organizational Effects of Computer-Automated Technology. Journal of Business and Psychology, 2, (1).

Majchrzak, A. (1988). Towards a Framework for Identifying Organizationally-Compatible Advanced Manufacturing Technology (AMT).

Mirabella, A. (1990). Test Development and Analysis Program (TDAP) and Army AIT Performance Testing. Proceedings of the 29th Annual U.S. Army Operations Research Symposium (AORS XXIX), Pp. 161-171, Volume 2, 9-11 October, 1990, Fort Lee, Virginia.

Mirabella, A. (1991). Performance Support Technology for Distributed Training Development. Proceedings of the 5th Annual Meeting of the DoD Training Technology Technical Group, 10-11 September, 1991, San Diego, California.

Mirabella, A., Macpherson, D.H., and Patterson, C.A. (1989). Application of Training Research Literature to Maintenance Performance Training (ARI Research Report 1527). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences. (AD A217 594)

Puterbaugh, G. (1990, Winter). Performance Support Tools: Theory and an Example. The AT&T Training Test Consultant. Journal of Interactive Instruction, 23-25.

Simon, Charles W. (1973). Economical Multifactor Designs for Human Factors Engineering Experiments. (Technical Report No. P73-326A), Culver City, CA.: Hughes Aircraft Company.

Science Applications International Corporation. (1990, 15 June). Formative Evaluation Report: Automated Systems Approach To Training. (Final Report, Contract DABT60-C-88-2741, CDRL A021). McLean, VA.

Stone, D.E. and Hutson, B.A. (1984, January). Computer-Based Job Aiding, Problem Solving at Work. (Technical Report No. 11). Ithaca, N.Y.: Department of Education, Cornell University.

Swezey, R.W., Llaneros, R.E., Perez, R.S., and Gittleman, S. (1991). Relationships Among Maintenance Skill Level and Use of Levels of Information Complexity in Verbal and Graphic Components of A Computerized Job Performance Aid (ARI Technical Report, In Preparation). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

Swezey, R.W. and Pearlstein, R.B. (1975). Guidebook for Developing Criterion-Referenced Tests (ARI Special Publication, P75-1). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences. (AD A014 987)

U.S. General Accounting Office. (1990, October). Report to the Secretary of Defense. Military Training. Its Effectiveness for Technical Specialties is Unknown. (GAO/PEMD-91-4), Washington, D.C.

U.S. General Accounting Office. (1989). Evaluation of Performance in General Support Maintenance. (NSIAD-89-183), Washington, D.C.

Uhlaner, J.E. (1970). Human Performance, Jobs, and System Psychology--The Systems Measurement Bed (BESRL Technical Report S-2). Arlington, VA: U.S. Army Behavior and Systems Research Laboratory. (AD A716 346)

Warrick-Bell, T'Wana. (1991). Application of Intelligent Systems to Maintenance Training. (Final Report, Contract No. DAAL03-86-D-0001) Research Triangle Park, N.C.:Battelle.

Wulfeck, W.H., Dickieson, J.L., Vogt, J.L., and Cody, H. (1991) The Automation of Curriculum Development Using the Authoring Instructional Materials (AIM) Program. American Society of Naval Engineers Joint Symposium on Fleet Maintenance in the 21st Century, 22-23 October, 1991.

8.0 APPENDIX A
Analysis of TRADOC and GAO Documents

Source: TRADOC Reg 350-7

Requirement/Issue/Method for PST Support	Performance Support Technology Concept
1. Compliance with TRADOC policy	1.1 Job aid that lays out SOP. 1.2 Tutorial on policy and SOP 1.3 Automated procedures which generate data, reports, training products per TRADOC criteria
2. Need to develop & follow SOP that complies with TRADOC policy	2.1 PST = mechanism to develop, document, and implement SOP
3. Page 2-2 e(1)(e) cites criterion-referenced testing as evaluation doctrine. Implied issue: what is CRT and how SME know he's doing it	3.1 Tutorial
4. Par e(2) requires evaluation of unit training programs. Implied issue: differences in approach and methods between program evaluation, tests of understanding, practice exercises and certification at the end of training segments	4.1 Tutorial 4.2 Menu that branches user to different job aids for the different types of measurement.
5. Page 2-2. f. Standardization. The regulation calls for application of uniform practices and procedures".	5.1 Any PST feature will help standardize testing practice in the Army.
6. Page 2-3 g. on audit trails	6.1 Book marking, electronic checklisting, disk notes
7. Page 2-4, Para 2-4a. Quality Control.	
What does the paragraph mean and how would a new officer assigned to TRADOC implement it	7.1 Tutorials and job aiding

How do you certify the quality of performance tests	7.2 Test quality measures: validity, reliability, Raush techniques
8.0 Page 2-4, Para 2-4b. Analysis	8.1 Task analysis tools, e.g. AKAT
9.0 Page 2-4, Para 2-4c. Design	9.1 Test plan and item construction (TDAP, Examiner, MircroCAT)
	9.2 Scenario generation (Primitive form in Examiner; more advanced form in AFHRL software
10 Page 2-4, Para. 2-4d. & e. Development & Implementation	10.1 Test compilation, delivery, and data processing TDAP, MicroCAT, and Examiner have similar capabilities here
11. Page 2-4, Par 2-5a TRADOC responsibilities, especially: (6) Training for TRADOC staff (7) Training for trainers (8) Promotion of automation	11.1 Tutorials geared to populations cited in (6) and (7)
12. Page 2-5, Par 2-5c. ATSC responsibilities.	
(2) ATSC is responsibile for compiling parts of the new Individual Development Tests (replacements for SQT) and	12.1 Test compiling routines which generate hard copy as well as computerized forms.
(4) distributing test results and	12.2 Data processing routines which give individual as well as summarized group results.
(5) training its staff.	12.3 Tutorials and job aiding.

13. Page 2-6, Par. 2-5d.
Proponents

(1) Apply SAT	13.1 Tutorials, job aids, auto routines
(2) Evaluate materials	13.2 Test quality routines
(3) Provide plans as proponents	13.3 Test plan routine in TDAP
(4) Provide materials as proponent	13.4 Test item data banks or hard copy
(8) and (9) Provide to or receive from other organizations test development products to incorportate into one package, e.g., get test items from a 76C unit and add to existing item bank	13.5 Routines to read in data from other workstations without rekeying.
(11) Maintain audit trail	13.6 Bookmarking, electronic checklist, disk notes
(13) Provide specialized training to combat and doctrine staff as well as to trainers and training developers.	13.7 Job aiding and tutorials

(14) Provide formal training for training developers before they enter skill code 7Q, SQI 6, or, SQI 2.	13.8 Tutorial, and job aiding.
(15) Formally train instructors before they enter skill codes 5K, SQI 8, or SQI.	13.9 Tutorial and job aiding 14.1
14. Ch 3 deals with diagnostic assessment of combat skills and unit readiness as a basis for revising existing training or developing new training. It also deals with checking the quality of training and related testing.	14.1 The initial focus of Task 3310 is on the design phase of SAT and on individual training. But without going beyond the scope of our initial mission, we could add PST features that would tangentially support the program evaluation described in Ch 3. Statistical analyses to measure test quality Group summaries of test data Included in TDAP Tutorials explaning the roles of measurement in evaluation vs. courseware.

Review of 350-7 Issues (Continued)

15. Ch4. Analysis:
P.4-3, Table 4-1.

1. Perform needs analysis
-Triggering circumstances

15.1 Data analysis routines in TDAP, Examiner or MicroCAT could access and analyze performance data in or ASAT.

15.2 Some of that data could be initially collected by the above tools, as was done for the BEST project.

2. Perform doctrine and mission analysis.

15.3 Incorporate special word processing macros which would serve double purpose of helping construct test items and write analyses, HEL program to write training device specs.

7. Perform task Analysis

15.4 AKAT or similar tool to do analysis.

15.5 Output goes to data base which provides input for Test Development Plan TDAP (or similar routine Examiner.

Review of 350-7 (Continued)

16. Ch 5 Design. Initial focus of Task 3310 is on this chapter, especially on Items 2 & 3, Table 5-1, Page 5-2.

Item 2. Perform learning analysis: enabling learning objectives; Skills and knowledges for mastery.

16.1 Special purpose word processing e.g. item generation techniques; HEL spec. generator.

16.2 Graphical techniques AKAT, Test Development Plan (TDAP), Test dimensioning technique of MicroCAT.

3. Develop test items	16.3 Task input from item 2 above, i.e. Item 2 work generates data basess for storing items.
	16.4 Special purpose word processing, e.g. item & generators; HEL spec generator.
17. Para 5-3.C., Page 5-2 Develop appropriate tests	
(1) Design CRTS	17.1 Tutorial is critical. Basic problems of understanding need to be sorted out clearly. e.g. Reg 350-7, glossary 3 fails to define CRT correctly. (GAO report and misunderstanfings.
18. Para 5-3e, Page 5-3 Develop training sequence and design.	18.1 AKAT or other flow-charting procedure (e.g. even MacProject). In fact MacProject may be well suited to this with tests as milestones.
19. Para 5-3f, Page 5-3 Maintain audit trail.	19.1 Bookmarking, electronic check listing disk notes and all the electronic files generated by PST.
20. Chapter 6. Development Page 6-1, Para 6.3.	
Para 6.3 a&b. Review/revise existing materials (Also Table 6-1, Items 1 & 2).	20.1 Editing capability to access and revise stored test items or
	20.2 Test compiling routines in TDAP & Examiner.
Para 6.3 C. Validate training materials (Also Table 6-1, Item 3)	20.3 Validity/ liability, item analysis procedures.
	20.4 Tutorial on how to develop measurement instruments to validate training.
Para 6.3 page 6-2 e. Develop staff plan	20.5 Tutorials on developing and administery perf. eval instruments.
Para 6.3f, page 6-2. Maintain	20.5 Disk notes, book marking,

audit trail	electronic check listing.
21. Chapter 7 Implementation, Table 7-1, Page 7-2	
1. Train staff, faculty, and cadre.	21.1 Tutorials geared to needs of each.
2. Conduct training	21.2 Test delivery on computer and hard copy. Need to consider different delivery conditions, e.g. during MOS training vs. IDT.

Source: TRADOC Pam 351-13

1. Page 2, Para. 4-1.a. discusses differences between external and internal evaluation

1.1 Tutorial, perhaps job aids to help SMEs distinguish differences between performance assessment strategies and tactics under external and internal evaluation.

2. Chapter 2, Page 4, Para. Para. 2-2.c. As part of mission analysis, SME is told to "determine the deficiencies and develop a plan specifically designed to meet the requirements."

2.1 Tutorial on how to use several sources of data, including perfomance measures to assess deficiencies. Alert SME to non-training solutions, e.g., job aiding, job re-design.

3. Chapter 3, Page 13, Para. 3-3.g. Standard setting.

3.1 Tutorial on issues in standard setting. e.g. standards vs. retention; distortions from non critical steps in process standards.

Source: TRADOC Pam 25-33, "Army Training Glossary.

1. Page 8. <u>Absolute Standard</u>

1.1 First sentence makes sense, the rest aren't clear. Could explain and illustrate this concept in the help routines or tutorials

2. Page 16. <u>Criterion</u>

2.1 Definition is circular; implies that criterion = scale; doesn't distinguish between criterion in CRT and criterion as a cut-off score.

3. Page 16. <u>Criterion Behavior</u>

3.1 Reasonable definition

4. Page 16 Criterion-Referenced Grading

4.1 The 1st sentence is inaccurate if "grading" means "assigning a grade". The test is the basis for grading. The standard is the basis for interpreting the grade. The second sentence states an incorrect generalization. That is, the standard (as defined here) ought to be based on job requirements but seldom is, especially at the schools which use arbitrary cut-off scores.

4. Page 16 Criterion-Referenced Test

4.1 Confounds and confuses the tool to measure criterion behavior and the cut-off score to judge success or failure.

5. Page 17 Diagnostic Test

5.1 Not a bad definition, but methodology for developing and applying diagnostic scales needs to be laid out and communicated either through tutorial or job aid.

6. Page 19 Entry Skills Test & Entry Test.

6.1 Definitions, issues, and methods need to be made clear in tutorial or job aid

7. Page 19 Evaluation:
Criteria
Phase
Plan
Program

7.1 Need to elaborate on definitions, issues, and methods for different conditions, e.g. school house MOS course vs. operational unit.

8. Page 19 Face Validity

8.1 Synonym for Content Validity. Concept is developed in the Test Battery report. Need to operationalize it through tutorial or job aiding.

9. Page 19 Feedback

9.1 Need to determine what processing and reporting techniques are needed for various circumstances.

10. Page 20 Field Test

10.1 Performance assessment situation. Needs elaboration.

11. Page 20 Field Validation	11.1 Performance assessment situations. Needs Elaboration.
12. Page 21 Group Trial Group-paced training Go/no-go-pass or fail Hands-on Hard data Hard skill	12.1 Performance assessment situations. Need elaboration
13. Page 22 Individual Training Evaluation Program Individualized Instruction	13.1 Performance Assessment situations. Need elaboration
14. Page 22 Indicator behavior	14.1 Candidate topic for tutorial or job aiding
15. Page 23 Item analysis	15.1 Need to examine and explicate role in criterion-referenced testing.
16. Page 24 Job fidelity Job performance measure Job performance test	16. Candidate topics for tutorial or job aiding.
17. Page 28 Norm-referenced test	17.1 Confusing definition. Doesn't reflect the essential difference between CRT and NRT. The difference lies in how test content is created. Score processing and interpretation are related but separate activities. These need to be kept apart.
18. Page 29 & 30 Pre & Post Testing	18.1 Issues and methods need to be worked out.
19. Page 30 Process & Product standards	19.1 Incorporate into tutorial on standard setting.

20. Page 32 Sampling	20.1 Explain in tutorial or help routines when and how to use in building tests for different purposes.
21. Page 33 Speed vs Power test Standard of performance	21.1 Candidate issues for tutorial
22. Page 33 Soft data	22.1 Information may be inaccurate or misleading. Candidate for tutorial or help screen.
23. Page 39 Validity	23.1 Candidate for tutorial or help screen. Also candidate for job aiding routines, e.g. criterion-referenced validity computation.

Source: Memo for all TDAD Divisions, Review of Training Development Policy, 16 Nov 90, COL Seger

1. Page 3, Para. 3. treats schools as TOE units, with soldier evaluation as major task.	1.1 Helps justify our focus on PST for performance evaluation
2. Page 4 Para. 4.a.(1) highlights Pam 25-33 (Glossary) as tool for standardization. The glossary has many definitions which seem to be inaccurate.	2.1 Further justification for focus on PST for performance evaluation.
3. Page 4 Para. 4.a.(5) defines workload vs. unresourced requirements.	3.1 Possible basis for evaluating PST tools. i.e. do they increase the number of requirements which can be resourced?
4. Page 5 Par. 4.b.(1) Formats	4.1 Evaluation PST can help reduce the cost and time of approval.
5. Page 6 Para. 4.c.(1) Automation schedule.	5.1 ASAT to be implemented in Late Fy 92 or early FY 93. Therefore Task 3310 is timely.

6.1 Training Support Package, Page 2. PST can support a number of items in the TSP	
Exercises	6.1 Item writing and scenario generation techniques for producing part-task and practical exercise sheets.
Checks on learning (Page 10)	6.2 Same as above but formatted for checks on learning, along with tutorial, help sheets, procedural guidance
End-of-block tests	6.3 Ditto
7. Page 12. Guidelines for evaluating students	7.1 Consider incorporating this template into PST perhaps as part of a scenario generating program or an HEL type program.
	7.2 Check HumRRO decision aid against this page.
8. Page 14 Practical Exercise Format.	8.1 Consider working this into the PST system via job aiding, tutorials and help screens. Check the HumRRO/AISTA decision aid against this page.
9. Enclosure 2. Writing Acceptable Tasks	9.1 Check against HumRRO/AISTA aid for possible revision of the aid.

Source: MILITARY TRAINING. Its Effectiveness for Technical Specialties is Unknown GAO Report PEMD 91-4, October 1990

1. Letter to the SECDEF from Assistant Comptroller General. GAO alerted the SECDEF, Congress, and OMB to problems with performance evaluation in the services. The problem therefore has high visibility, notwithstanding GAO misunderstandings about criterion- reference testing.

1.1 Supports the focus of Task 3310 on performance evaluation.

2. Page 2. Exec Sum Background. "... three points in a recruit's service career where data critical to evaluating the success of training must be collected: at entrance to military life, during and upon completion of formal training, and after assignment to a military specialty in the field. An adequate system of assessing training effectiveness must include reliable and valid information at each of these points...

2.1 Further support. Task 3310 focuses on the middle point, but its results should generalize to the third point.

3. Page 2 Results in Brief, second paragraph beginning "Each service has ...". The report compliments the SQT and criticizes Navy - Air Force evaluation.

3.1 Reference to SQT is relevant because Army is about to eliminate SQTs. PST for evaluation can help insure that the alternative (IDT) doesn't come under attack in the future.

3.2 Reference to Navy is relevant because it will become lead lab for individual instruction. ARI work on evaluation could provide useful supporting work. In any case we should review the GAO complaints about Navy and Air Force as well as complaints about Army to determine how PST can deal with them.

4. Page 5. Recommendations.

"GAO... recommends that the Secretary of the Army direct the Training and Doctrine Command to review for accuracy, appropriateness, and reliability the classroom grading procedures identified within the report as deficient.

Agency Comments "In a written response to a draft of this report DOD concurred with all of its recommendations and identified specific actions to be taken toward implementing them."

4.1 The report and DOD response indicate that research on PST for performance evaluation can be useful. We need to define the terms "accuracy" "reliability" and "appropriateness" as used in the report. Then we should think about how PST can address these.

4.2 We need to look at DOD comments (Page 78 -102) in more detail. Focus on sections dealing with Army classroom and field evaluation.

9.0 APPENDIX B
Review of Test Development and Analysis Program (TDAP)

A BRIEF SUMMARY OF
THE TEST DEVELOPMENT AND ANALYSIS PROGRAM (TDAP)

by
Tracy A. Dunlap
Planning Systems Incorporated

for
US Army Research Institute
5001 Eisenhower Avenue
Alexandria, Va 22333-5600

1.0 INTRODUCTION

The Test Development and Analysis Program (TDAP) is a DBASE III Plus based software package, that allows its users to develop, administer and analyze tests via database management system technology. The system utilizes a four-part design scenario in order to achieve the objective. Figure 1-1 illustrates the Main menu setup screen.

2.0 SECURITY MANAGEMENT

The system's security package incorporates a log-in/password method, in addition to a specific user capability code which is designed to limit user access. These accounts/codes are setup/modified by the System Manager. Specific user passwords can also be edited by the user. This is accomplished by choosing select n [S] from the main menu screen, refer to Figure 1-1.

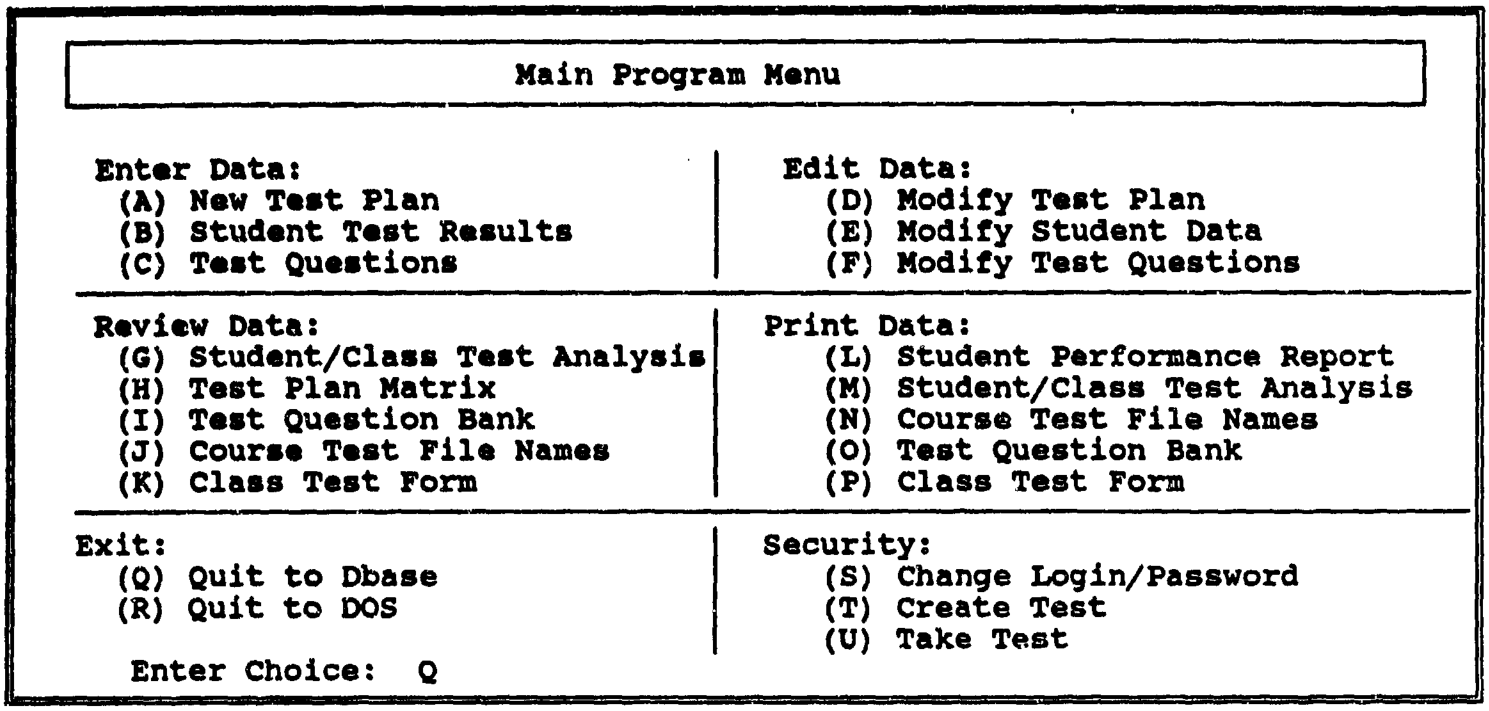
The final component of the security package is the audit trail. The audit trail is a record of which user(s) attempted to sign onto the system, the number of attempts at entering an incorrect password, the menu options selected and selected information about what was done within certain selection modules. The audit trail can be reviewed, printed, edited or erased by the System Manager. Figure 2-1 illustrates the primary option screen after selection [S] has been chosen.

3.0 TEST DEVELOPMENT PLAN AND ITEM WRITING

By choosing options [A] and [C] from the main menu, the user can create a two-dimensional test matrix (template), which can then include questions (items) that are specifically linked to each cell within the template.

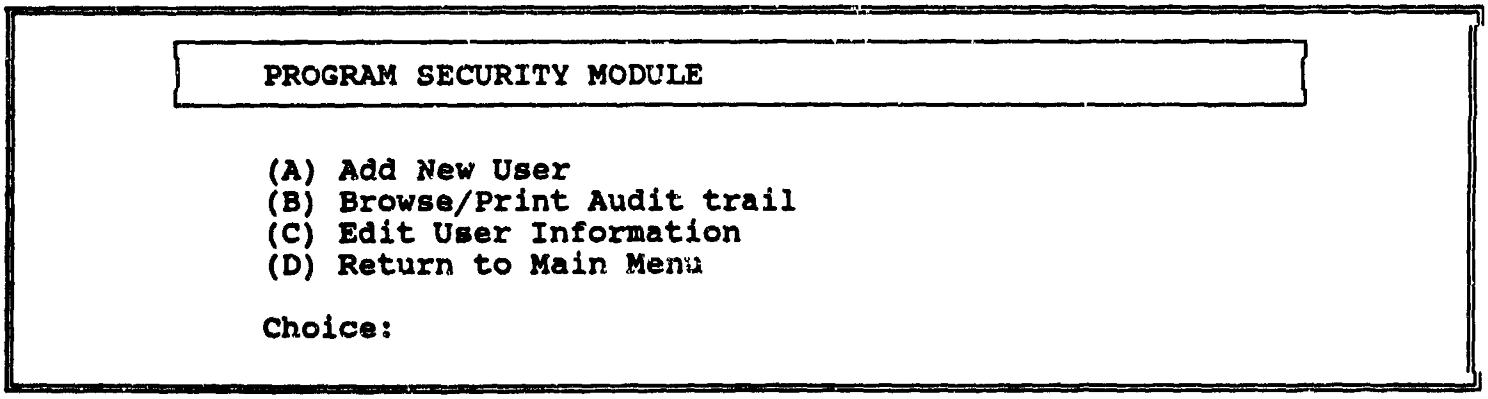
3.1 Test Development Plan

The test development plan is a two-dimensional template that can only be differentiated by its course number and annex letter. The user types in titles or headings for the matrice's rows and columns, in order to categorize the item elements. A maximum template of 6 rows by 8 columns can be created. The course number/annex letter matrice (name) is listed in file Plaslist.DBF; the Course Test File Name file. Figure 3-1 shows the screen used to generate the test template, Figure 3-2 illustrates the resulting test plan matrix, while Figure 3-3 lists sample entries from the Plaslist.DBF file.



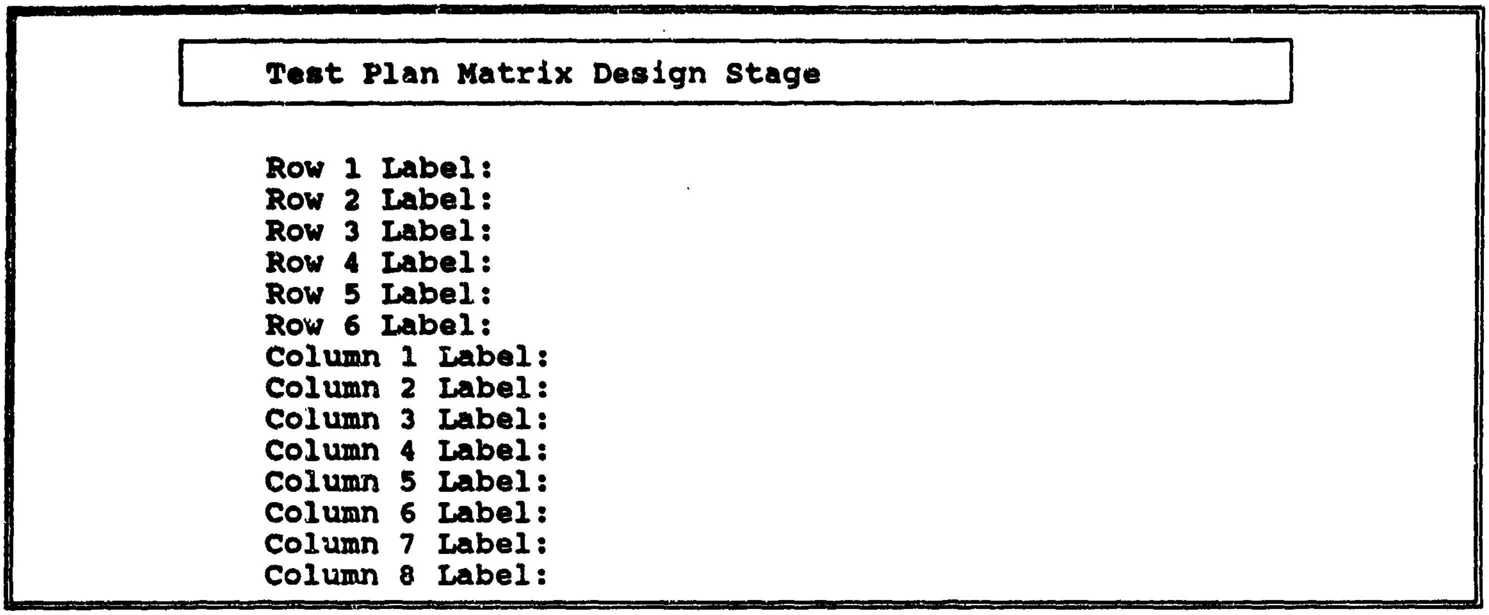
```
                         Main Program Menu

 Enter Data:                         | Edit Data:
  (A) New Test Plan                  |    (D) Modify Test Plan
  (B) Student Test Results           |    (E) Modify Student Data
  (C) Test Questions                 |    (F) Modify Test Questions
 ---------------------------------------------------------------------
 Review Data:                        | Print Data:
  (G) Student/Class Test Analysis    |    (L) Student Performance Report
  (H) Test Plan Matrix               |    (M) Student/Class Test Analysis
  (I) Test Question Bank             |    (N) Course Test File Names
  (J) Course Test File Names         |    (O) Test Question Bank
  (K) Class Test Form                |    (P) Class Test Form
 ---------------------------------------------------------------------
Exit:                                | Security:
  (Q) Quit to Dbase                  |    (S) Change Login/Password
  (R) Quit to DOS                    |    (T) Create Test
                                     |    (U) Take Test
   Enter Choice:  Q
```

FIGURE 1-1: TDAP Main Menu Setup Screen



```
     PROGRAM SECURITY MODULE

     (A) Add New User
     (B) Browse/Print Audit trail
     (C) Edit User Information
     (D) Return to Main Menu

     Choice:
```

FIGURE 2-1: Primary Option Screen For Selection "S"



Test Plan Matrix Design Stage

Row 1 Label:
Row 2 Label:
Row 3 Label:
Row 4 Label:
Row 5 Label:
Row 6 Label:
Column 1 Label:
Column 2 Label:
Column 3 Label:
Column 4 Label:
Column 5 Label:
Column 6 Label:
Column 7 Label:
Column 8 Label:

FIGURE 3-1: Test Template Generation Screen

Test Plan Matrix As Designed

	Req. For Issue NSN	Turn In	Rep. Exch.	PLL Maint.	Supply Status	RFI Non NSN	Follow Up	General
Form Functions								
Definitions								
Form Fill Out								
Form Related								
Postings								
System Procedures								

FIGURE 3-2: Resulting Test Plan Matrix

Course Number	Annex	Class Number	Test Form	Test Bank File	Class Results File	Class Test File
76C-AIT-PLL	B			TB5363		
76C-AIT-PLL	B	9	A	TB5363	CL7014	TS6208
76C-AIT-PLL	B		B	TB5363		TS6221
76C-AIT-PLL	B		C	TB5363		TS6234
76C-AIT-PLL	B		D	TB5363		TS6247
76C-AIT-PLL	B		E	TB5363		TS6260
76C-AIT-PLL	B		F	TB5363		TS6273
76C-AIT-PLL	B		G	TB5363		TS6286
76C-AIT-PLL	B	11	A		CL7661	TS7661
76C-AIT-PLL	B	14	A		CL7703	TS7703
76C-AIT-PLL	B	8	A		CL7001	TS7001
76C-AIT-PLL	B	10	A		CL7647	TS7647
76C-AIT-PLL	B	5	A		CL6962	TS6962
76C-AIT-PLL	B		H	TB5363		TS6299
76C-AIT-PLL	B		I	TB5363		TS6312
76C-AIT-PLL	B		J	TB5363		TS6325
76C-AIT-PLL	B	2	J		CL7049	TS7049

FIGURE 3-3: Sample Entries From Plaslist.DBF File

3.2 Item Writing

Multiple choice items, which are specifically associated to each cell within the test plan matrix, are used. The user specifies the location of the item by inputting the precise course number, annex letter and cell location (row and column number). A cell can contain any number of items, and it is from these cells that the user can choose the questions which are to be included into the test. The program automatically creates the item database file called TB****.DBF. The "****" is an encrypted string created from the course number and annex letter based on the test matrice used. Figure 3-4 illustrates the sample item entry screen. Refer to Figure 3-3 as to where the item database file name and location (course number/annex letter) are listed.

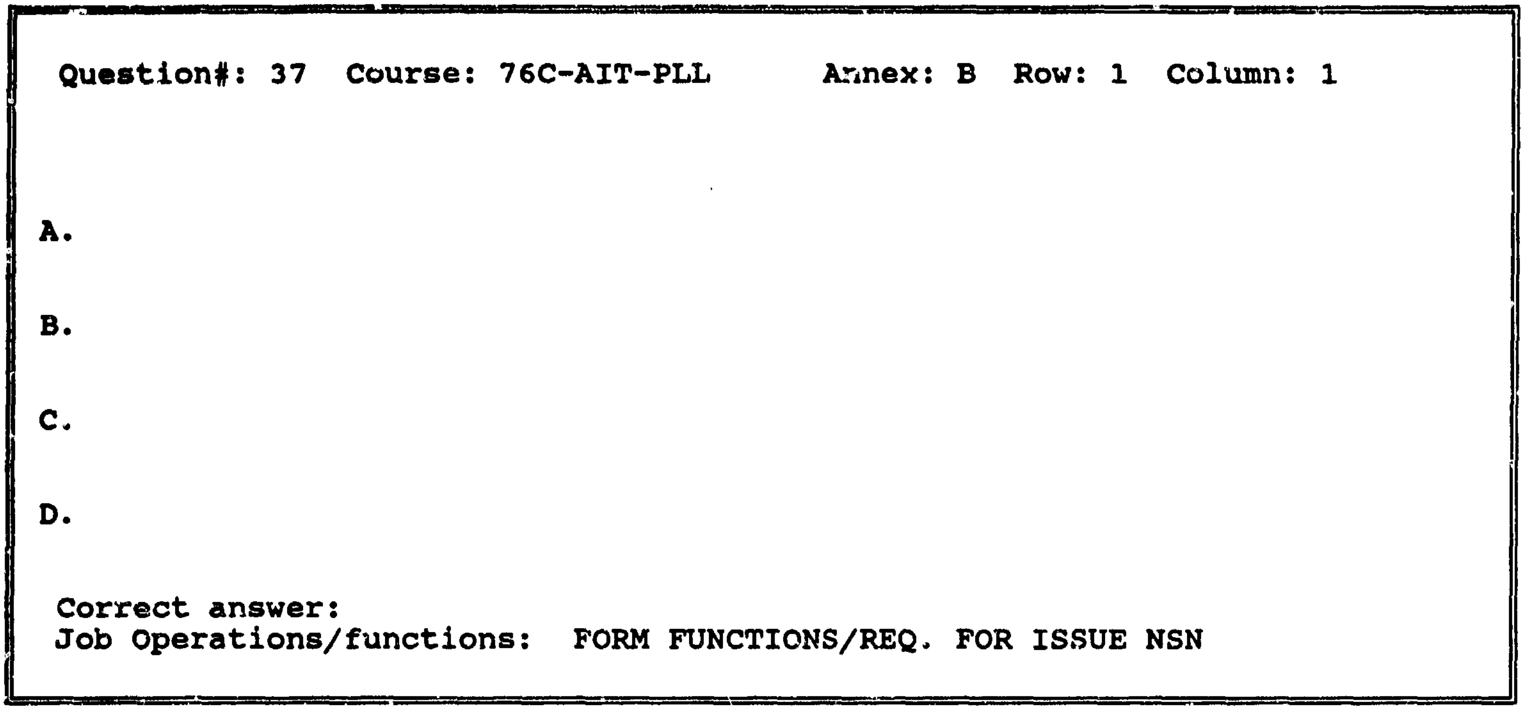
4.0 ITEM SAMPLING AND TEST COMPILATION

By selecting option [T] from the Main Selection Menu, exams can be constructed once a sufficiently large item data base has been created. Figure 4-1 shows the screen used to create a customized test via individual cell selection. This screen is similar to the test plan template. The upper left corner of each cell is the number of items available. The box on the lower left is used to type in the number of items to be sampled from the pool of items. The user specifies the total number of test items that are to be included, and then the system randomly selects questions contained within the rows or individual cells; for this case it's per individual cell. In short, the user only has to input the course number/annex letter and the total number of questions required per row or individual cell (test weights); the system does the rest. The program automatically creates the test file called TS****.DBF. The "****" is an encrypted string created from the course number, annex letter and test form letter. The test form letter is automatically determined by scanning the Plaslist.DBF file, determining the last test form letter used and taking the next letter as the letter designation for the new test form. Refer to Figure 3-3.

5.0 TEST ADMINISTRATION AND DISPLAY OF PERFORMANCE RESULTS

5.1 Test Administration

Menu option [U] enables any course test to be administered via computer, as long as a test form has been created. Figure 5-1 illustrates the main screen used to enter the required course information. After the neccessary information is inputted, the test is given. When all of the questions have been answered, Figure 5-2 is displayed, which shows the individual's performance on the test. (The test can also be administered via hardcopy. This is possible by selecting option [P].) A Class Results file



Question#: 37 Course: 76C-AIT-PLL Annex: B Row: 1 Column: 1

A.

B.

C.

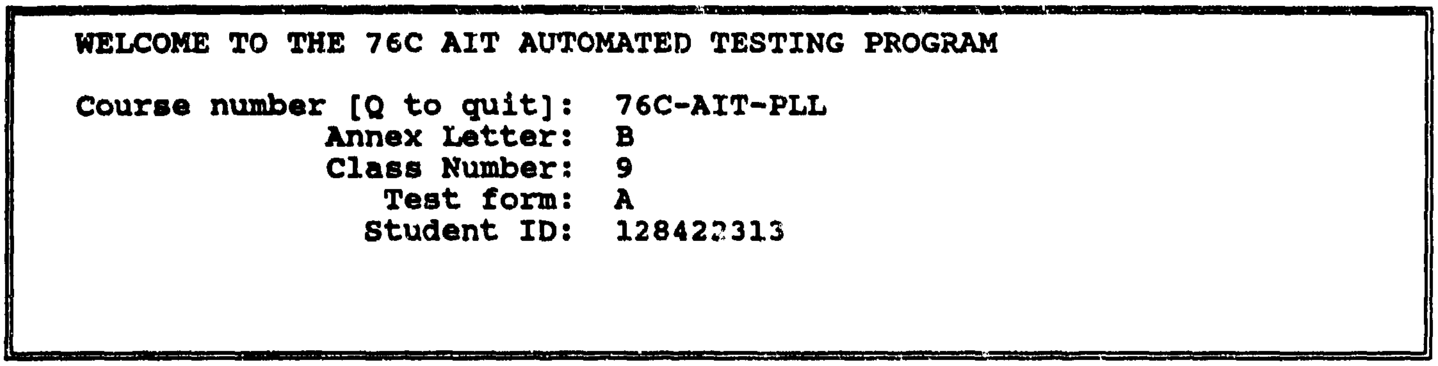
D.

Correct answer:
Job Operations/functions: FORM FUNCTIONS/REQ. FOR ISSUE NSN

FIGURE 3-4: Sample Item Editing Screen

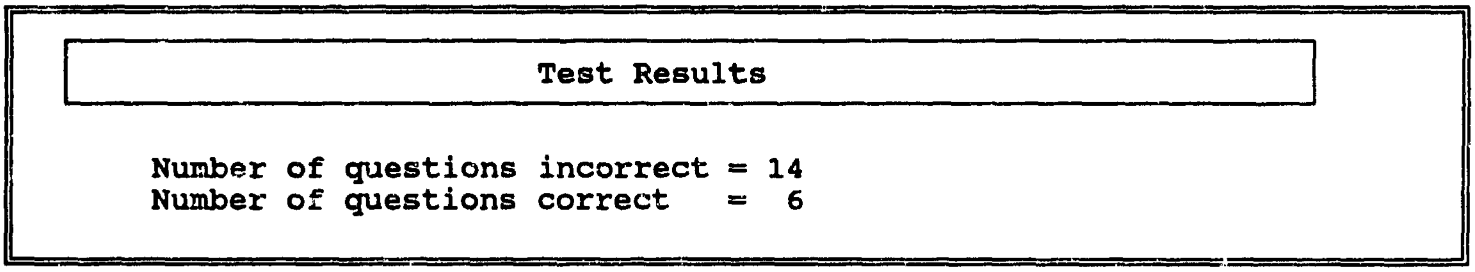
	Req. For Issue NSN	Turn In	Rep. Exch.	PLL Maint.	Supply Status	RFI Non NSN	Follow Up	General
Form Functions	1	1	1	1	1	1	1	0
	0	0	0	0	0	0	0	0
Definitions	0	1	0	1	1	1	0	1
	0	0	0	0	0	0	0	0
Form Fill Out	1	1	0	1	1	1	1	0
	0	0	0	0	0	0	0	0
Form Related	1	1	0	1	0	1	0	0
	0	0	0	0	0	0	0	0
Postings	1	1	1	1	0	1	1	0
	0	0	0	0	0	0	0	0
System Procedures	1	1	0	1	1	1	1	0
	0	0	0	0	0	0	0	0

FIGURE 4-1: Customized Test Creation Via Cell Selection



```
WELCOME TO THE 76C AIT AUTOMATED TESTING PROGRAM

Course number [Q to quit]:  76C-AIT-PLL
             Annex Letter:  B
             Class Number:  9
                Test form:  A
               Student ID:  128422313
```

FIGURE 5-1: Main Screen For Entering Course Information
In Order To Take Test



```
                           Test Results

    Number of questions incorrect = 14
    Number of questions correct   =  6
```

FIGURE 5-2: Individual Test Performance
Immediately After Test Completion

CL****.DBF is automatically created once a test has been computer administered and it is added to Plaslist.DBF. The "*****" is an encrypted string created from the specific course number, annex letter class number and test form letter. This file can also be updated by selecting option [E] from the main menu. Refer to Figure 3-3.

5.2 Display Of Performance Results

The same table used to lay out the test development plan is also used to display the number of errors made by a test taker(s). By selecting menu option [G] or [M], Figure 5-3 illustrates the distribution of errors across rows and columns for a fictitious student or class. Only the total number of incorrect items per cell, row or column can be viewed; statistics are not performed. In addition, these options can also list the actual items missed.

6.0 TDAP CAPABILITIES

The following lists the capabilities of TDAP.

6.1 Security Management

- The System Manager is able to add new users to TDAP through the use of login accounts, encoded passwords and capability codes.
- The System Manager can modify existing user capability codes.
- The user can modify his/her own password.
- An audit trail is automatically created which records all logins, password entries and menu selections.
- The audit trail can be viewed on screen, printed, edited or deleted.

6.2 Test Development Plan and Item Writing

- The Test Development Plan is based on a two-dimensional matrix, which is identified by the course number and annex letter.
- The items are written as multiple choice questions with up to four different alternatives permitted.
- The Test Development Plan and the Item Test Bank can be viewed, printed or modified.

	Req. For Issue NSN	Turn In	Rep. Exch.	PLL Maint.	Supply Status	RFI Non NSN	Follow Up	Generl	
Form Functions	1	0	0	1	0	1	0	0	3
Definitions	1	0	0	0	1	1	1	0	3
Form Fill Out	0	0	0	0	0	0	1	0	1
Form Related	0	1	0	0	0	0	0	1	2
Postings	0	1	1	0	0	0	0	1	3
System Procedures	0	0	0	0	0	0	1	0	1
	2	2	1	1	1	2	3	2	14

FIGURE 5-3: Distribution Of Errors For A Single Student

- The system automatically creates an Item Test Bank database which is based on the Test Development matrix. The database name will be TB****.DBF, where "****" is an encoded string based on the course number and annex letter.

- The system automatically creates a record and/or datafiles within Plaslist.DBF, when the matrix is created and it is updated as information becomes known. This record includes the course number, annex letter, class number, test form designator, test bank filename, class results filename and test filename.

6.3 Item Sampling and Test Compilation

- The total number of items selected per test can be chosen from the Test Development Matrix by cell or across rows. This is a simple method of "weighing" the test.

- After the item selection is made by either cell or by row, the items are automatically chosen randomly.

- The system automatically creates a test file, TS****.DBF, where "****" is an encoded string based on the course information.

- The system automatically generates a new test form designator (version letter), if the Test Development Matrix was reused. Also, the updated information is automatically added to the Plaslist.DBF file.

- The test versions can be modified, reviewed or printed.

6.4 Student Testing and Display of Performance Results

- Student testing can be accomplished via computer or hardcopy.

- The system automatically creates a class results file, CL****.DBF when the test is given on the computer, where "****" is an encoded string based on the course information. Also, the updated information is automatically added to the Plaslist.DBF file.

- The class results file can be modified, reviewed or printed.

- Performance results for either the student or the class can be obtained in a format like the Test Development Matrix.

- Performance results display only the total number of incorrect answers obtained by the student or class per cell, row and column.

6.5 General

- The user is able to backup the Item Test Bank, Test Form or Class Results files to floppy disk.

7.0 TDAP LIMITATIONS

The following lists the limitations of TDAP.

7.1 Security Management

- The System Manager is warned not to edit his security capability code or he will not be able to return to the security setup module.
- All passwords must be eight characters long and they must be completely typed, including spaces.

7.2 Test Development Plan and Item Writing

- The Test Development Plan is solely linked to the course number and annex letter. If the only difference between two matrices is the annex letter (the items are the same), an entirely new plan needs to be created along with the item entries.
- The Test Development Plan is limited in size, 6 rows by 8 columns.
- The capability doesn't exist to copy items from other matrices to the current matrix.
- If the Test Development Plan is modified by changing the ordering of the rows and/or columns, then the items need to be updated accordingly.
- Only multiple choice items with up to four alternatives can be created.
- There is a limited writing area for items.
- If accidentally enter an item and then abort the item writing session, sometimes that item will still be included within the Test Plan Matrix; however, the total item count will not reflect that item within the matrix.

- The system doesn't test for an incorrect alternative, i.e., choose an letter other than A through D.
- Items, which have multiple answers, are not allowed.
- System error exists; sometimes cannot enter more than one item within one cell.

7.3 Item Sampling and Test Compilation

- The user has no control over item selection. The system randomly selects items across rows or within cells.
- A maximum number of 99 items can be incorporated into a test.
- The user is unable to use items from other Test Plan Matrices to incorporate into a test. This is due to the inflexibility of the Test Plan / Item Data Bank link.
- The system automatically generates the Test Form Designator; however, the user hasn't any idea what that letter will be until he browses the Course Test File Names file (Plaslist.DBF).
- System error exists; the system has a tendency to duplicate certain fields and rename the test file within the Plaslist.DBF file, even though the test form wasn't modified.

7.4 Student Testing and Display of Test Results

- If the test is administered via computer, the capability doesn't exist for the user to skip, backtrack or change answers. Also, there are no set time testing limits.
- System error exists; confusing method of creating the Class Results file. Sometimes the system will enter the class updates automatically and sometimes it will not.
- If the user needs to edit the results, the method is awkward and clumsy.
- System error exists; the user cannot analyze the Student/Class test results. Each time the Test File is accessed, its filename is changed and a record is added to the Plaslist.DBF file. The record is added so that the Test File and the Item Data Bank are not within the same record; thereby, not allowing analysis of the results data.

- Statistical results are non-existent for the student, class or item. Only simple counts of incorrect items can be performed.

7.5 General

- Cannot import or export datafiles.
- Graphics are non-existent.
- An efficient method doesn't exist to copy, rename or delete datafiles, since the Test Development Plan, Item Test Bank, Class Results and Test files are course number / annex letter dependent.
- A search option doesn't exist for browsing through datafiles.
- The user must exit DBASE III, in order for any print options to work.
- The system is extremely limited, overall.
- The user screens are too cluttered and help screens are non existent.

10.0 APPENDIX C
Review of EXAMINER

A BRIEF SUMMARY OF THE TEST DEVELOPMENT AND ANALYSIS PROGRAM (TDAP)

by
Tracy A. Dunlap
Planning Systems Incorporated

for
US Army Research Institute
5001 Eisenhower Avenue
Alexandria, Va 22333-5600

1.0 INTRODUCTION

The EXAMINER software package is a test item banking and examination delivery system, that allows its users to create, administer and analyze tests using database technology in a DOS environment. The EXAMINER has four functions: the creation of databases of items and related information; the description and production of tests; the gathering and display of statistics; and the administration of tests via computer. The first function is accomplished using the Item Editor, the second through the Examination Editor, the third through the Statistics program, and the fourth, if delivered via computer, by means of the Administration system. Figures 1-1 and 1-2 illustrate the Directory of EXAMINER Databases and the Main EXAMINER Menu screens, respectively.

2.0 ITEM EDITOR

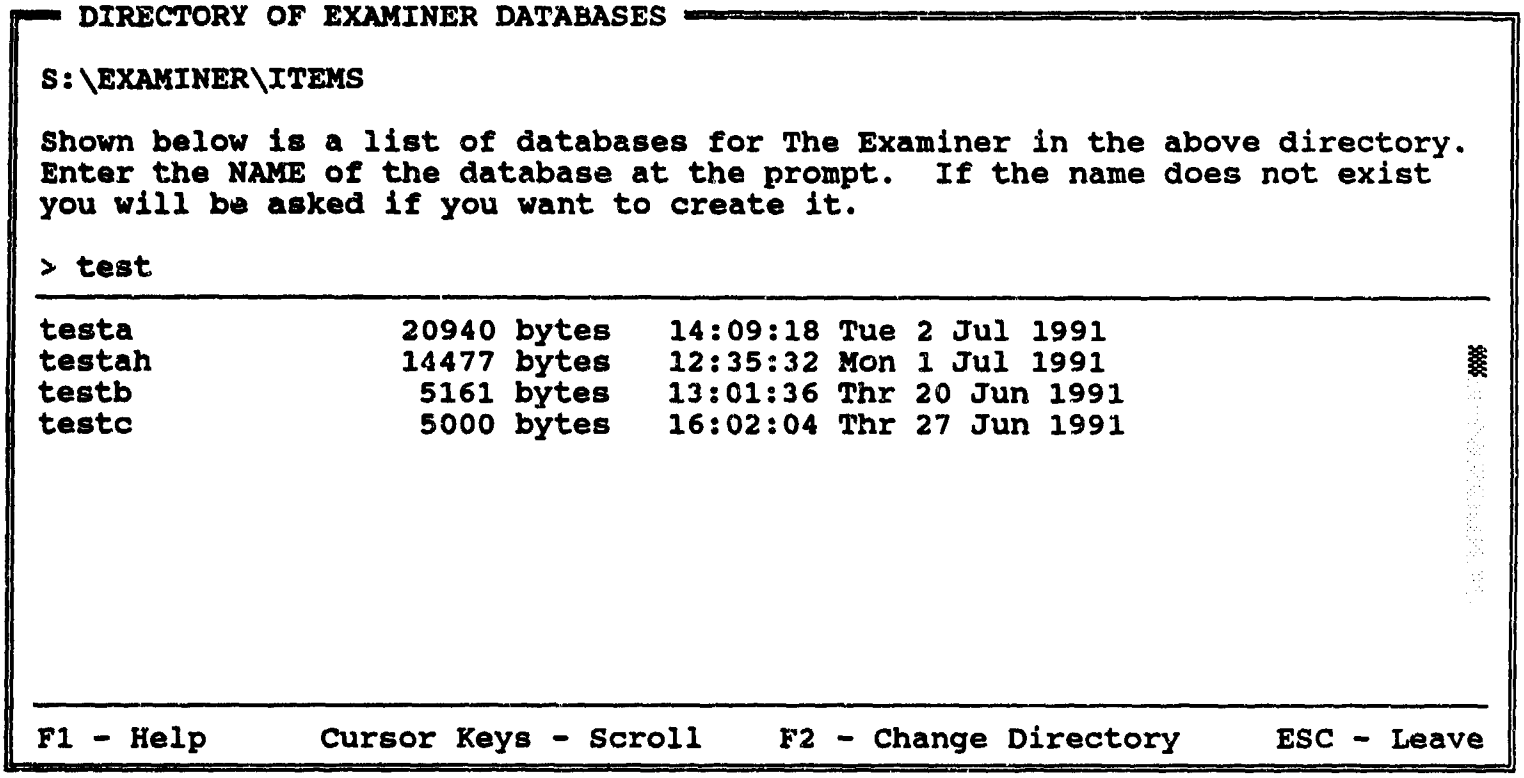
The Item Editor allows the user to define a hierarchial or tree structure of the database up to six levels deep. It incorporates a word processor to enable items to be stored in the database.

2.1 Database Creation

Database creation is a three step process; password initialization; database structure initialization; and the naming of the database levels. If it is the first time the database has been used, it is desirable to specify a set of passwords. There are four levels of access available, each with decreasing capability. They are Director, Edit, View and Delete. The Delete password is an additional level of protection, which is required by anyone who wants to delete an item. Figure 2-1 illustrates the screen on which passwords are initially entered.

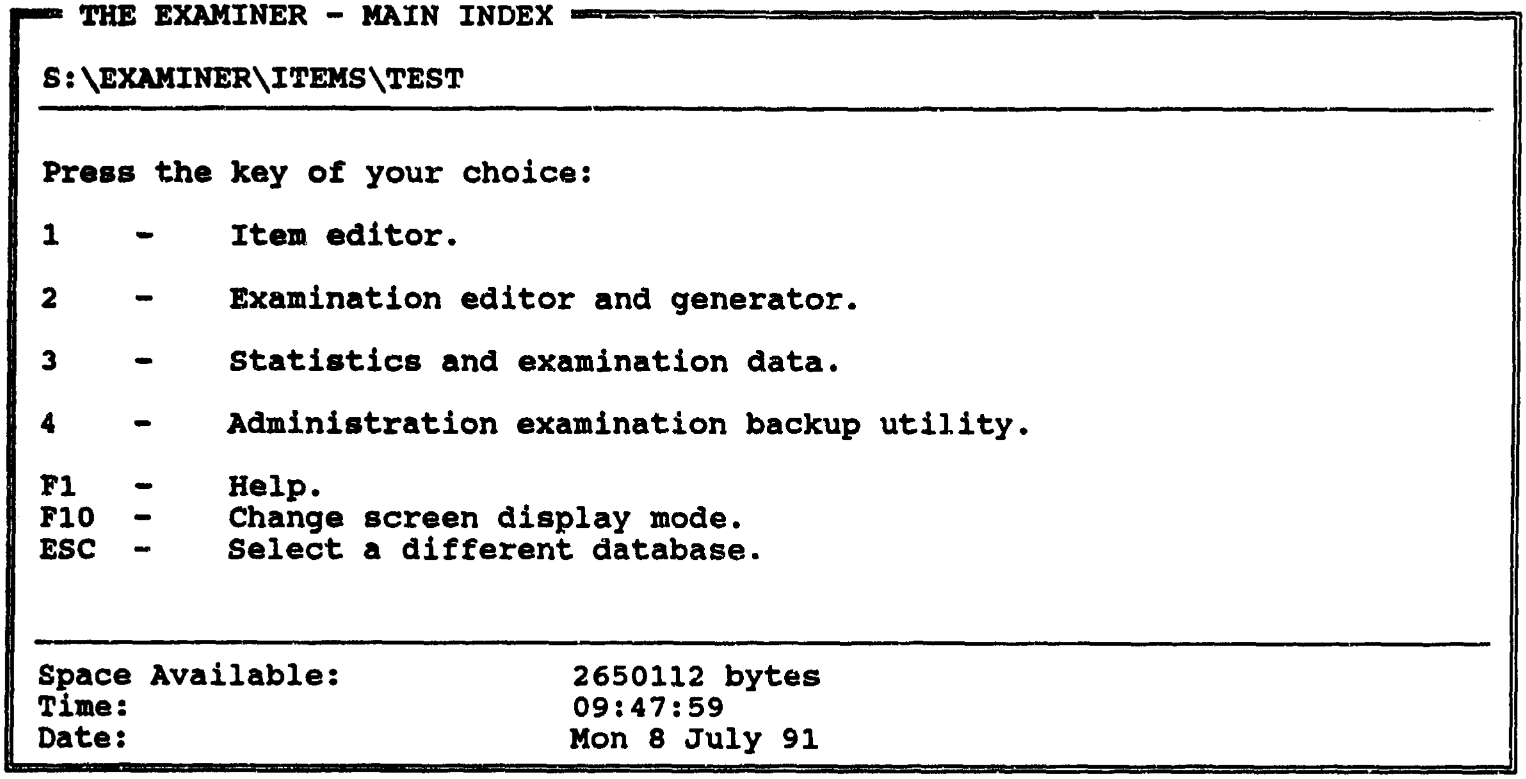
The database structure initialization is the next crucial step that affects the future use of the database. This screen is shown in Figure 2-2. The database structure is like a tree. At the top level are the broadest classifications and at the bottom are the actual items themselves. Between the two are the number of levels required to structure the content in the most efficient manner. Once the structure has been established and the database created, it is impossible to change.

After having specified the number of levels, Figure 2-3 appears on which each of the levels are to be named. The lowest level is always named ITEM. Even though it is impossible to change the number of levels within the database, the names can be edited at any time. This is the last step of the database-creation process.



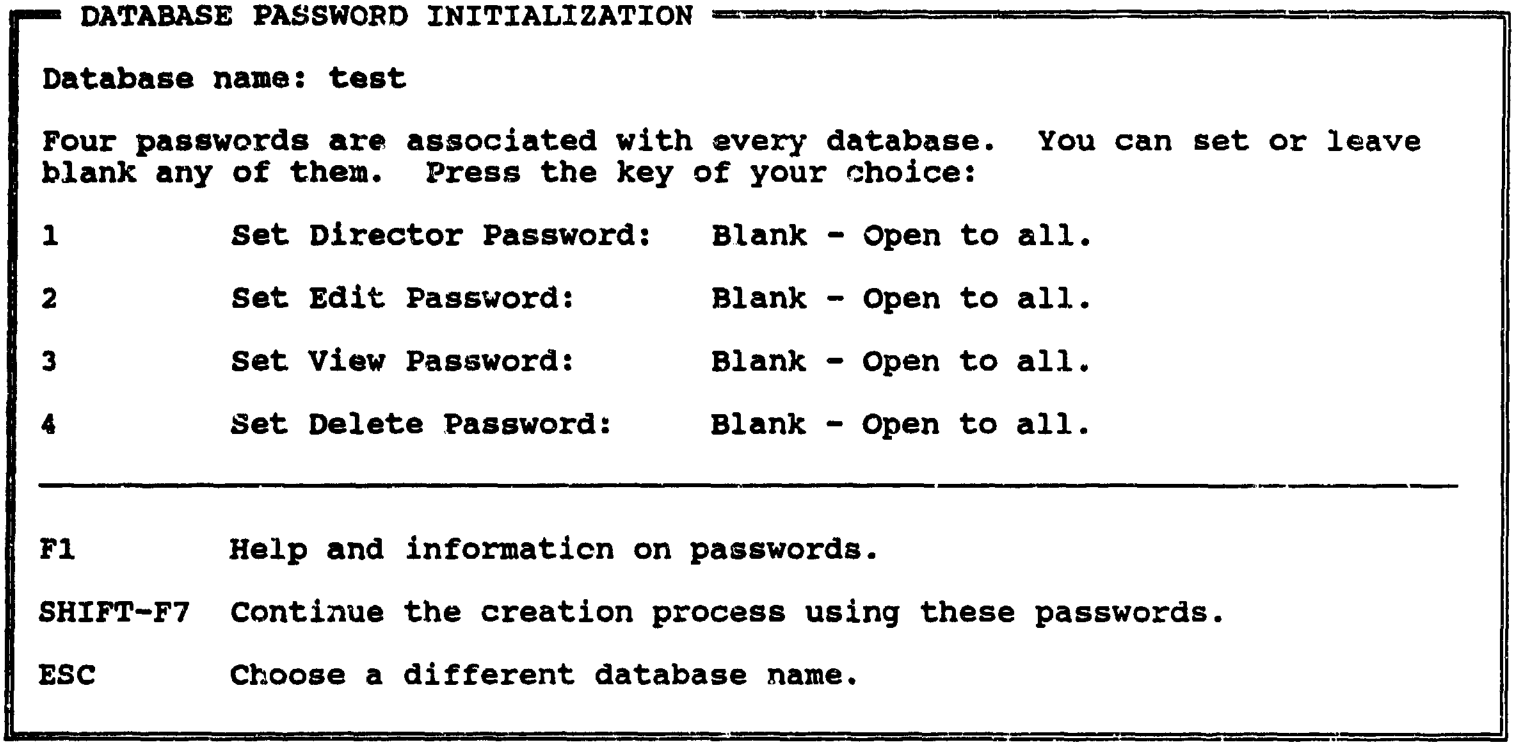
```
DIRECTORY OF EXAMINER DATABASES

S:\EXAMINER\ITEMS

Shown below is a list of databases for The Examiner in the above directory.
Enter the NAME of the database at the prompt.  If the name does not exist
you will be asked if you want to create it.

> test

testa                 20940 bytes   14:09:18 Tue 2 Jul 1991
testah                14477 bytes   12:35:32 Mon 1 Jul 1991
testb                  5161 bytes   13:01:36 Thr 20 Jun 1991
testc                  5000 bytes   16:02:04 Thr 27 Jun 1991
```

FIGURE 1-1: Directory Of EXAMINER Databases



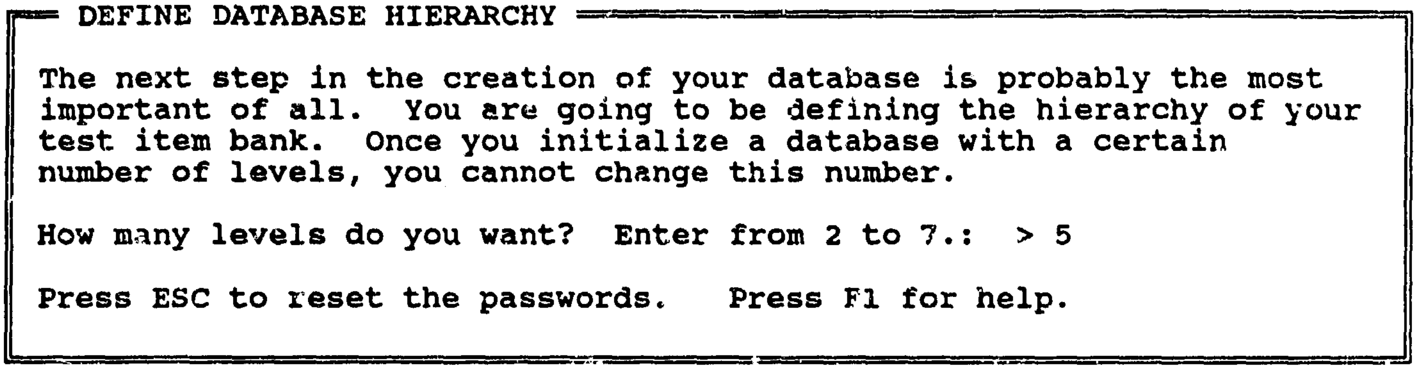
```
THE EXAMINER - MAIN INDEX
S:\EXAMINER\ITEMS\TEST

Press the key of your choice:

1      -    Item editor.

2      -    Examination editor and generator.

3      -    Statistics and examination data.

4      -    Administration examination backup utility.

F1     -    Help.
F10    -    Change screen display mode.
ESC    -    Select a different database.

Space Available:            2650112 bytes
Time:                       09:47:59
Date:                       Mon 8 July 91
```

FIGURE 1-2: EXAMINER Main Menu Screen



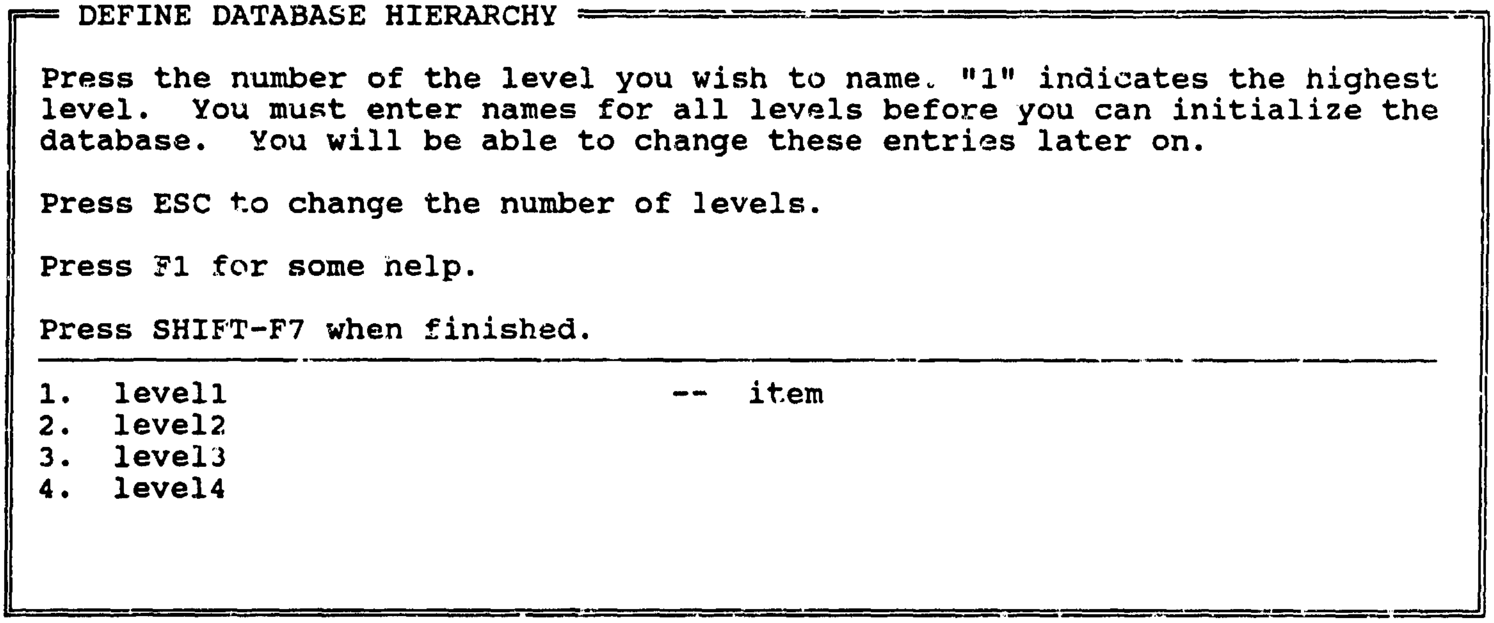
```
DATABASE PASSWORD INITIALIZATION

Database name: test

Four passwords are associated with every database.  You can set or leave
blank any of them.  Press the key of your choice:

1          Set Director Password:   Blank - Open to all.

2          Set Edit Password:       Blank - Open to all.

3          Set View Password:       Blank - Open to all.

4          Set Delete Password:     Blank - Open to all.

F1         Help and informaticn on passwords.

SHIFT-F7   Continue the creation process using these passwords.

ESC        Choose a different database name.
```

FIGURE 2-1: Database Password Screen



```
DEFINE DATABASE HIERARCHY

The next step in the creation of your database is probably the most
important of all.  You are going to be defining the hierarchy of your
test item bank.  Once you initialize a database with a certain
number of levels, you cannot change this number.

How many levels do you want?  Enter from 2 to 7.:  > 5

Press ESC to reset the passwords.   Press F1 for help.
```

FIGURE 2-2: Database Hierarchy Definition



```
DEFINE DATABASE HIERARCHY

Press the number of the level you wish to name. "1" indicates the highest
level.  You must enter names for all levels before you can initialize the
database.  You will be able to change these entries later on.

Press ESC to change the number of levels.

Press F1 for some help.

Press SHIFT-F7 when finished.
-------------------------------------------------------------------------
1.  level1                          --  item
2.  level2
3.  level3
4.  level4
```

FIGURE 2-3: Database Naming Of Levels

2.2 Entering Items Into The Database

Once the database has been created, items can be entered. This accomplished by selecting the first option on the Main EXAMINER Menu, Item Editing (Figure 1-2). Figure 2-4 illustrates the EXAMINER Item Editor Menu, which enables the user to enter new items or to amend or delete existing ones. It also allows a way to print out the database, as well as providing a printout of item statistics. Since the number of screens within this section are numerous, only a brief discussion of the Item Editing option will be presented.

The database structure is enhanced by adding a classification number at each entry. Each place where the database tree branches is called a "node" and has a unique classification number. For example, 2.0.0 indicates the second entry at the top level. 2.3.0 indicates the third entry at the second level below the second entry at the top level. And 1.2.1 indicates the first item under the second entry at the second level below the top level's first entry. Each node number uniquely identifies the position of a piece of information in the database.

Since every item is classified by a path needed to reach that item through the database, Figure 2-5 shows the screen required to do so. The current level in the database is shown in the top left corner. For example, an index of 1.3.5.?.? would indicate that the entries indicated were reached by choosing the first entry at the top level, the third entry at the second level, and the fifth entry at the third level. Whether the user is moving down an existing database structure or creating a new one, the path decided upon is reached through this screen.

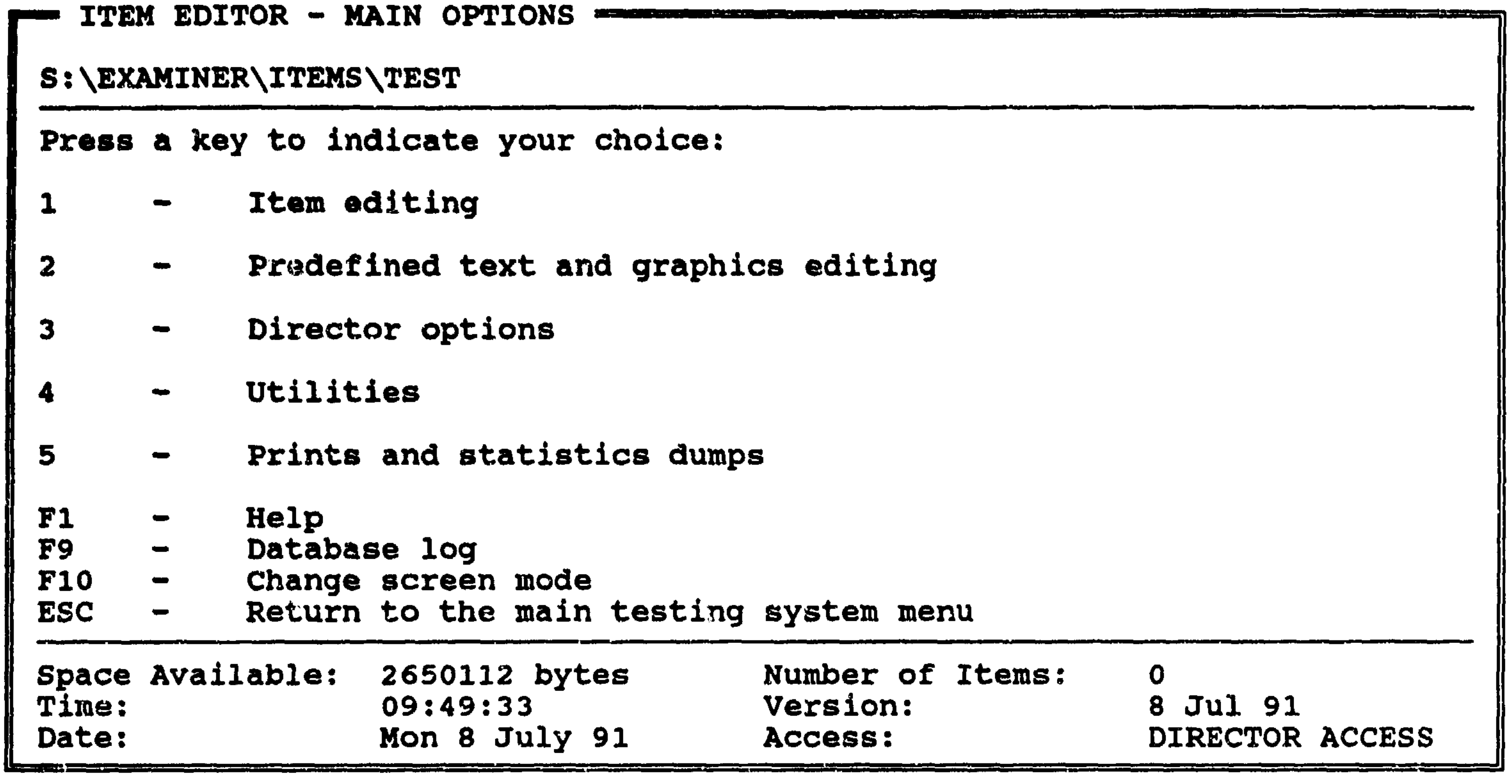
Once the item destination has been reached, the item can be added (if it's not already there), edited, copied or deleted. Figure 2-6 illustrates the General Item Editing screen used to add or edit items.

3.0 EXAMINATION EDITOR AND GENERATOR

The Examination Editor allows the user to define the nature of the examinations to be administered, and enables the examination to be produced for paper-and-pencil or computer delivery.

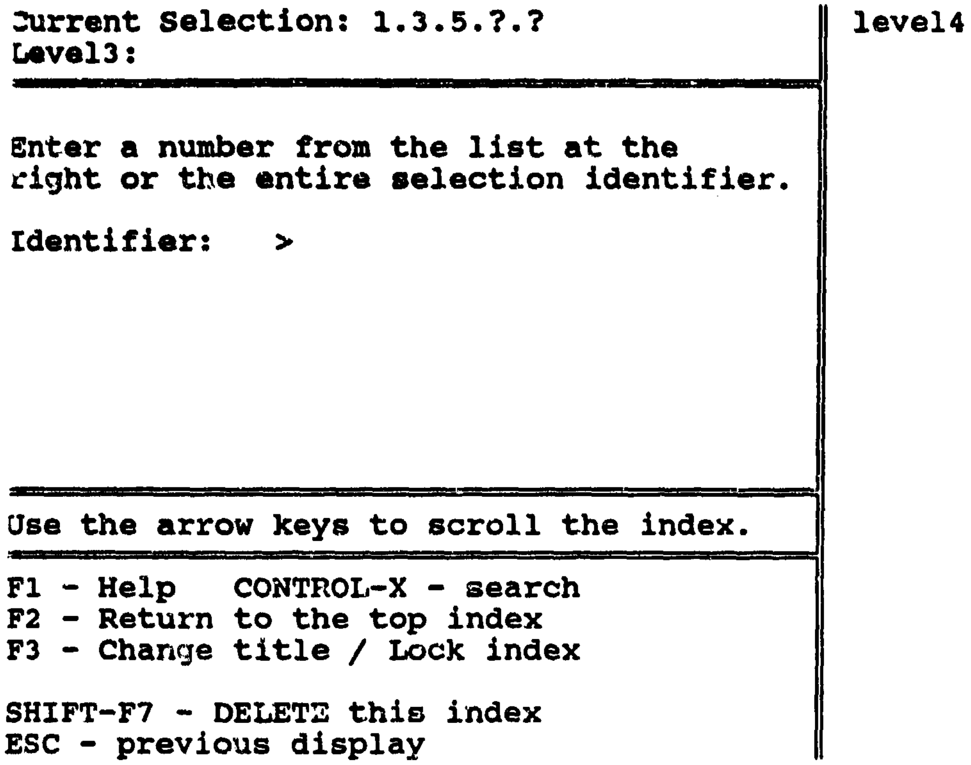
3.1 Examination Editor

Once the database contains items, option 2 from Figure 1-2 enables the user to create the test. The process that EXAMINER uses is very flexible. Descriptions of tests are contained in "profiles", which are stored on the computer. Only the description of the tests are stored, not the test themselves. When a test is



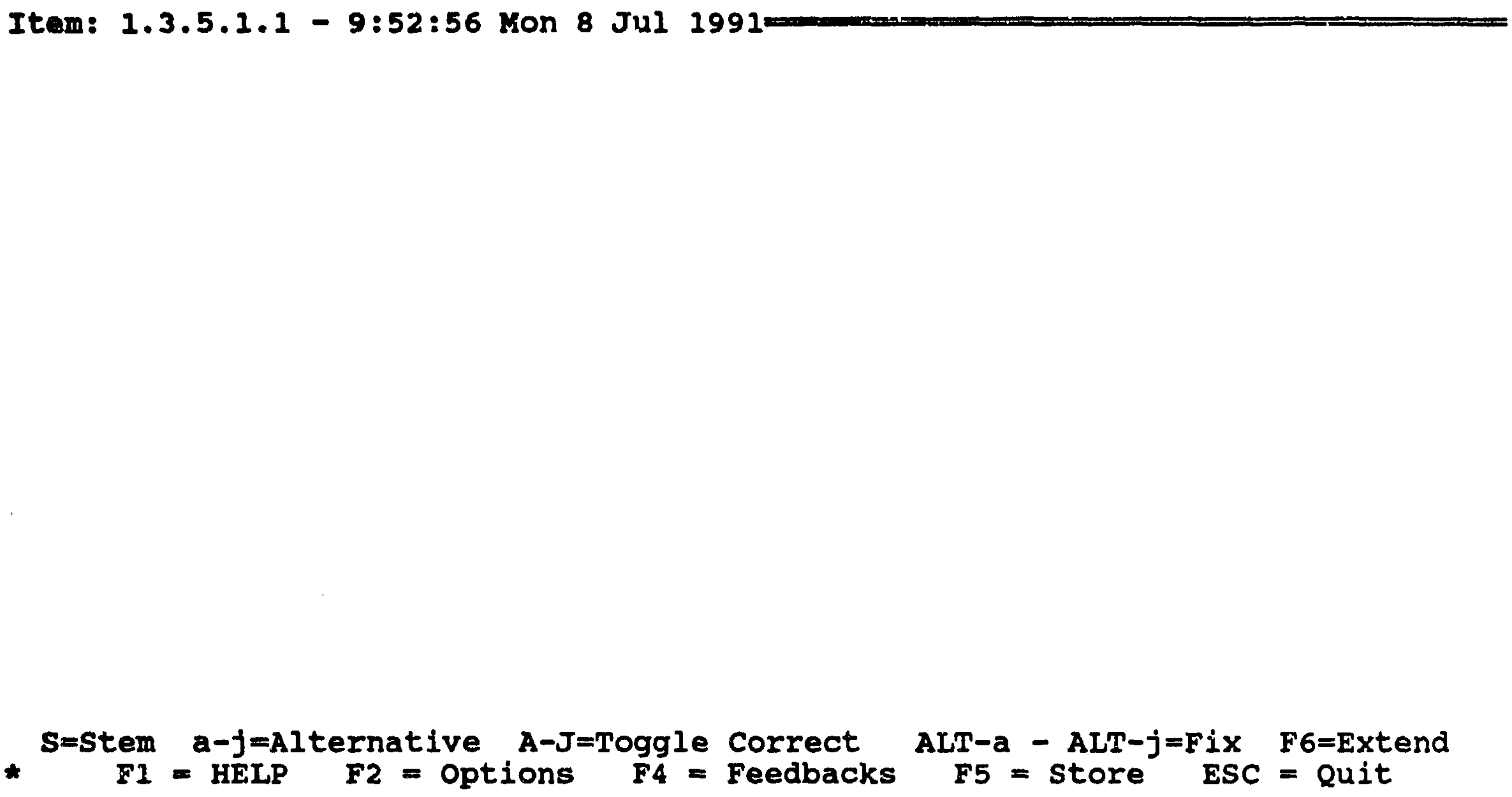
```
ITEM EDITOR - MAIN OPTIONS
S:\EXAMINER\ITEMS\TEST
Press a key to indicate your choice:

1      -    Item editing

2      -    Predefined text and graphics editing

3      -    Director options

4      -    Utilities

5      -    Prints and statistics dumps

F1     -    Help
F9     -    Database log
F10    -    Change screen mode
ESC    -    Return to the main testing system menu

Space Available:   2650112 bytes      Number of Items:    0
Time:              09:49:33           Version:            8 Jul 91
Date:              Mon 8 July 91      Access:             DIRECTOR ACCESS
```

FIGURE 2-4: EXAMINER Item Editor Menu



```
Current Selection: 1.3.5.?.?                || level4
Level3:                                     ||
============================================||
                                            ||
Enter a number from the list at the         ||
right or the entire selection identifier.   ||
                                            ||
Identifier:    >                            ||
                                            ||
============================================||
Use the arrow keys to scroll the index.     ||
============================================||
F1 - Help   CONTROL-X - search              ||
F2 - Return to the top index                ||
F3 - Change title / Lock index              ||
                                            ||
SHIFT-F7 - DELETE this index                ||
ESC - previous display                      ||
```

FIGURE 2-5: Item Editing Table



```
Item: 1.3.5.1.1 - 9:52:56 Mon 8 Jul 1991═══════════════════════════════════════════════
```

```
  S=Stem  a-j=Alternative  A-J=Toggle Correct   ALT-a - ALT-j=Fix  F6=Extend
*     F1 = HELP   F2 = Options    F4 = Feedbacks    F5 = Store    ESC = Quit
```

FIGURE 2-6: General Item Editing Screen

needed, the user retrieves the profile, specifies whether the test will be given on paper or via computer, and creates the test at that time.

There are numerous advantages to this approach. First, because the profile can be stored permanently on the computer, it can be used over and over again without redefining the contents of the test. Second, because the specific items of the test are not stored in the profile, security is enhanced. Third, because profiles can be easily created or amended, it is possible to create a new test very quickly. And fourth, the same profile can be used for paper-and-pencil or computer-based tests.

Prior to entering the Examination Editor, a password is needed. This password set is not the same as the password set required to enter the Item Editor. After entering the password, Figure 3-1 appears, the Examination Editor -- Main Menu. Whether the user is creating, editing or deleting a profile (option 1), Figure 3-2 will appear listing the profiles that are available. A profile can be either for a single or multiple test. A multiple test profile is a profile that contains other profiles. This option is useful for when a multi-section test is required.

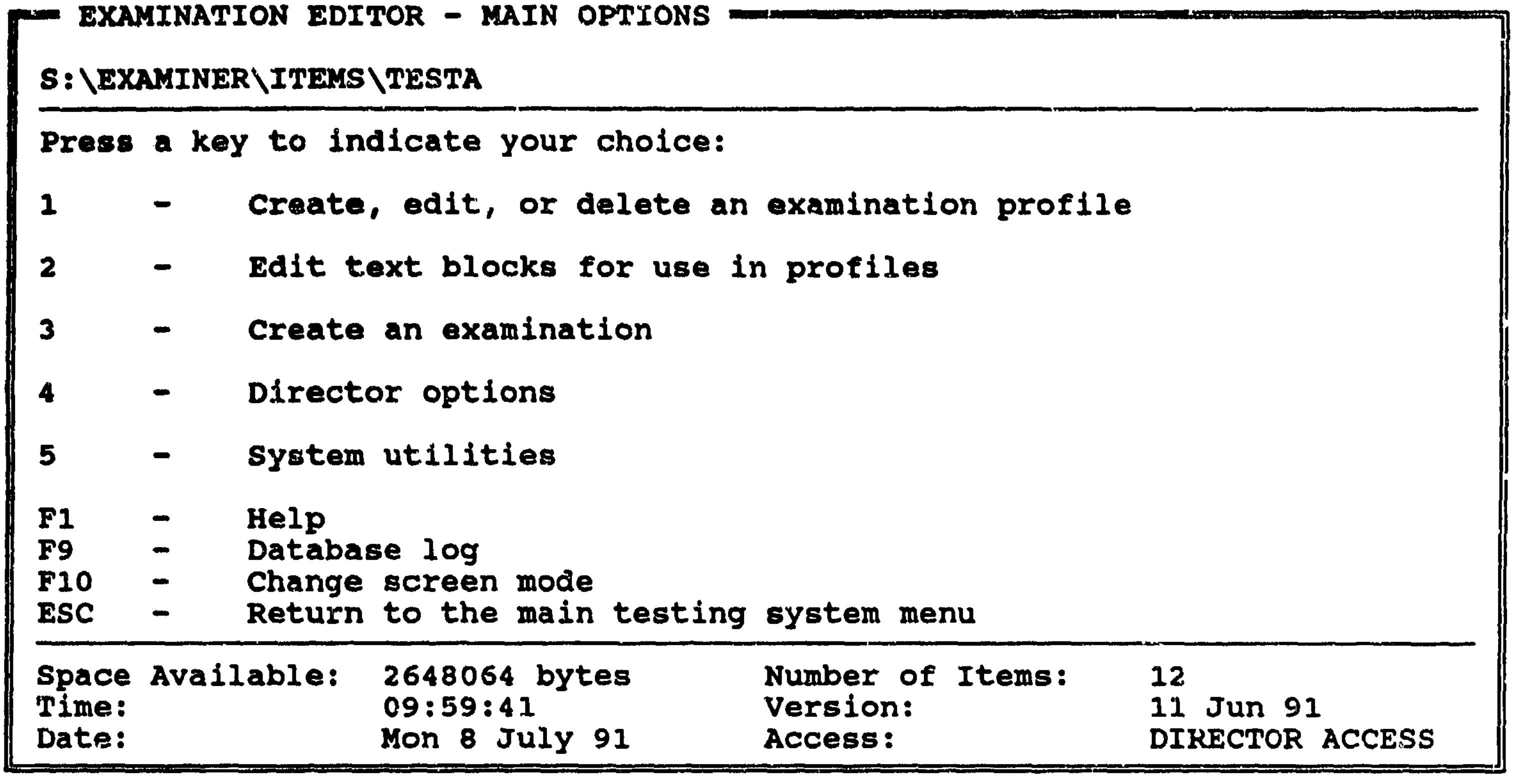
Figure 3-3 illustrates the Main Profile Menu. This menu is used whenever the user want to create, edit or view the profile. By selecting option 1 (Set Examination Parameters), options such as total number of items, testing time limits, examination pass marks, etc. can be determined.

Option 2 sets the question selection options. Here the user selects the items for the test. Figure 3-4 illustrates the three alternatives; random; stratified; and user-defined. Random selection means that ALL items are selected at random from the ENTIRE database. A stratified sampling ensures that two items are never drawn from one part of the database. While user-defined, the most powerful, allows the user to shape the classification tree in any way that he wants. By using this selection, different pools of items can be constructed. From these pools, a specific number of items can be drawn at random, or specified items can be marked for definite inclusion in the test. Figure 3-5 illustrates the "path-like" item screen used for the user-defined method.

As mentioned previously, a profile can be one that defines a multi-part test. Figure 3-6 shows how other profiles are included within a multiple profile.

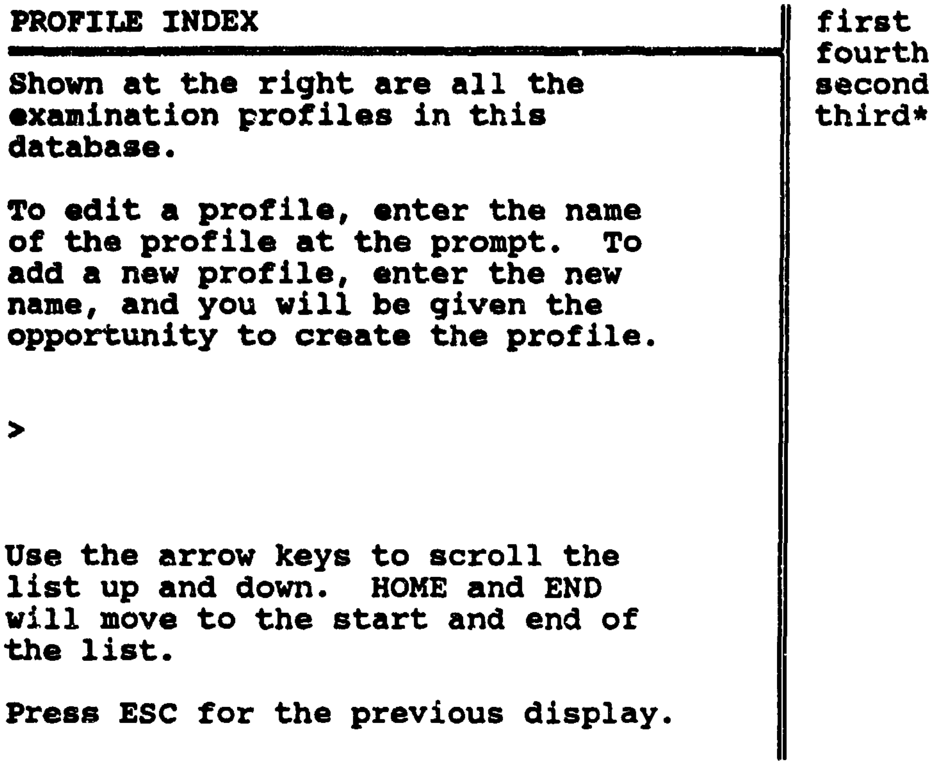
3.2 Examination Generation

Selecting option 3 from the Examination Editor -- Main Menu (Figure 3-1), enables the user to create an examination from a profile. From a list of profiles available, two types of



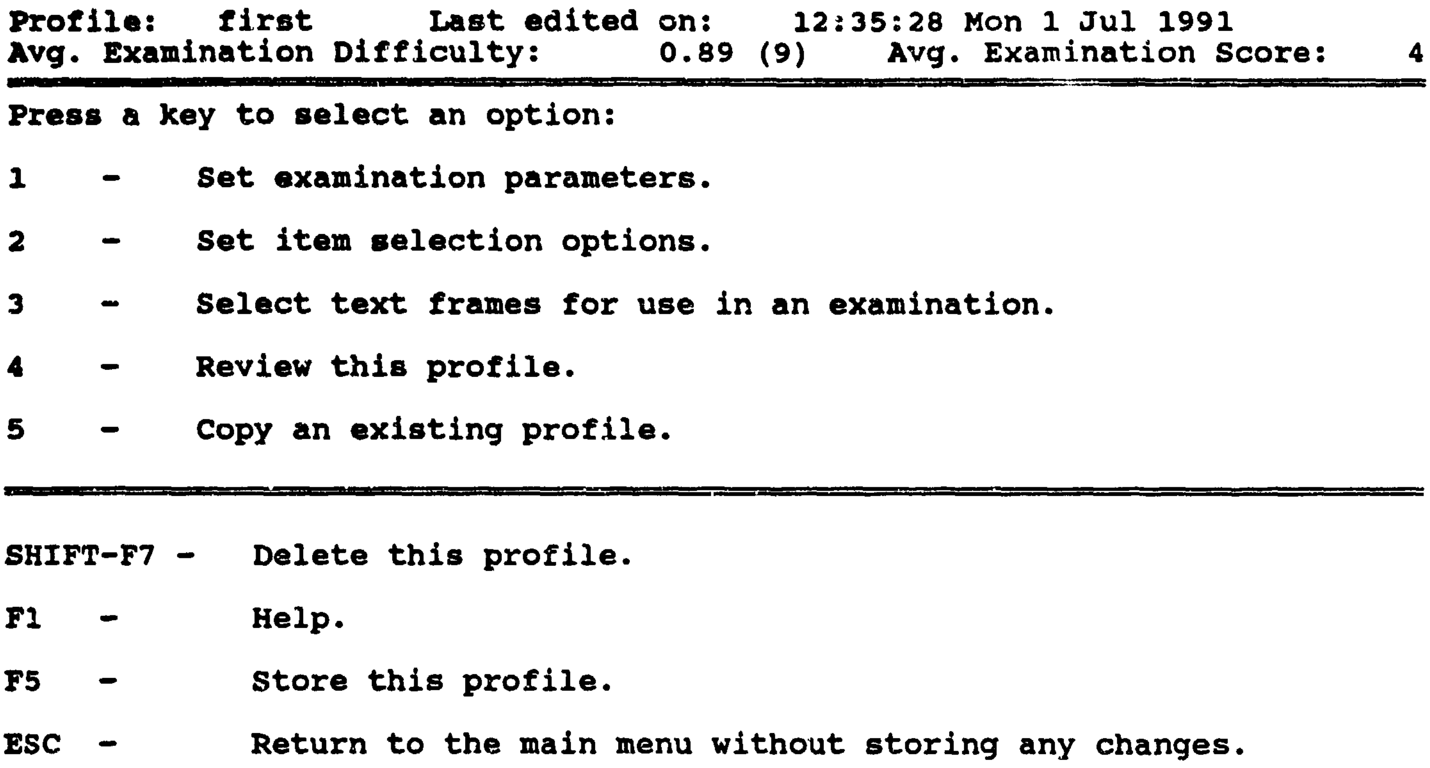
```
EXAMINATION EDITOR - MAIN OPTIONS

S:\EXAMINER\ITEMS\TESTA

Press a key to indicate your choice:

1      -    Create, edit, or delete an examination profile

2      -    Edit text blocks for use in profiles

3      -    Create an examination

4      -    Director options

5      -    System utilities

F1     -    Help
F9     -    Database log
F10    -    Change screen mode
ESC    -    Return to the main testing system menu

Space Available:  2648064 bytes      Number of Items:    12
Time:             09:59:41           Version:            11 Jun 91
Date:             Mon 8 July 91      Access:             DIRECTOR ACCESS
```

FIGURE 3-1: Examination Editor -- Main Menu



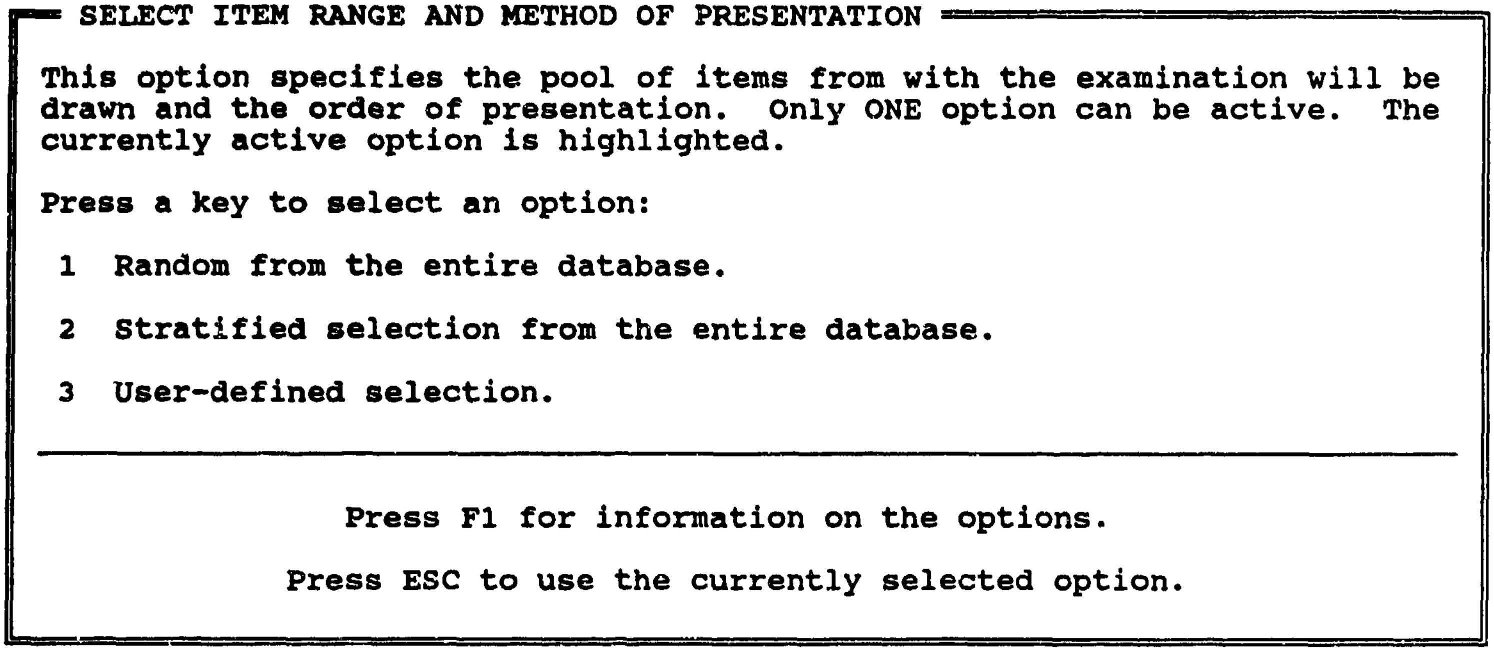
```
PROFILE INDEX                           ║ first
________________________________________║ fourth
Shown at the right are all the          ║ second
examination profiles in this            ║ third*
database.                               ║
                                        ║
To edit a profile, enter the name       ║
of the profile at the prompt.  To       ║
add a new profile, enter the new        ║
name, and you will be given the         ║
opportunity to create the profile.      ║
                                        ║
                                        ║
>                                       ║
                                        ║
                                        ║
                                        ║
Use the arrow keys to scroll the        ║
list up and down.  HOME and END         ║
will move to the start and end of       ║
the list.                               ║
                                        ║
Press ESC for the previous display.     ║
```

FIGURE 3-2: EXAMINER Profile Index



```
Profile:   first      Last edited on:    12:35:28 Mon 1 Jul 1991
Avg. Examination Difficulty:      0.89 (9)    Avg. Examination Score:    4
==========================================================================
Press a key to select an option:

1      -     Set examination parameters.

2      -     Set item selection options.

3      -     Select text frames for use in an examination.

4      -     Review this profile.

5      -     Copy an existing profile.
==========================================================================

SHIFT-F7 -      Delete this profile.

F1     -        Help.

F5     -        Store this profile.

ESC    -        Return to the main menu without storing any changes.
```

FIGURE 3-3: Main Profile Menu



SELECT ITEM RANGE AND METHOD OF PRESENTATION

This option specifies the pool of items from with the examination will be drawn and the order of presentation. Only ONE option can be active. The currently active option is highlighted.

Press a key to select an option:

1 Random from the entire database.

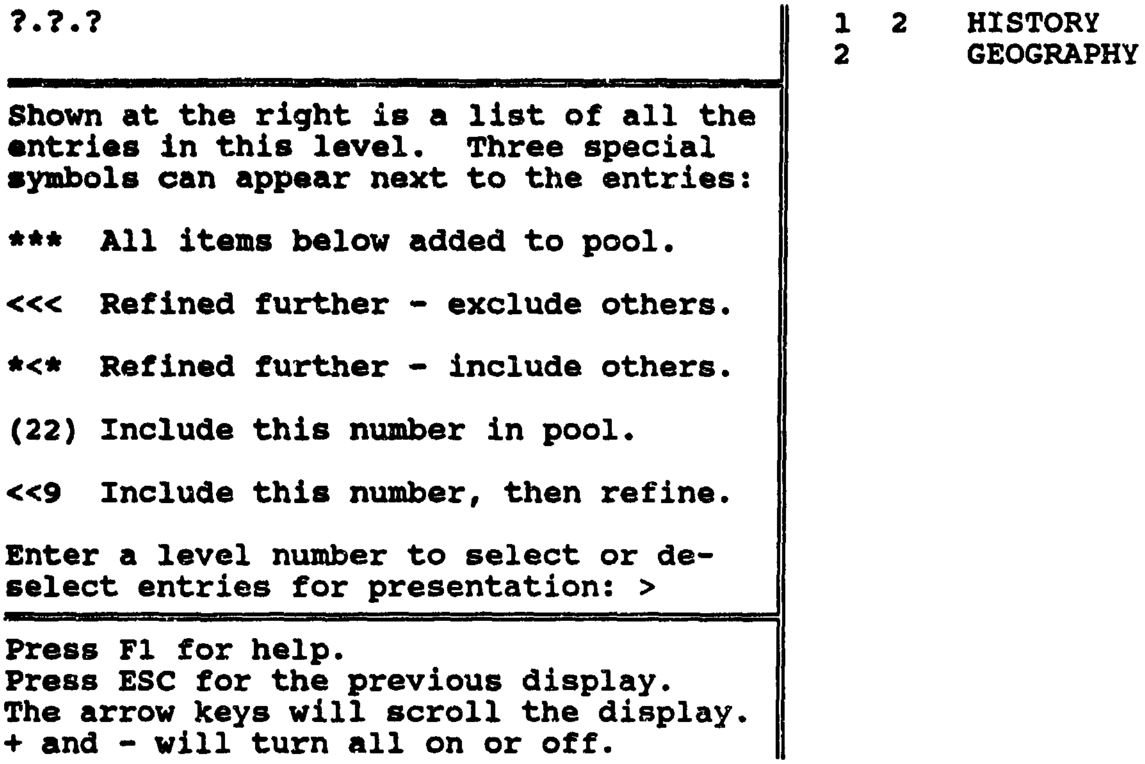
2 Stratified selection from the entire database.

3 User-defined selection.

Press F1 for information on the options.

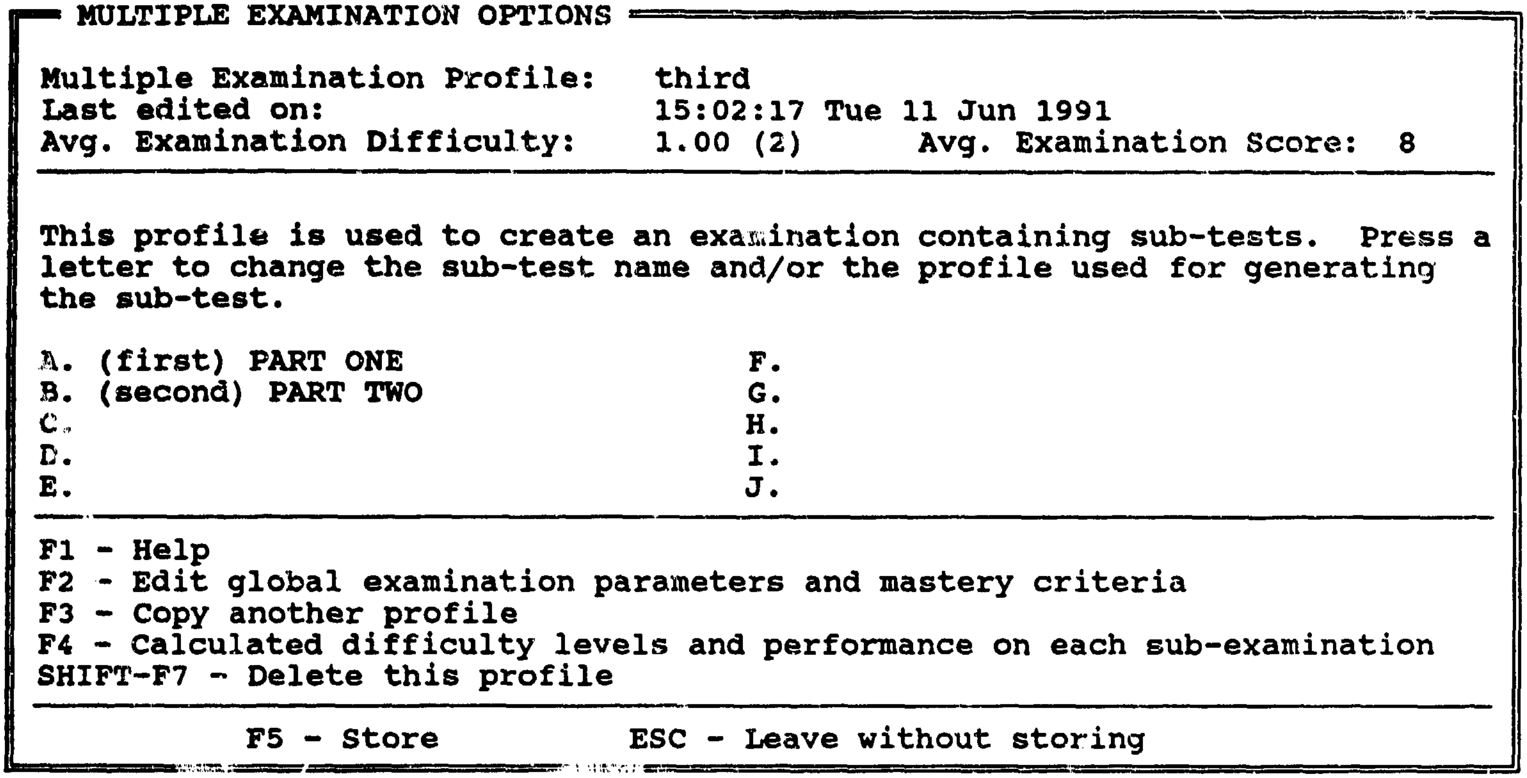
Press ESC to use the currently selected option.

FIGURE 3-4: Select Item Range and Method Of Presentation



```
?.?.?                                          1  2   HISTORY
                                               2      GEOGRAPHY
--------------------------------------------
Shown at the right is a list of all the
entries in this level.  Three special
symbols can appear next to the entries:

***  All items below added to pool.

<<<  Refined further - exclude others.

*<*  Refined further - include others.

(22) Include this number in pool.

<<9  Include this number, then refine.

Enter a level number to select or de-
select entries for presentation: >
--------------------------------------------
Press F1 for help.
Press ESC for the previous display.
The arrow keys will scroll the display.
+ and - will turn all on or off.
```

FIGURE 3-5: Item Selection Path



```
MULTIPLE EXAMINATION OPTIONS

Multiple Examination Profile:   third
Last edited on:                 15:02:17 Tue 11 Jun 1991
Avg. Examination Difficulty:    1.00 (2)      Avg. Examination Score:  8

This profile is used to create an examination containing sub-tests.  Press a
letter to change the sub-test name and/or the profile used for generating
the sub-test.

A. (first) PART ONE                  F.
B. (second) PART TWO                 G.
C.                                   H.
D.                                   I.
E.                                   J.

F1 - Help
F2 - Edit global examination parameters and mastery criteria
F3 - Copy another profile
F4 - Calculated difficulty levels and performance on each sub-examination
SHIFT-F7 - Delete this profile

          F5 - Store                ESC - Leave without storing
```

FIGURE 3-6: Multiple Test Profile Screen

examinations can be created; paper-and-pencil; and computer based. Figure 3-7 shows the screen used to determine which option is to be used. Depending on the option chosen, successive options are available. Figure 3-8 illustrates the screen used to determine computer-based examination options. Refer to the capabilities section for additional information.

4.0 STATISTICAL PROGRAM

The Statistics program provides the user with detailed statistics about items in the database, as well as about particular examinations that were given.

Figure 4-1 illustrates the Statistics Options screen used to store and access statistical information. When option 1 is chosen, Figure 4-2 appears which determines the location and type of examination results (histories) to gather.

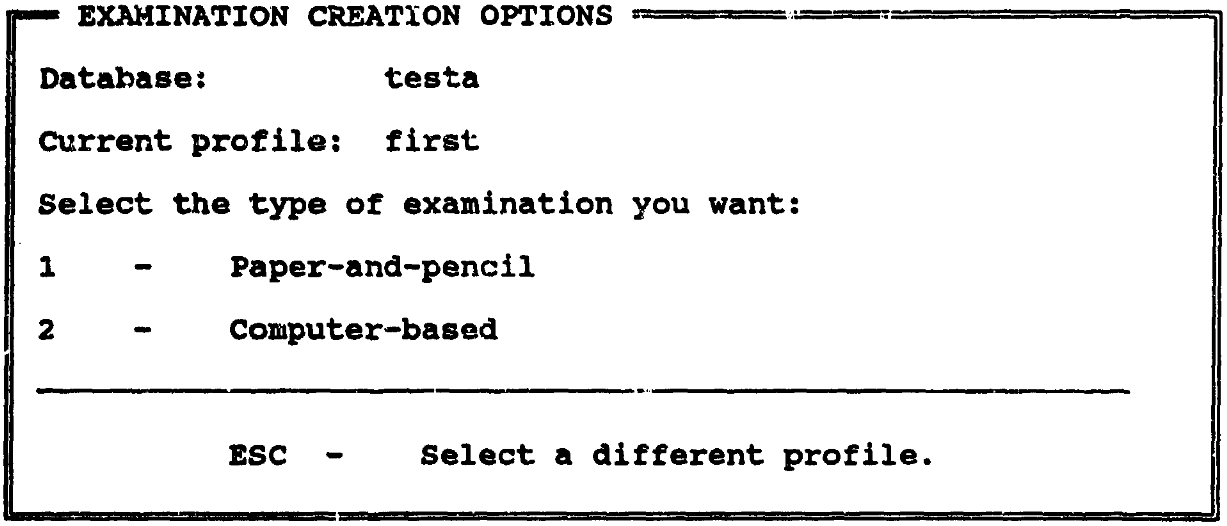
Once the histories have been stored, option 3 allows access and statistical calculation of the information. A separate password set is required for entry into this option. Once entry is obtained, Figure 4-3 appears. This screen lists all the available examinees along with their respective examination information. The record's information includes; the exam date; the examinee's name; the examinee's ID (optional); the exam database used to draw the items; the exam profile used; the exam ID (this is a unique number associated with each generated examination); and whether the examinee passed or failed.

In order to obtain the statistical information, the records need to be sorted and calculated in terms of examination ID and/or profile. Figure 4-4 illustrates the main screen used to select the calculation method (via examination ID and/or profile), while Figure 4-5 shows the screen used to sort out the examination IDs themselves.

Once the statistics have been calculated, Figure 4-6 lists some sample output. A scroll bar located on the right side of the screen, enables the user to view additional statistical information. Refer to the capabilities section for additional Statistics program abilities.

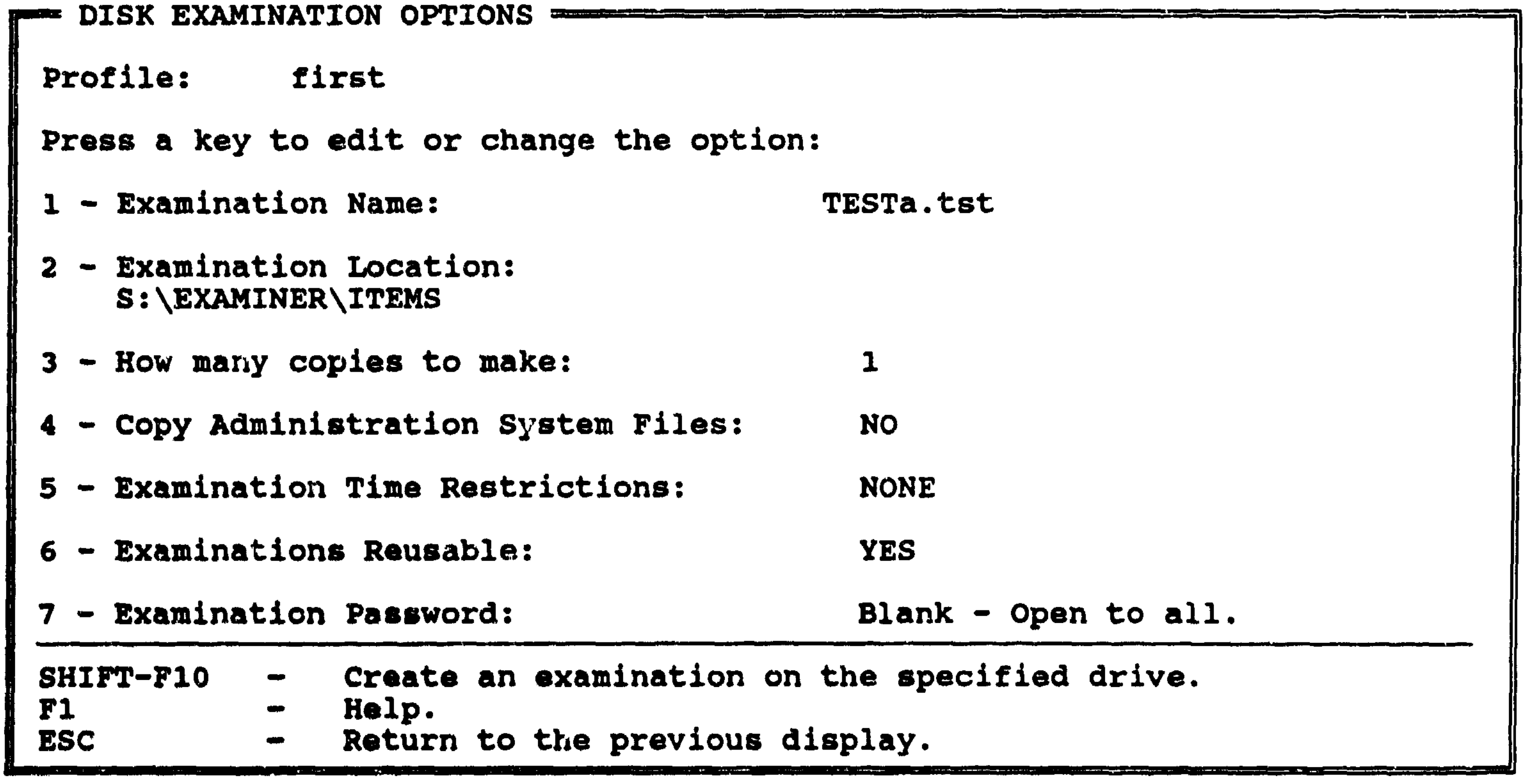
5.0 ADMINISTRATION SYSTEM

The Administration system is an integrated environment used to deliver computer-based examinations generated from profiles in the Examination Editor. The Administration system consists of two parts; the Administration Examination Backup Utility, accessible by selecting option 4 in the Main EXAMINER Option screen (Figure 1-2); and the administration program used to deliver the examinations.



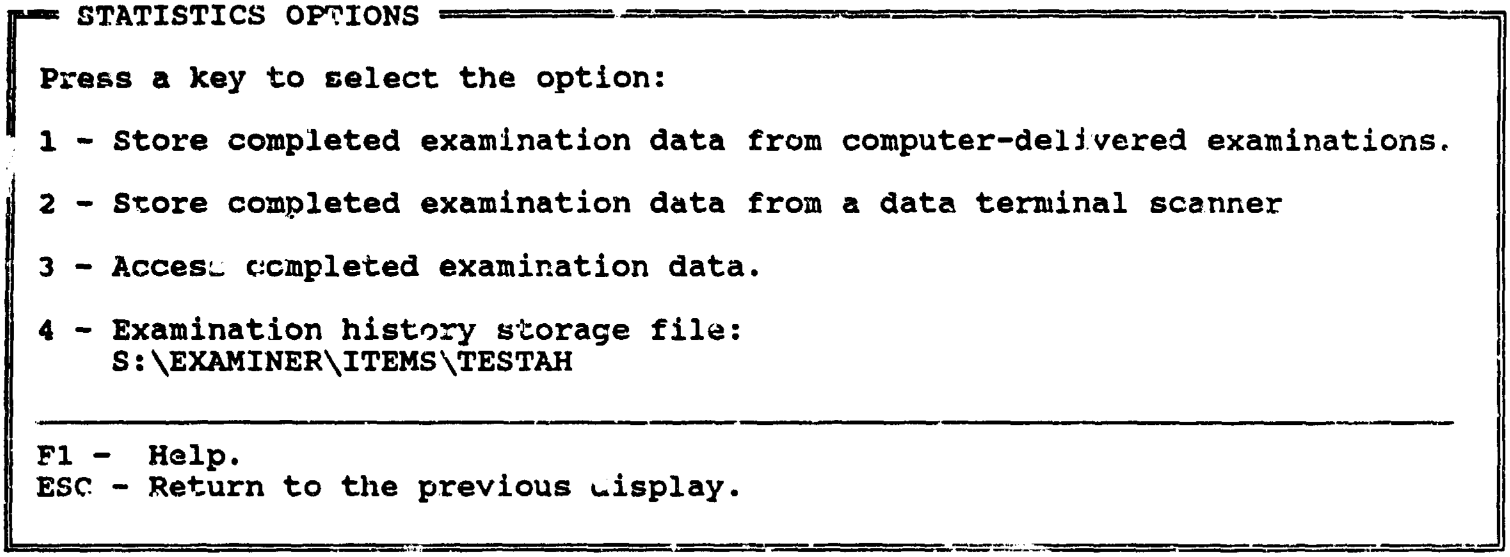
```
EXAMINATION CREATION OPTIONS

Database:          testa

Current profile:   first

Select the type of examination you want:

1     -     Paper-and-pencil

2     -     Computer-based

          ESC   -    Select a different profile.
```

FIGURE 3-7: Examination Creation Options



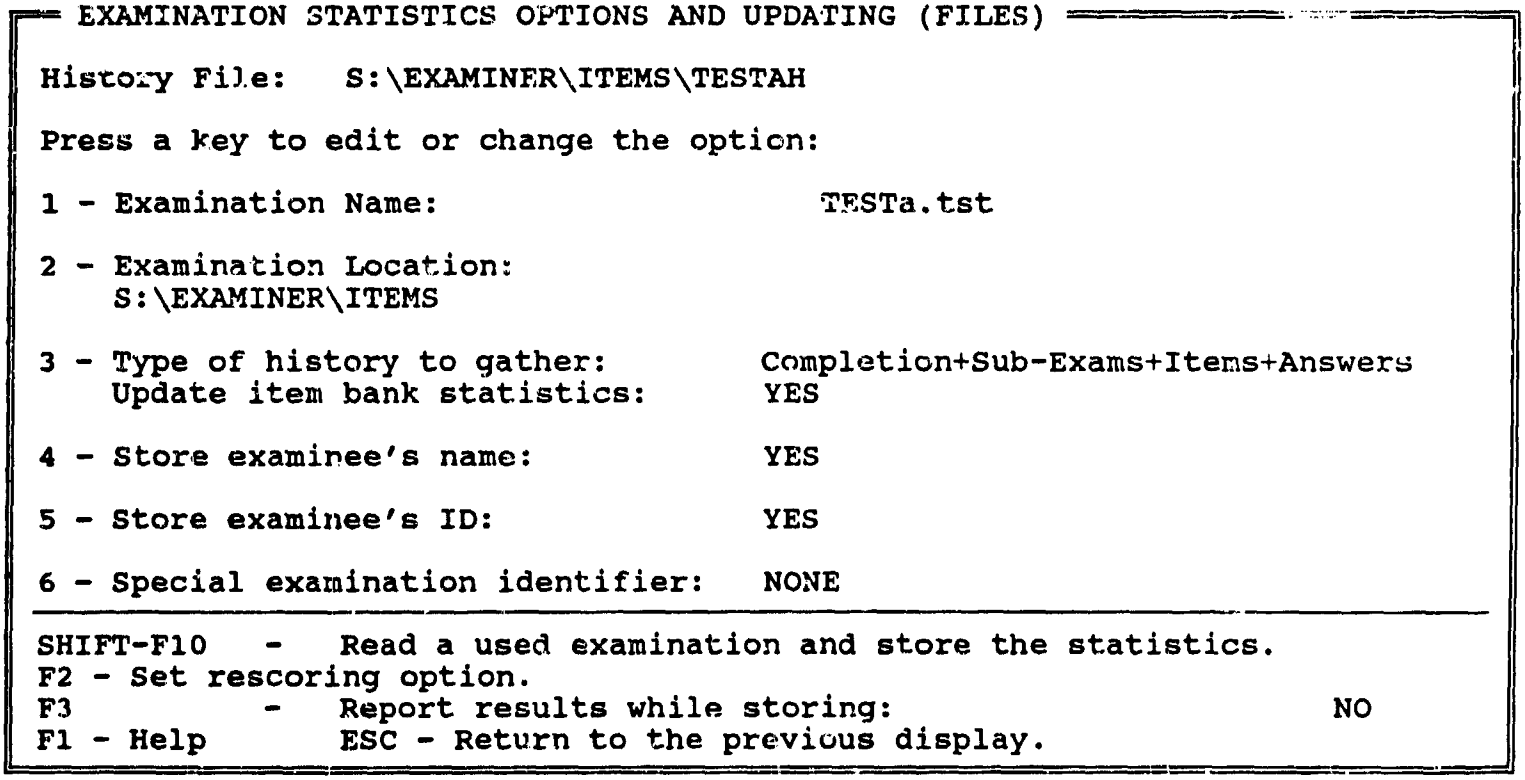
```
DISK EXAMINATION OPTIONS

Profile:     first

Press a key to edit or change the option:

1 - Examination Name:                          TESTa.tst

2 - Examination Location:
    S:\EXAMINER\ITEMS

3 - How many copies to make:                    1

4 - Copy Administration System Files:           NO

5 - Examination Time Restrictions:              NONE

6 - Examinations Reusable:                      YES

7 - Examination Password:                       Blank - Open to all.

SHIFT-F10    -    Create an examination on the specified drive.
F1           -    Help.
ESC          -    Return to the previous display.
```

FIGURE 3-8: Computer Examination Options



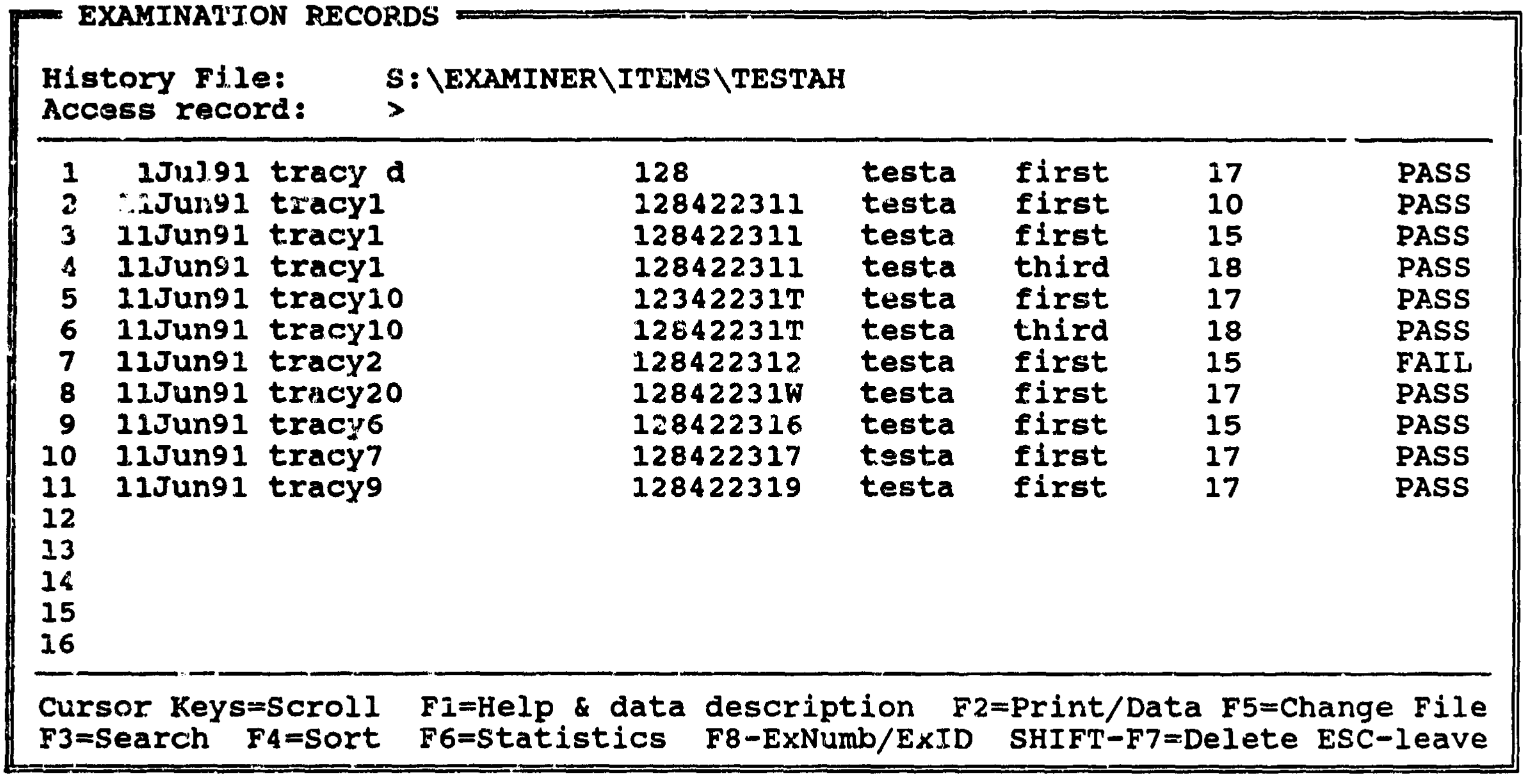
```
STATISTICS OPTIONS

Press a key to select the option:

1 - Store completed examination data from computer-delivered examinations.

2 - Store completed examination data from a data terminal scanner

3 - Access completed examination data.

4 - Examination history storage file:
    S:\EXAMINER\ITEMS\TESTAH

F1 -  Help.
ESC - Return to the previous display.
```

FIGURE 4-1: Statistics Options



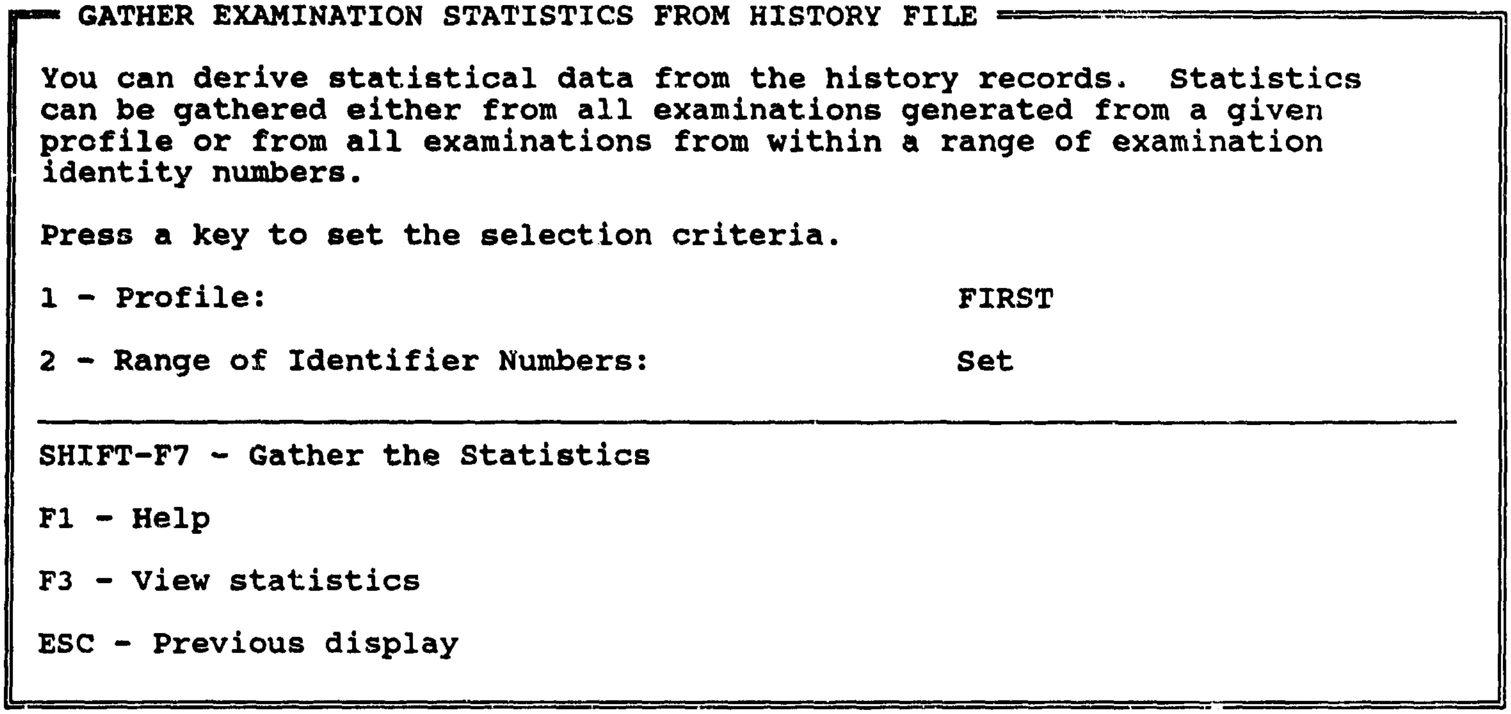
```
EXAMINATION STATISTICS OPTIONS AND UPDATING (FILES)

History File:   S:\EXAMINER\ITEMS\TESTAH

Press a key to edit or change the option:

1 - Examination Name:                           TESTa.tst

2 - Examination Location:
    S:\EXAMINER\ITEMS

3 - Type of history to gather:          Completion+Sub-Exams+Items+Answers
    Update item bank statistics:        YES

4 - Store examinee's name:              YES

5 - Store examinee's ID:                YES

6 - Special examination identifier:     NONE

SHIFT-F10    -   Read a used examination and store the statistics.
F2 - Set rescoring option.
F3           -   Report results while storing:                        NO
F1 - Help        ESC - Return to the previous display.
```

FIGURE 4-2: Examination Statistics Options
and Updating Files



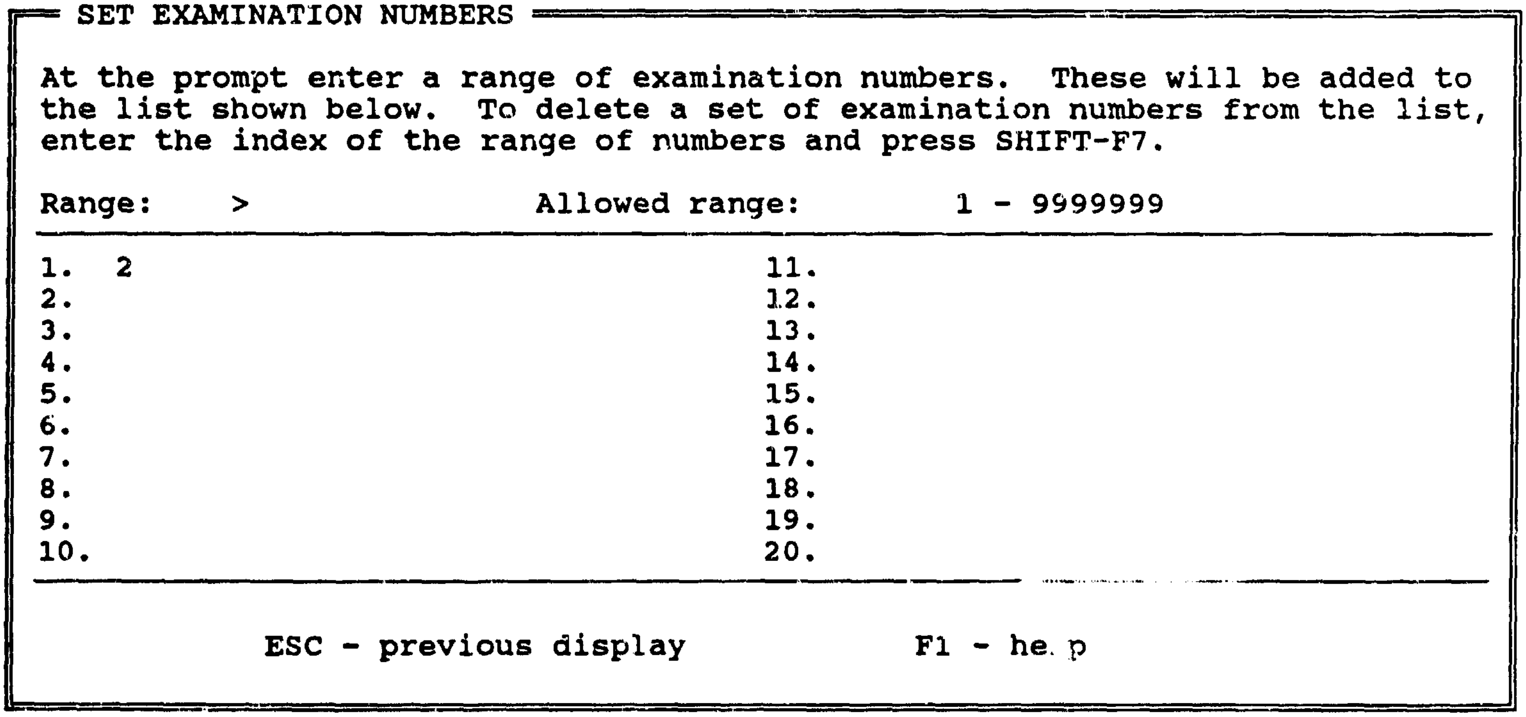
```
EXAMINATION RECORDS

History File:     S:\EXAMINER\ITEMS\TESTAH
Access record:    >
--------------------------------------------------------------------------------
 1   1Jul91 tracy d           128        testa   first     17         PASS
 2  11Jun91 tracy1            128422311  testa   first     10         PASS
 3  11Jun91 tracy1            128422311  testa   first     15         PASS
 4  11Jun91 tracy1            128422311  testa   third     18         PASS
 5  11Jun91 tracy10           12342231T  testa   first     17         PASS
 6  11Jun91 tracy10           12842231T  testa   third     18         PASS
 7  11Jun91 tracy2            128422312  testa   first     15         FAIL
 8  11Jun91 tracy20           12842231W  testa   first     17         PASS
 9  11Jun91 tracy6            128422316  testa   first     15         PASS
10  11Jun91 tracy7            128422317  testa   first     17         PASS
11  11Jun91 tracy9            128422319  testa   first     17         PASS
12
13
14
15
16
--------------------------------------------------------------------------------
Cursor Keys=Scroll  F1=Help & data description  F2=Print/Data F5=Change File
F3=Search  F4=Sort  F6=Statistics  F8-ExNumb/ExID  SHIFT-F7=Delete ESC-leave
```

FIGURE 4-3: Examination Records



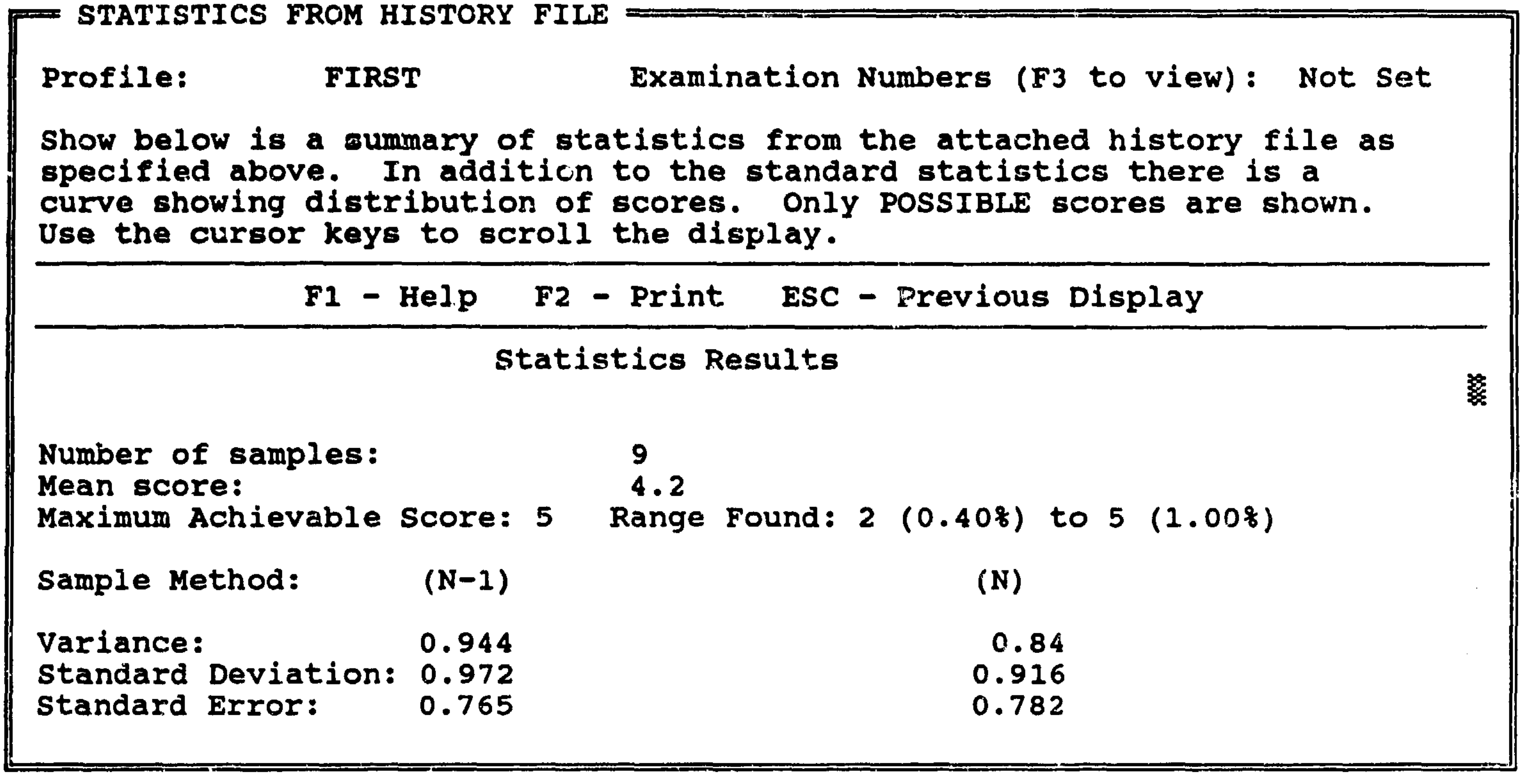
```
GATHER EXAMINATION STATISTICS FROM HISTORY FILE

You can derive statistical data from the history records.  Statistics
can be gathered either from all examinations generated from a given
profile or from all examinations from within a range of examination
identity numbers.

Press a key to set the selection criteria.

1 - Profile:                                       FIRST

2 - Range of Identifier Numbers:                   Set

SHIFT-F7 - Gather the Statistics

F1 - Help

F3 - View statistics

ESC - Previous display
```

FIGURE 4-4: Gather Examination Statistics From The History File



```
SET EXAMINATION NUMBERS

At the prompt enter a range of examination numbers.  These will be added to
the list shown below.  To delete a set of examination numbers from the list,
enter the index of the range of numbers and press SHIFT-F7.

Range:    >                Allowed range:        1 - 9999999

1.   2                                  11.
2.                                      12.
3.                                      13.
4.                                      14.
5.                                      15.
6.                                      16.
7.                                      17.
8.                                      18.
9.                                      19.
10.                                     20.

           ESC - previous display            F1 - help
```

FIGURE 4-5: Statistical Examination Numbers



```
STATISTICS FROM HISTORY FILE

Profile:       FIRST          Examination Numbers (F3 to view):  Not Set

Show below is a summary of statistics from the attached history file as
specified above.  In addition to the standard statistics there is a
curve showing distribution of scores.  Only POSSIBLE scores are shown.
Use the cursor keys to scroll the display.

              F1 - Help   F2 - Print   ESC - Previous Display

                          Statistics Results

Number of samples:            9
Mean score:                   4.2
Maximum Achievable Score: 5   Range Found: 2 (0.40%) to 5 (1.00%)

Sample Method:       (N-1)                          (N)

Variance:            0.944                           0.84
Standard Deviation: 0.972                           0.916
Standard Error:      0.765                          0.782
```

FIGURE 4-6: Statistical Results From History File

The administration program which delivers the examination isn't located within the EXAMINER system. Access to this program requires exiting EXAMINER and running the command file ADMIN using command line options within the DOS environment. For example, the command line:

```
C:\EXAMINER> admin -ts:\examiner\testa.tst -full -r
```

will administer the test Testa.tst, which is located in drive S: under directory Examiner. In addition, a full examinee history will be created, along with the ability for the test to be administered again (-full and -r).

6.0 EXAMINER CAPABILITIES

The following lists the capabilities of EXAMINER.

6.1 Item Editor

- Access via password. The level of access depends on the password used.
- The database structure can range from 2 to 7 levels.
- Each database level must be named; however, the names can be changed.
- Items can have up to five category assignments. This is useful for exam creation.
- Capability to add or delete items by creating or removing "nodes on the tree".
- Wide range of item types:
  - Multiple Choice (One or more answer with up to ten alternatives; extended-increased writing field; dynamic-system generated alternatives)
  - Numeric (Answer within an absolute or percentage range.)
  - Alpha (Option for extra words and misspellings.)
  - Dual (Two pair statement in which the system generates the alternatives.)
  - Linked Item (Series of items linked as one scenario.)
  - Parallel (Series of items which test the same content and direct the system to randomly select from these items.)
  - Copy Items (Copy items from the current or other databases.)
- Items can be weighed with weights ranging from 0 to 100.
- Items and indices can be locked out.

- Predefined text enables:
  - identical paragraphs to be used repeatedly or not at all,
  - the text itself can be "locked out" from other users,
  - graphical ability.

- Director options enable:
  - name changes of database levels,
  - item classification category changes,
  - password changes,
  - database release version changes,
  - database compression (defragmentation),
  - verification prints of exams; records to ensure exam completion.

- System utility options enable:
  - DOS command execution while still within EXAMINER,
  - external ASCII files, which contain item statistics, to be read in,
  - old database statistics to be deleted.

- Prints and statistics dumps enable:
  - items and their associated statistical information to be printed,
  - formatted, statistical dumps sent to a printer or an output disk file,
  - item statistics dumped to ASCII datafiles, which can be further used by external programs,
  - production of an ASCII datafile compatible with the EXAMINER text import program; good when statistics are remotely generated.

6.2 Examination Editor and Generator

- Access via password. Level of access depends on the password used.

- Test format based on a "profile", which can be reviewed, stored, copied, updated or deleted. The items themselves aren't stored.

- Can create single or multiple (profiles within a profile) profiles.

- Profiles can be used repeatedly, as either computer or paper based tests or within a multiple profile.

- Particular exam parameters may be set. These include:
  - test content from 1 to 500 items,
  - passing criteria based on an absolute number or percentage of total points,
  - exam time limits,

- changes to the presentation of item and/or alternative order,
- exam difficulty levels,
- user ability to exit exam, change answers or observe item feedback,
- adjustable exam composition; mix of question types or categories,
- ability to display mastery level requirements and exam results.

- Test contents can be chosen at random, stratified (every sublevel is fairly represented at random) or strictly user-defined.

- Text frames for example, introduction and instruction screens, can be included within the exam at various points.

- Computerized tests can:
  - be stored on either hard or floppy disks under any filename,
  - include the administration system on a single diskette,
  - include a specified range of examination testing dates,
  - be used repeatedly via the -r option within the administration system command line,
  - include a password for exam security.

- Paper tests can:
  - include an optional examination title,
  - have multiple choice alternatives labeled with letters or numbers,
  - include the phrase "Enter your answer" after each item,
  - produce answer and key sheets which under certain conditions, can be used with optical scanners,
  - have up to 999 test copies produced.

- Numerous profiles can be linked together to produce a single, multi-test profile. This multiple profile includes the following capabilities:
  - parameters specified within the original sub-test profiles will remain in effect, with the exception of time limits and particular scoring criteria,
  - options set within the multi-exam profile will override any similar options in the referenced profiles,
  - scoring criteria can include passing all sub-exams, passing a certain number of sub-exams or passing all sub-exams plus achieving a certain number or percentage of points across the entire exam,
  - exam time limits can be changed or not exist,
  - the alternative order can be changed,
  - examinee feedback, review and restart options can be set,
  - sub-exams can be intermixed in numerous ways; randomly, in increasing or decreasing difficulty, according to categories or not at all,

- predefined text frames can be included,
- items can be excluded according to certain dates,
- examinee mastery requirements and feedback can be displayed.

6.3 Statistical Program

- Ability to store completed data from either computer or scanner delivered exams. Up to 5000 records can be stored.

- The following data storage options can be implemented:
  - store examination name and location,
  - the type(s) of history information gathered,
  - store examinee's name or ID with history information,
  - store under special identifiers; used to label groupings of examinees.

- The levels of detail for history information include:
  - no long term history data,
  - exam completion information only; one line is added to the overall history file for each examinee; this is the least amount of information that can be sent,
  - gather the above PLUS sub-examination results,
  - gather the above PLUS individual item results; this includes what items were shown to the examinee, and whether the examinee passed or failed the item,
  - gather the above PLUS show the examinee's answers,
  - a flag whether or not item bank item statistics will be updated along with the listing file.

- To access the completed history file requires a password. The level of access is dependent upon the password used.

- All exam histories can be shown. Special function keys enable various abilities, such as:
  - print data to a printer, a file or to a "flat ASCII" file which can be read by an external program,
  - setting search criteria for displaying and printing individual and class performance,
  - changing the order in which individual history data is displayed,
  - changing the history file being accessed,
  - a toggle switch that can show a unique exam ID or special identifier,
  - deleting a range of history records and/or backing out their statistics,
  - deriving various statistical information for a class or section.

- Statistical information includes:
  - statistics gathered either from all exams generated from a given profile, or from all exams within a range of examination IDs; up to twenty IDs can be included,
  - the total number of samples used,
  - the mean, maximum achievable and range of test scores,
  - an (N-1) or (N) sampling method, which calculates information such as, variance, standard deviation, standard error and Kuder-Richardson 21 values,
  - a histogram showing the number of examinees that received each percentage score.

- It is possible to access an individual record. This will print a detailed history file, which includes a general and detailed description of the exam and a line by line listing of the examinee's items. An individual item can be viewed, if the history was obtained from an exam generated under the current database.

6.4 Administration System

- This utility administers exams using a variety of command line arguments. These arguments allow various exam testing methods, full histories, exam location, etc. It is possible to use these command lines within a batch file. In addition, options exist in order to rebuild damaged databases or indices.

- The Administration Examination Backup Utility insures the ability to retrieve and maintain backup files created by the Administration program. In addition, this utility allows the Director to set or change passwords that control access to the Administration System Control program.

6.5 General

- The Text Import Utility uses command line arguments to enable EXAMINER to read in plain ASCII files generated by other word processing systems. Entire databases can be created with this method.

- The EXAMINER is very user-friendly and help screens are readily available throughout the system.

7.0 EXAMINER LIMITATIONS

The following lists the limitations of EXAMINER.

7.1 Item Editor

- The number of database levels cannot be changed once the database has been created.
- Extended multiple choice cannot be changed back to "normal" mode.
- In order for graphics to be performed, a special graphics package called "PC Paintbrush", needs to be purchased separately.
- Graphics are not possible with items that are multiple choice extended or dynamic.
- It isn't possible to change the name at the ITEM level.
- Verification print could compromise the security of the exam.

7.2 Examination Editor and Generator

- If there are not a sufficient number of test items called for within the test, and the items are randomly chosen from a database with numerous sublevels, certain sublevels may not contribute any items.
- If the clock option is used to administer the exam via computer, it is imperative to ensure that the system clock is correct.
- If the computerized exam is to be used more than once, the examinee's results must be stored immediately using the Administration System's history option or by having the file read in directly via the main systems statistics option. If this is not performed prior to the next administration of the examination, the examinee's testing data will be lost.
- Commercially available scanning sheets for examinee testing should be used, if the test data is to be read in via optical methods.
- If the profile is in multiple test format, in order to pass the exam, either ALL or SOME subtests have to have been passed PLUS a certain number of achievement points throughout the test need to have been reached. The passing criteria is not solely based on the total number or percentage of points obtained from the ENTIRE exam.
- If the feedback and allowance of exam item change options are enabled at the same time, the examinee then has the

means to master every item within the exam.

7.3 Statistical Program

- If statistics are performed on data composed of different exam profiles, then certain statistical scores may be meaningless.

7.4 Administration System

- If the -full option is not used within the command line, not all statistical information will be saved.

7.5 General

- If Director passwords are forgotten, access to various EXAMINER routines will be jeopardized.

- Passwords need to be typed in with the "Caps Lock" key turned on, or access will be denied.

- Passwords should vary slightly between the EXAMINER functions. If not, the system sometimes gets "confused" and access will be denied even though the password is correct.

11.0 APPENDIX D
Review of MICROCAT

A BRIEF SUMMARY OF
THE MICROCAT (tm) TESTING SYSTEM

by
Fu Li
Statcom Inc.

for
US Army Research Institute
5001 Eisenhower Avenue
Alexandria, Va 22333-5600

1.0 INTRODUCTION

The MicroCAT(tm) Testing System is a complete microcomputer-based system for developing, administering, scoring and analyzing computerized tests. It also supports the printing of tests for traditional paper-and-pencil administration. The system is consisted with four subsystems:

1) The Development Subsystem -- To create new tests and test items

2) The Examination Subsystem -- To administer and score tests

3) The Assessment Subsystem -- To assess the measurement properties of items and tests

4) The Conventional Testing Subsystem -- To create, print, and score conventional tests

You may install the MicroCAT system to use a menu system based on a batch file or menus based on Windows. The Window-based system is more convenient to use; the batch file system requires less memory and is easier to customize.

2.0 BASIC FUNCTIONS IN THE DEVELOPMENT SUBSYSTEM

The basic functions of the Development subsystem are listed as follows:

1)	Bank	-- Enter or edit test items.
2)	MakeFont fname	-- Enter or edit special text characters.
3)	Create	-- Create a test specification using a template.
4)	Edit fname	-- Edit the test specification.
5)	Compile	-- Compile a test specification.
6)	Import	-- Import text items from a file.
7)	Utility	-- Use item bank utilities.

The MicroCAT Graphics Item Banker provides procedures for interacting entering, revising, and storing test items (and instructional screens) organized by content areas. With each item or instructional screen, certain characteristics are stored in addition to the actual item content, such as, the item's ID, the response type, the IDs of pre- or post-items that must always precede or follow the presentation of the item, and so on. Please see Appendixes Figure 1 for the submenu of Bank, and Figure 2 for the definition of characteristics of a item.

The Font Generation is used to create special text characters that can be included in test items and instructional screens developed using Banker. Please see Appendixes Figure 3. Submenu of Font Generation Program. There is a 8 X 8 grid in the middle of the screen which can not be printed out due to the different graphics mode. The user will create a special character in the grid.

There are eight predefined templates of test specification in the function of Create. They can be used to specify a wide variety of tests using your own items and scoring criteria. The result is a complete test specification file written in MCATL. You can also design your own templates for specific assessment needs.

The function of Edit is used for entering and editing the test specification files. A text editor is not provided with the MicroCAT system. The default editor invoked by the EDIT function is the IBM Personal Editor. The EDIT function can be customized to invoke any editor you choose to use with the system.

The function of Compile will translate test specification files into a form that the computer can efficiently process to administer tests (executable files).

The function of Import converts items that contain text only (i.e., no graphics) from a simple format in an ASCII file to the MicroCAT item bank format.

The function of Utility provides six features that may be useful to the advanced user of the MicroCAT Testing System. These features include the capability to copy, merge, pack, rename, and salvage banks, as well as the capability to recover items that have accidentally been erased. Please see Appendixes Figure 4 for the submenu of Utility.

3.0 BASIC FUNCTIONS IN THE EXAMINATION SUBSYSTEM

The basic functions of the Examination subsystem are listed as follows:

1) Testone -- Test a single examinee and then return to this menu.
2) Testmany -- Test examinees continuously, one after the other.
3) Clear -- Clear the log file on the diskette in drive A.

Test administration can be initiated by one of two functions: Testone or Testmany. Testone administers a test to one examinee; Testmany administers tests to examinees one after the other. You can log an examinee's responses on a diskette so that you can recover his or her test if administration is interrupted. If you do, you must erase the loq file created for that examinee before attempting to administer a test to another examinee using the same diskette.

4.0 BASIC FUNCTIONS IN THE ASSESSMENT SUBSYSTEM

The basic functions of the Assessment subsystem are listed as follows:

1) Collect -- Collect individual examinee data for analysis.
2) Analyze -- Perform conventional test and item analyses.
3) Restimate -- Estimate item parameters for the Rasch model.
4) Estimate -- Estimate item parameters for the three -parameter model.
5) Evaluate -- Evaluate a test using its item parameters.
6) Validate -- Perform test validation analyses.

The function of Item analysis requires that the input data be formatted in ASCII files. You can format the data using a text editor, a word-processing editor that produces true ASCII output, or a program you write to format the data. You can also use the function of Collect to collect data from individual examinee response files and consolidate them in a single file that can be used in the function of Analysis. This Collect function works only for test records that result from tests created using the CALIB template.

The function of Analysis will produce conventional item and scale statistics. It can also be used to compute scale scores for individual examinees. These statistics can be used to determine which items should be incorporated into an IRT (Item Response Theory) item calibration project or whether a particular scale is a reliable measure of the trait it assesses.

The function of Analysis will allow you to analyze scales containing either dichotomously scored (e.g., correct-incorrect scoring) or multipoint items. The outputs of these two scales will both include the item's sequence number (Seq. No.), the scale number to which the item was assigned and the item's sequence number within the scale (Scale-Item), and the proportion of the right answers (Prop. Correct), the biserial correlation between correct responses to the item and number-correct scores on the scale to which the item was assigned (Biser), and the point-biserial correlation between correct responses on the item and the total scale scores (Point-Biser) for Dichotomously scored items, and the average response to the item (Item Mean), the variance of the responses to the item (Item Var), the Pearson correlation between responses to the item and average scores for examinees (Item-Scale Correlation), and the number of cases on which the item statistics are based (N per Item) for Multipoint items respectively.

The function of Restimate is to use the Rasch, or one-parameter logistic (the difficulty parameter), IRT model to scale the parameter estimates either traditionally by setting the mean item difficulty to 0 or consistently with ASCAL and the rest of the MicroCAT Testing System by standardizing the distribution of true ability. The former is recommended for use of the parameters outside of the MicroCAT system; the latter is recommended for use within the system.

The function of Estimate computes the discrimination, difficulty, and guessing parameters for the two- and three-parameter logistic IRT models. It uses a combined maximum likelihood and modal Bayesian estimation procedure.

The function of Evaluation uses IRT to predict characteristics of newly developed tests using previously calibrated items and their parameters. The function provides mean item parameters, estimated test reliabilities, and several test information estimates.

The function of Validation can be used to investigate the criterion-related validity of a set of test scores or other variables. It provides descriptive summary statistics for both the predictor variables and the criterion. It computes the multiple-regression equation for predicting the criterion from the set of predictor variables. It also computes the multiple correlation between the criterion and the set of predictor variables.

5.0 BASIC FUNCTIONS IN THE CONVENTIONAL TESTING SUBSYSTEM

The basic functions of the Conventional Testing subsystem are listed as follows:

1) Build -- Build a conventional domain-referenced test.
2) Printest -- Print a conventional test or an item bank.
3) Score -- Score a conventionally administered test.

The function of Build writes a specification file for a conventional test. The resulting test specification can be used for either computerized or paper-and-pencil administration.

The function of Printest can print single tests or entire banks of items. It supports printing on both dot-matrix and Laser printers. Graphics items are printed pixel by pixel, just as they appear on the screen during computerized administration. Text-only items entered in B/W mode are printed using internal printer fonts and do not use fonts created with the font generator. Please see Appendixes figure 5 for the print out of the bank DEF with 3 items and characteristics of each items.

The function of Score computes number-correct, formula, Bayesian-modal, maximum-likelihood, and EAP scores. The input file should be in the standard MicroCAT data format, and a Parameter file is also required.

6.0 MICROCAT CAPABILITIES

Based on the learning, there are some comments of "good things" and "bed things" about this system. The capabilities of each subsystems are summarized in the section 2.0 to 5.0. Now let us look at the "good things", in general, of the system:

- Special character creation from MakeFont. This function can support local language learning, such as Chinese, Japanese, and some of the special signals.
- Graphics ability provided by the system.
- Several statistics analysis methods can be used to evaluate the test.

7.0 MICROCAT LIMITATIONS

Now, let us look at the "bed things", limitations, in the system from the end user view of the point:

7.1 The Development Subsystem

- During entering or editing test items, there is NO ON SCREEN HELP to guide a user.
- There is NO good support for the text generation and editing.
- The instruction for the Special Character Creation in the item adding is not complete. You can not follow it to go through the process.
- The function Compile does not provide good explanation for the error. The error list can only be seen from the DOS system. It is very inconvenient.
- There is NO WAY OUT, if the compile couldn't go through. There is NO MESSAGE on the screen to show what is the problem and how to get out.
- The function Import can not deal with the graphics,and it requires a very specific format of the input data.

7.2 The Examination Subsystem

- The Testmany function does not work on Window system.

7.3 The Assessment Subsystem

- The function Collect can not work on the test records that result from tests created using the predefined templates, provided by the system, except CALIB. This is really a "pain" in the Assessment process.

7.4 The Conventional Testing Subsystem

- The function Printest does not work all the time. If it is doesn't work, there is No MESSAGE to explain how to fix it and how to get out.

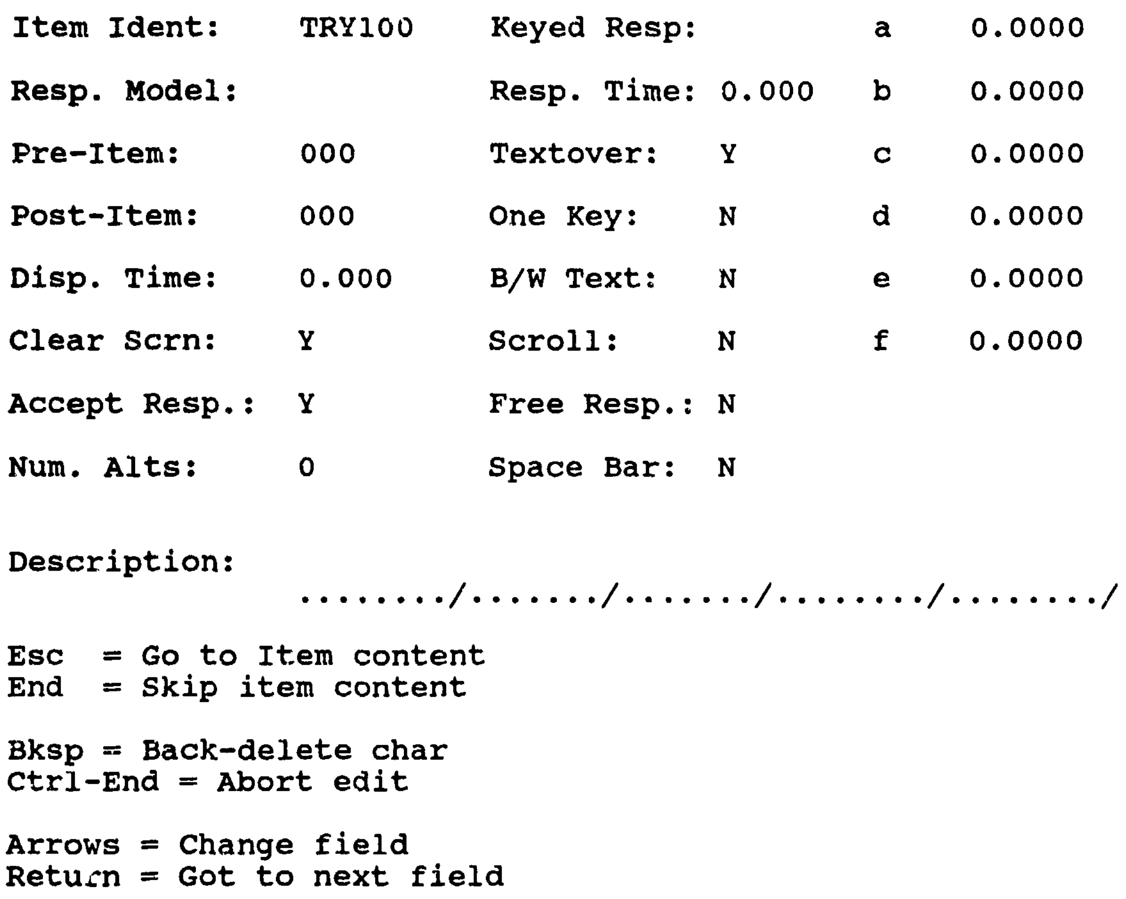
7.5 General

- There is NO ON SCREEN HELP for several processes. When a user don't know how to continue, On screen help is really useful.
- There is NO ERROR TOLERATION. When a user make a mistake, there is no way to make it up.
- There is NO WAY OUT. If a user couldn't give the "right" answer to the system, the user will be stuck there unless reboot the computer. Please see Appendixes Figure 6 for no way out, and Figure 7 for "smart" way to get out.
- There is NO FUNCTION KEY USED to assistant the input. All commands are in the manual. It is hard to remember all of them.
- There is NO CLEAR INSTRUCTION to illustrate how to create a item or a test. If the User Manual includes a example (sample) which go through all the steps from creating a item, a test, to the results of the assessment, it would be a big help.
- Several functions CANNOT be used in the Window system, such as HELP, even it doesn't give you too much help, and Testmany functions.
- There is NO HINT to help a user to answer the system questions, such as the list of the file names available for the input/question.
- There is NO SECURITY management in the system.

APPENDIXES

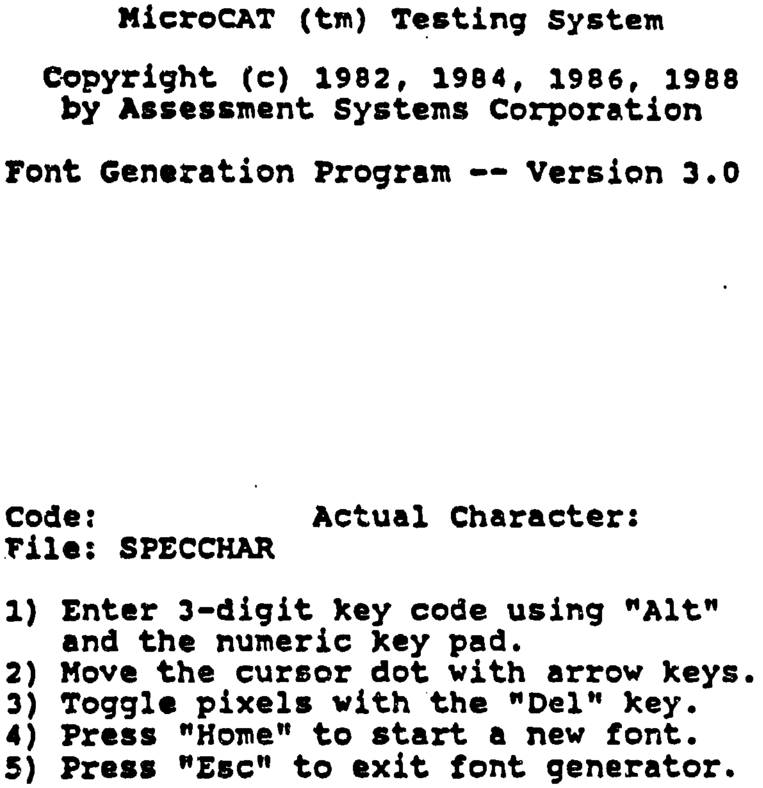
```
What would you like to do?

A = Add a new item to the bank (or write over an old one).

C = Show a catalog of all items in a content area.

D = Show a directory of all content areas.

E = Erase an item from the bank.

I = Install a different character set (SPECCHAR is loaded).

L = Look at an item but do NOT allow any changes.

Q = Quit the edit session.

R = Revise an existing item.

S = Set graphics mode (EGA monochrome).

Enter a letter:
```

Figure 1. Submenu of the Function BANK



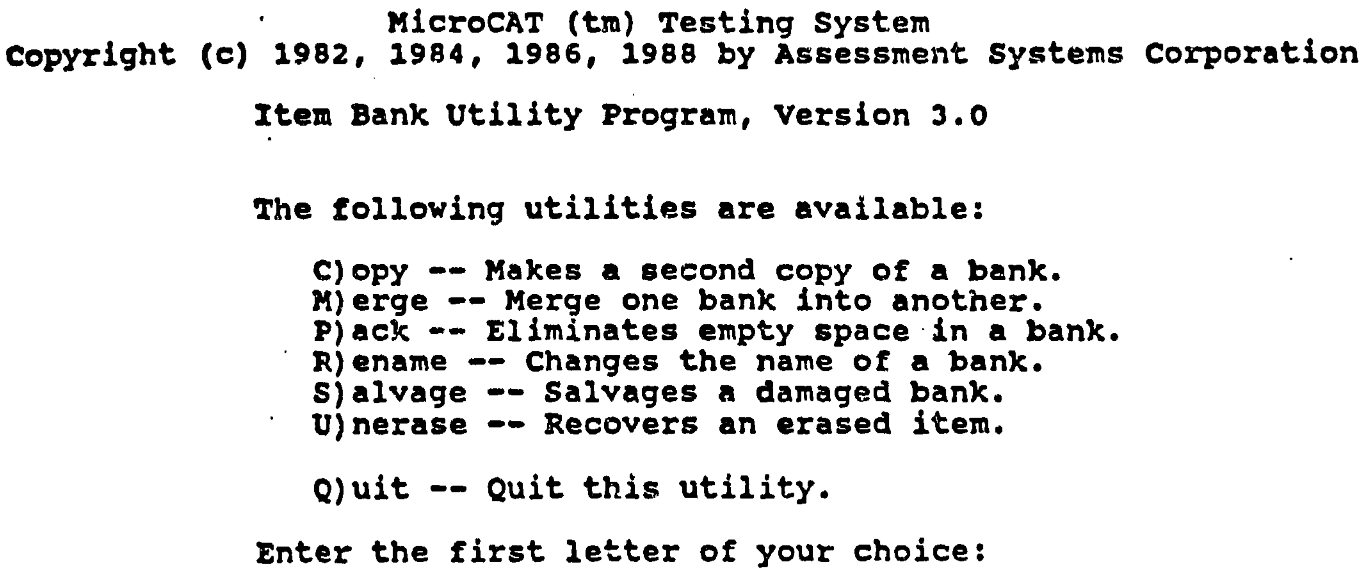
```
Item Ident:    TRY100   Keyed Resp:         a    0.0000

Resp. Model:            Resp. Time: 0.000   b    0.0000

Pre-Item:      000      Textover:   Y       c    0.0000

Post-Item:     000      One Key:    N       d    0.0000

Disp. Time:    0.000    B/W Text:   N       e    0.0000

Clear Scrn:    Y        Scroll:     N       f    0.0000

Accept Resp.:  Y        Free Resp.: N

Num. Alts:     0        Space Bar:  N


Description:
               ......../......./......./......../......../

Esc  = Go to Item content
End  = Skip item content

Bksp = Back-delete char
Ctrl-End = Abort edit

Arrows = Change field
Return = Got to next field
```

Figure 2. The definition of the characteristics of an item.



```
       MicroCAT (tm) Testing System

    Copyright (c) 1982, 1984, 1986, 1988
     by Assessment Systems Corporation

Font Generation Program -- Version 3.0




Code:              Actual Character:
File: SPECCHAR

1) Enter 3-digit key code using "Alt"
   and the numeric key pad.
2) Move the cursor dot with arrow keys.
3) Toggle pixels with the "Del" key.
4) Press "Home" to start a new font.
5) Press "Esc" to exit font generator.
```

Figure 3. Submenu of the Function MAKEFONT



```
                MicroCAT (tm) Testing System
Copyright (c) 1982, 1984, 1986, 1988 by Assessment Systems Corporation

             Item Bank Utility Program, Version 3.0

             The following utilities are available:

                C)opy -- Makes a second copy of a bank.
                M)erge -- Merge one bank into another.
                P)ack -- Eliminates empty space in a bank.
                R)ename -- Changes the name of a bank.
                S)alvage -- Salvages a damaged bank.
                U)nerase -- Recovers an erased item.

                Q)uit -- Quit this utility.

             Enter the first letter of your choice:
```

Figure 4. The Submenu of the Function UTILITY

Figure 5. The print out of the Bank DEF

Page 1

1.

Type your identification number on

the line at the bottom of the screen.

Then press the return key to go on.

```
Item Ident:   def1      Keyed Resp:             a:    0.0000
Resp. Model:            Resp. Time:      0.00   b:    0.0000
Pre-Item:     0         Textover:    Yes        c:    0.0000
Post-Item:    0         One Key:     No         d:    0.0000
Disp. Time:      0.00   B/W Text:    Yes        e:    0.0000
Clear Scrn:   Yes       Scroll:      No         f:    0.0000
Accept Resp:  Yes       Free Resp:   Yes
Num. Alts:    0         Space Bar:   No
Description:
```

2.

Now type your name on the line

at the bottom of this screen.

Press the return key to go on.

```
Item Ident:  def2       Keyed Resp:             a:     0.0000
Resp. Model:            Resp. Time:      0.00   b:     0.0000
Pre-Item:    0          Textover:   Yes         c:     0.0000
Post-Item:   0          One Key:    No          d:     0.0000
Disp. Time:       0.00  B/W Text:   Yes         e:     0.0000
Clear Scrn:  Yes        Scroll:     No          f:     0.0000
Accept Resp: Yes        Free Resp:  Yes
Num. Alts:   0          Space Bar:  No
Description:
```

What is the name of the test you want to take?

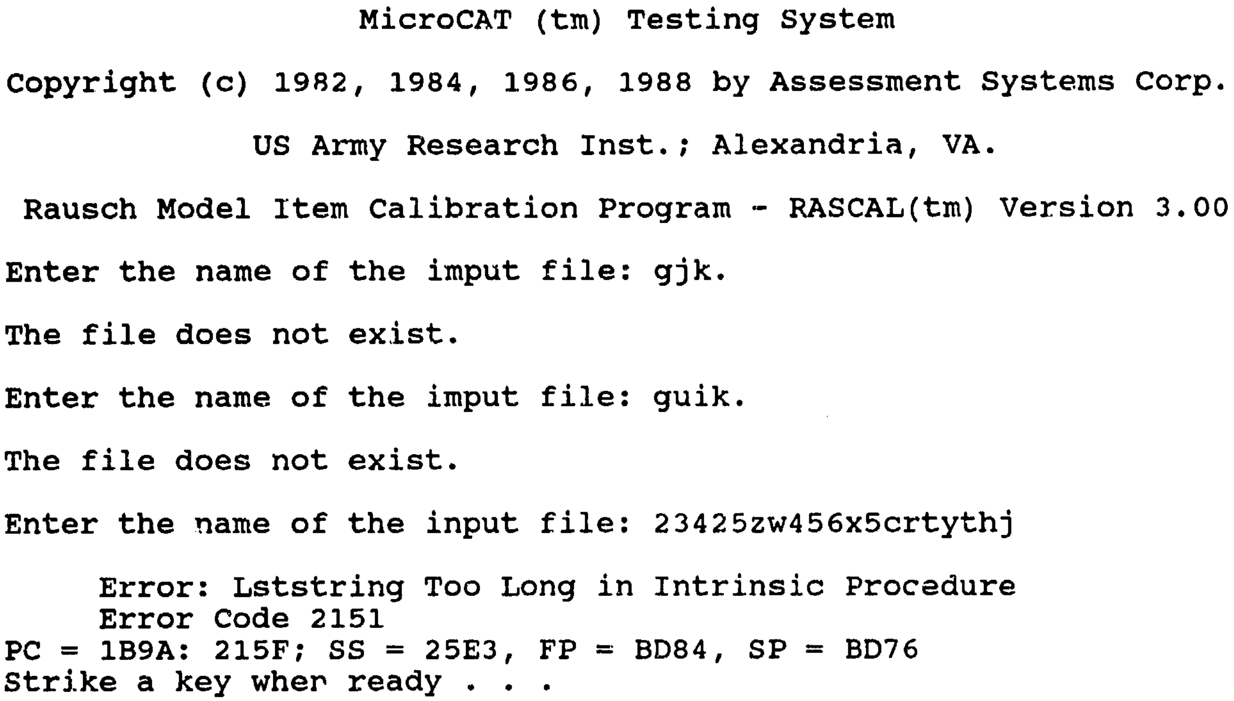
Type the name of the test on the line at the bottom of the screen.

Then press the return key to begin.

```
Item Ident:  def3       Keyed Resp:             a:     0.0000
Resp. Model:            Resp. Time:      0.00   b:     0.0000
Pre-Item:    0          Textover:   Yes         c:     0.0000
Post-Item:   0          One Key:    No          d:     0.0000
Disp. Time:       0.00  B/W Text:   Yes         e:     0.0000
Clear Scrn:  Yes        Scroll:     No          f:     0.0000
Accept Resp: Yes        Free Resp:  Yes
Num. Alts:   0          Space Bar:  No
Description:
```

```
                 US Army Research Institute, Alexandria, VA.
          Item and Test Analysis Program -- ITEMAN (tm) Version 3.00
Enter the name of the input file:  def
The file does not exist.
Enter the name of the input file:  try
The file does not exist.
Enter the name of the input file:
The file does not exist.
Enter the name of the input file:
The file does not exist.
Enter the name of the input file:
The file does not exist.
Enter the name of the input file:
```

Figure 6. No way out.



```
                    MicroCAT (tm) Testing System

Copyright (c) 1982, 1984, 1986, 1988 by Assessment Systems Corp.

              US Army Research Inst.; Alexandria, VA.

 Rausch Model Item Calibration Program - RASCAL(tm) Version 3.00

Enter the name of the imput file: gjk.

The file does not exist.

Enter the name of the imput file: guik.

The file does not exist.

Enter the name of the input file: 23425zw456x5crtythj

     Error: Lststring Too Long in Intrinsic Procedure
     Error Code 2151
PC = 1B9A: 215F; SS = 25E3, FP = BD84, SP = BD76
Strike a key when ready . . .
```

Figure 7. A "Smart" way to get out.

12.0 APPENDIX E
Review of Course of Action

A BRIEF SUMMARY OF THE COURSE OF ACTION

by
Fu Li
Statcom Inc.

for
US Army Research Institute
5001 Eisenhower Avenue
Alexandria, Va 22333-5600

1.0 INTRODUCTION

Course of Action is a software package produced by Authorware. Course of Action provides powerful tools for creating displays. In addition, it can support quickly importing graphics from external sources. Once imported, text and graphics can be refined until they integrate well into your screen design.

Course of Action is an icon-based system. The user can learn how to place icons and structure a course from a Tutorial Lessons, such as creating text using the toolbox, creating graphics using the toolbox, importing graphics and erasing, structuring the question, structuring a student menu, building simulation components, and packaging your course.

2.0 ADVANCED FEATURES

In order to understand the difference between Course of Action and the other authoring system, some of the advanced features are listed as follows:

1) Direct editability

You can make a change or addition to courseware while it is being presented as it would be to a student by just clicking on a screen object. Tools to make changes and additions will appear immediately. You can make changes directly on the screen and then proceed with courseware presentation.

2) Great graphics

You can create graphics directly on the presentation screen or import graphics and captured screens from almost any source. You can mix bit-mapped graphics and object-oriented graphics as freely as you please. Multiple objects can be resized proportionally.

A complete set of display modes assignable to each graphic provides sophisticated presentation capabilities, e.g., normal, transparent, inverse, and erase presentation modes are available.

3) Data-driven animation

This function not only provides movement but can also: scale an object's movement (as is needed for simulating a gauge), proportion the step size (as is needed to complete movement within a specified time period), and coordinate movement of multiple objects (as is needed when several mechanisms must interact with each other).

4) Talented text

Full word-processing power is available for text objects. You can position text anywhere on the screen. You can mix text styles, text fonts, and text sizes as freely as you wish. You can adjust margins, tabs, and decimal tabs as you type. You can center and right justify as desired.

5) Advanced answer analysis

There are nine answer types provided by the system. They are: (1) Text; (2) Click/touch area; (3) Move object; (4) Pulldown menu; (5) Keypress; (6) Pushbutton; (7) Conditional; (8) Time limit; and (9) Tries limit.

You can give hints if the student is taking too long or give bonus points if an answer is especially quick. You also can interrupt students if they seem to be too far off course, or you can allow them to continue along their own paths until they see the consequence of their choices. Then you can back them up to the point where they made a bad decision and let them try again.

6) Concurrency

You can display an animated process while you ask the student about it. You can animate several objects at the same time. You can let a ball rotate when and only when the student drags it across the screen. The system will support many concurrent capabilities through the use of "concurrent" and "perpetual" functions.

The "concurrent" option indicates that you want a function to be performed with the one which follows. So, for example, if you start an animation and then start to play digitized sound, the two functions will be performed together.

The "perpetual" option indicates that you want the current function to be performed along with all of those which follow until you have stopped the function or erased it.

7) Interactive video

It provides important enhancements for both authors and students. You can easily define a segment for the student to study without having to work directly with frame numbers. Using the on-screen controller, you can find the desired segment. By simply clicking a button, the current frame number will be read from the playback machine into the course to mark a start or stop frame.

8) Individualized branching

Individualization is achieved primarily through response-sensitive branching.

There is no limit on the number of anticipated responses you can have active at one time, and because it allows all entry types to be active at the same time, there is extraordinary strength in its ability to be response-sensitive. Separate branching options are available for each answer anticipated.

9) Electronic student notebook

It can be created in any course and allowing the student to capture screen displays for review later. The student provides a title for the display and puts it in the notebook which automatically provides an index of all displays currently being kept. The notebook can keep up to nine displays at a time.

10) Models

It provides a means to share courseware structures among courses and authors and, as a by-product, to replicate structures within a single course. The means for doing this is the creation of "models".

Models contain one or more courseware design icons together with their branching structures. When a model is created, it can be loaded into a copy of Course of Action as an additional, readily-available authoring tool which will be listed under a pulldown menu. Then, using the "Paste model" command, you can insert the model's structure anywhere you specify in your own course design.

11) Variables

The system provides extensive data collection capabilities through "system variables" and "system functions". These capabilities can retrieve any student answer, provide extensions to the already extensive text analysis capabilities, and plot data on the screen.

In addition, you can create as many "user variables" as you need for keeping additional data or performing calculations.

12) Computer-managed instruction

The system provides the capabilities not only to build Computer-managed instruction but also to directly activate the appropriate courseware.

It doesn't matter if the courseware you call on is written using the system (Course of Action), programmed in a programming language, or provided as a complete package. If it is an application which runs on your presentation hardware, the system can jump to it. When the student quits using the software jumped to, the system will resume exactly where it left off.

13) External interface

The system can easily load routines written in various languages (e.g., C and Pascal). You can pass data to these routines and receive output. It is even possible for external routines to post text and graphics to the course presentation window. Your course can create external data files during presentation and write any data into them and read data from them as you wish.

14) Portability

Authorware currently has the ability to present courseware (written on the Macintosh) on both Macintosh computers and IBM/IBM-compatible microcomputers. Conversion is automatic and very effective. It even converts fonts!

In the future, you can expect Authorware to support more machines for both authoring and courseware presentation with close to 100% portability of courseware.

15) Local languages

The system has a Japanese language version in operation, though at introduction the products are in English. Chinese versions are also under development, and many other versions are anticipated.

16) Automatic documentation

The system provides many options for printing. It automatically annotates your design by graphically printing the course flow and describing all of the options you have chosen for each icon. Graphics contributed to the presentation window by each icon can be shown in full size while their place on the screen is shown in a thumbnail display.

3.0 GENERAL STEPS FOR THE COURSEWARE DESIGN AND DEVELOPMENT

In order to understand the basic process of building a courseware, the general steps for its design and development are summarized as follows:

1) To structure a course title display

2) To create text using the toolbox

3) To create graphics using the toolbox

4) To add icons (There are eight basic design icons provided for the basic functions: display, animation, erase, time/pause, branch, question, calculate, and organize. Optional icons provide additional specialized functions such as digitized sound and video interface.)

5) To import graphics and to erase

6) To structure the question

7) To fill in the question

8) To provide feedback

9) To complete the question

10) To extending the question

11) To structure a text answer question

12) To structure a student menu

13) To transport course segments

14) To fill in a student menu

15) To complete the student menu

16) To build simulation components

17) To activate simulation components

18) To integrate simulation components

4.0 THE BASIC FUNCTIONS OF ICONS

Icons play very important roles in the course design. The brief definition of the basic function for each icon are listed as follows:

- Display icons — To put text and/or graphics on the screen for the student.
- Animation icons — To move the object(s) of a preceding display icon from one point to another in a given amount of time or at a specified speed.
- Erase icons — To erase display objects.
- Wait icons — To interrupt course flow until 1) the student presses a key or clicks the mouse or 2) a specified amount of time elapses.
- Decision icons — To select which icon(s) from a set of attached icons to use next.
- Question icons — To present a question for the student to answer and then, based on the student's answer, select and branch to attached icons for feedback.
- Calculation icons — To perform arithmetic or special control functions, execute user-written code, jump to other course files, or jump to other applications.
- Map icons — To organize and modularize the course by providing space to put more icons. Each Map icon provides its own course flow line on which you can place other icons, including Map icons.
- Start and Stop flags — To appear below the palette of basic icons. These flags can be placed above, below, or between icons to allow you to run and edit segments of your course design.

5.0 THE LIMITATIONS OF THE SYSTEM

Based on the learning from the tutorial lessons, there are few comments listed as follows to represent problems in the system:

1) The system cannot automatically save the changes.

2) It requires too much steps to complete the building of a courseware.

3) It is hard to remember all the steps in order to build a courseware without a clear instruction and on screen help.

4) There is no explicit way to modify/change the contents of a courseware.

5) If the situation shown on the screen is different from the Tutorial Book, the user has no way to get HELP.

6) Some of functions or processes used to build a courseware are not meaningful to the user. The explanation of the logic relations among these steps is needed.

7) The system requires too much memory space. There are some functions couldn't perform due to the space problem.

8) There is no any assistant tool to collect and analysis answers from a courseware.

9) There is no security management.

Even the system has these problems, it is still an excellent tool to produce a courseware, especially for a graphics part. Some of the problems can be solved by more practice.

Please see Appendixes for more information on how to build a courseware using the Course of Action.

APPENDIXES

Some screen printouts on an example of a courseware
created by the Course of Action
with a short description for each screens (Figures).

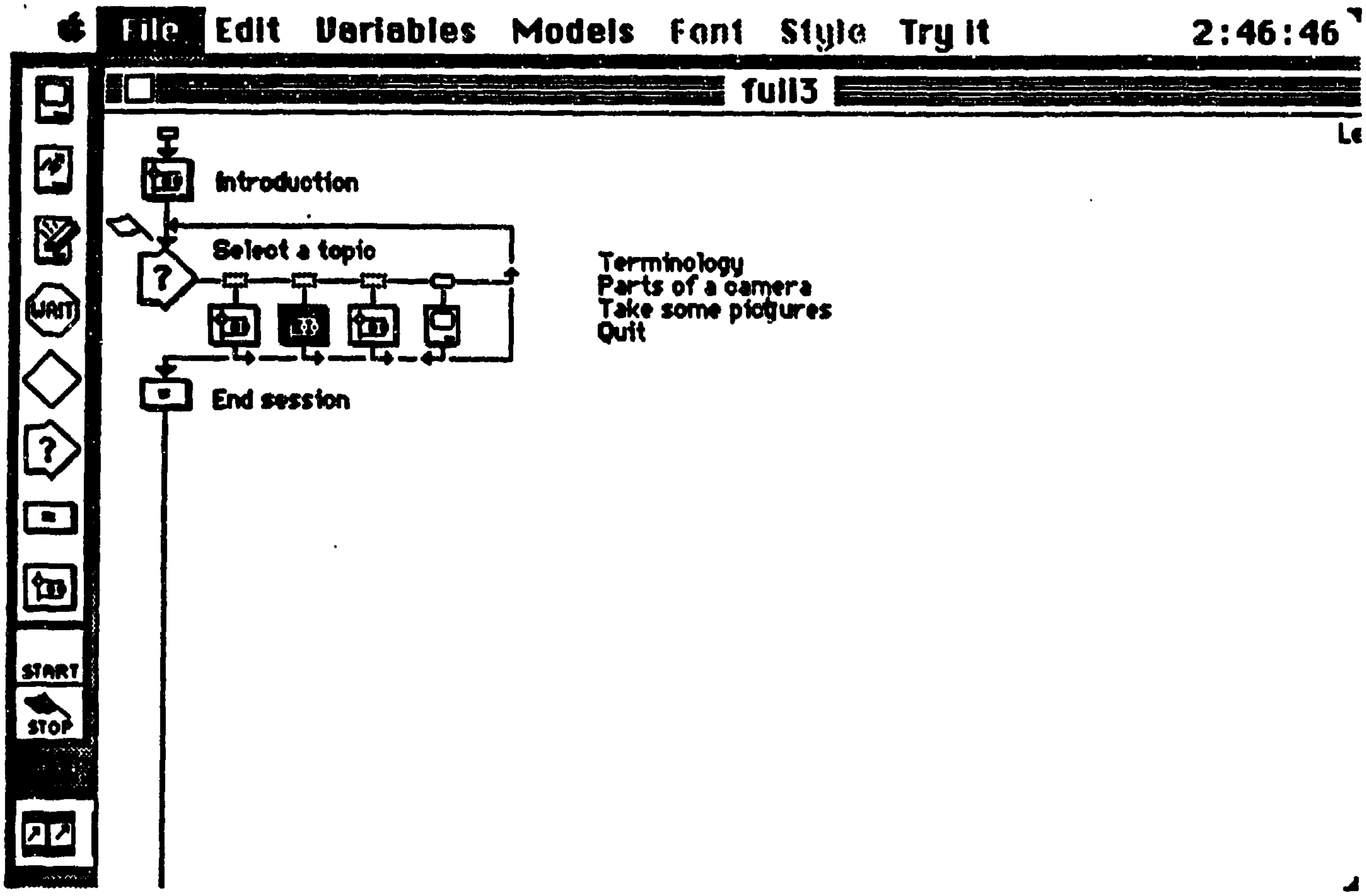


Figure 1. The icon overview of the courseware named fuli3

The introduction part includes two pictures shown on Figure 2 and Figure 3.

The main part of this courseware is to introduce the basic concepts of a camera.

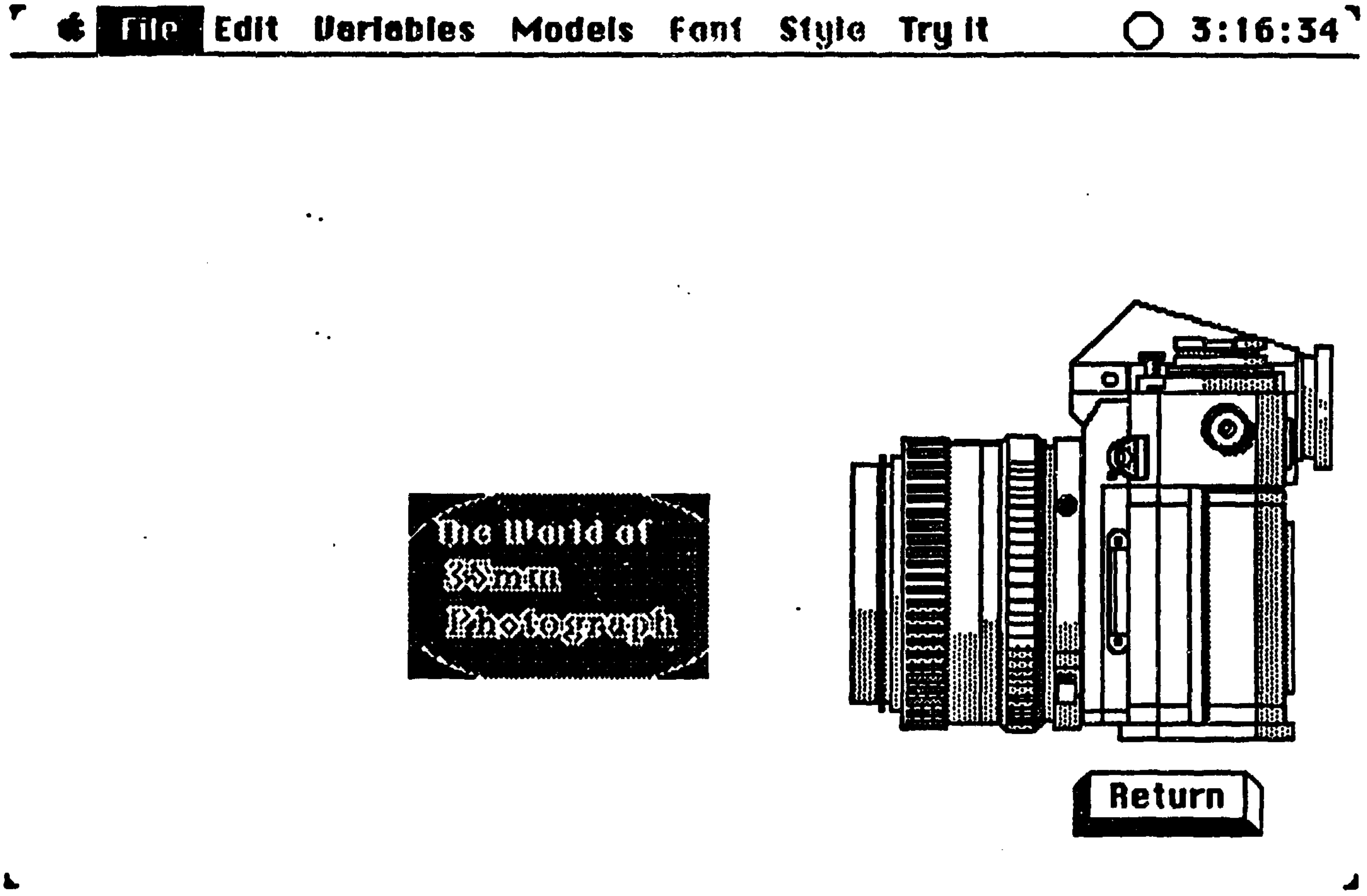
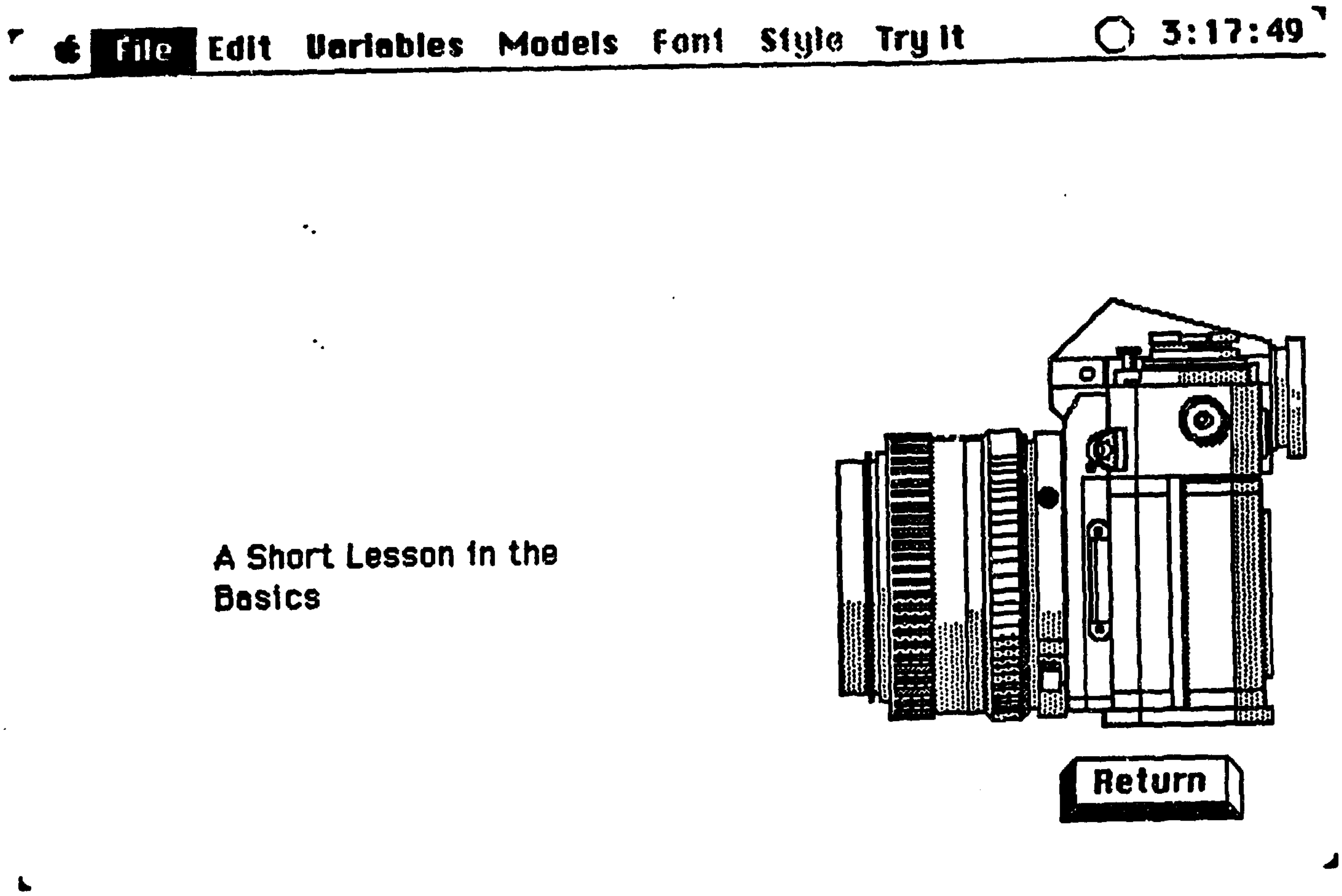
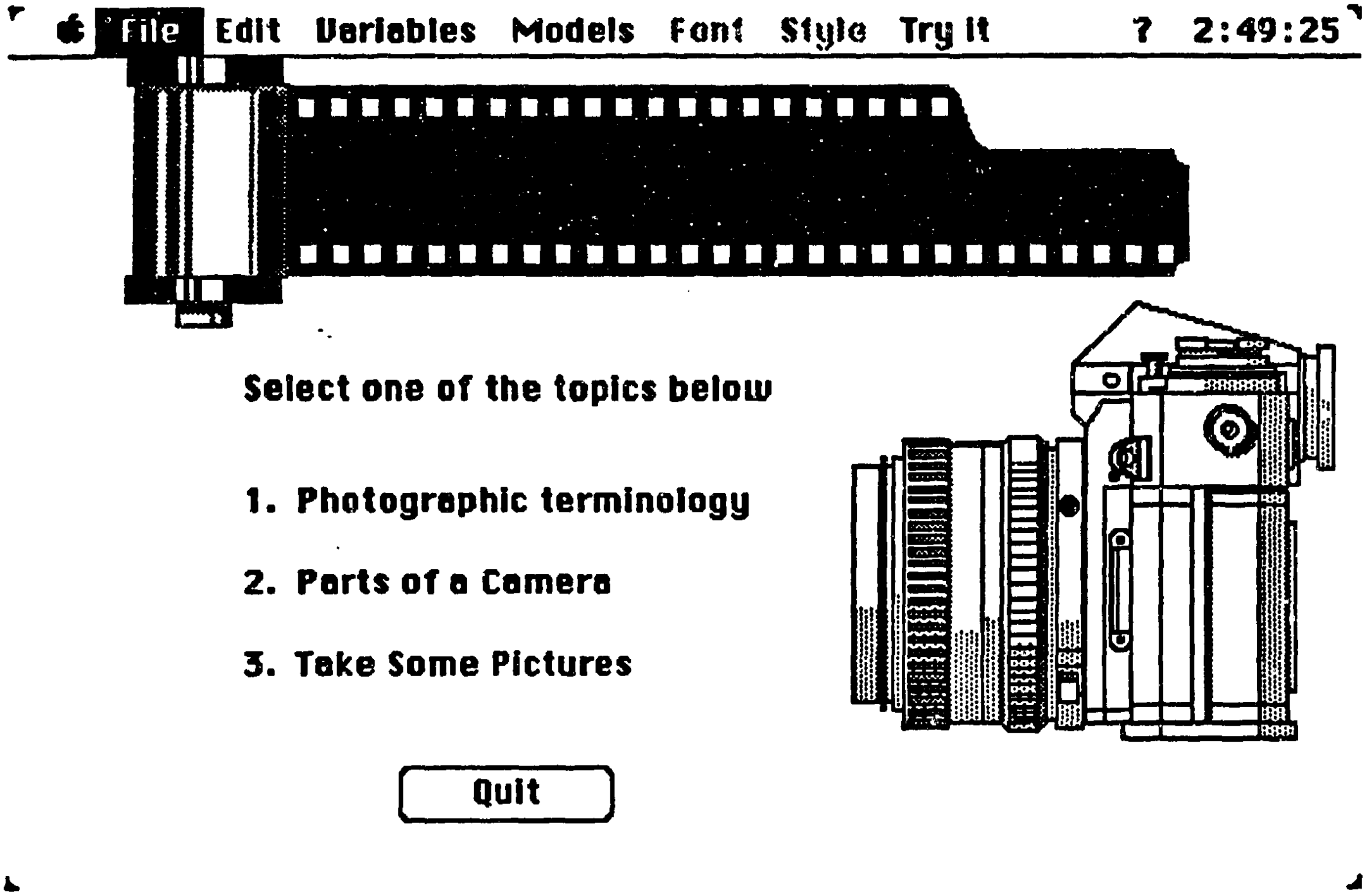


Figure 2. The first picture shows on the screen from INTRODUCTION part when you click on RUN from TRY IT menu (at the top of each screen).



A Short Lesson in the Basics

Figure 3. The second picture shows on the screen after you click on RETURN button at the right corner from the first picture.



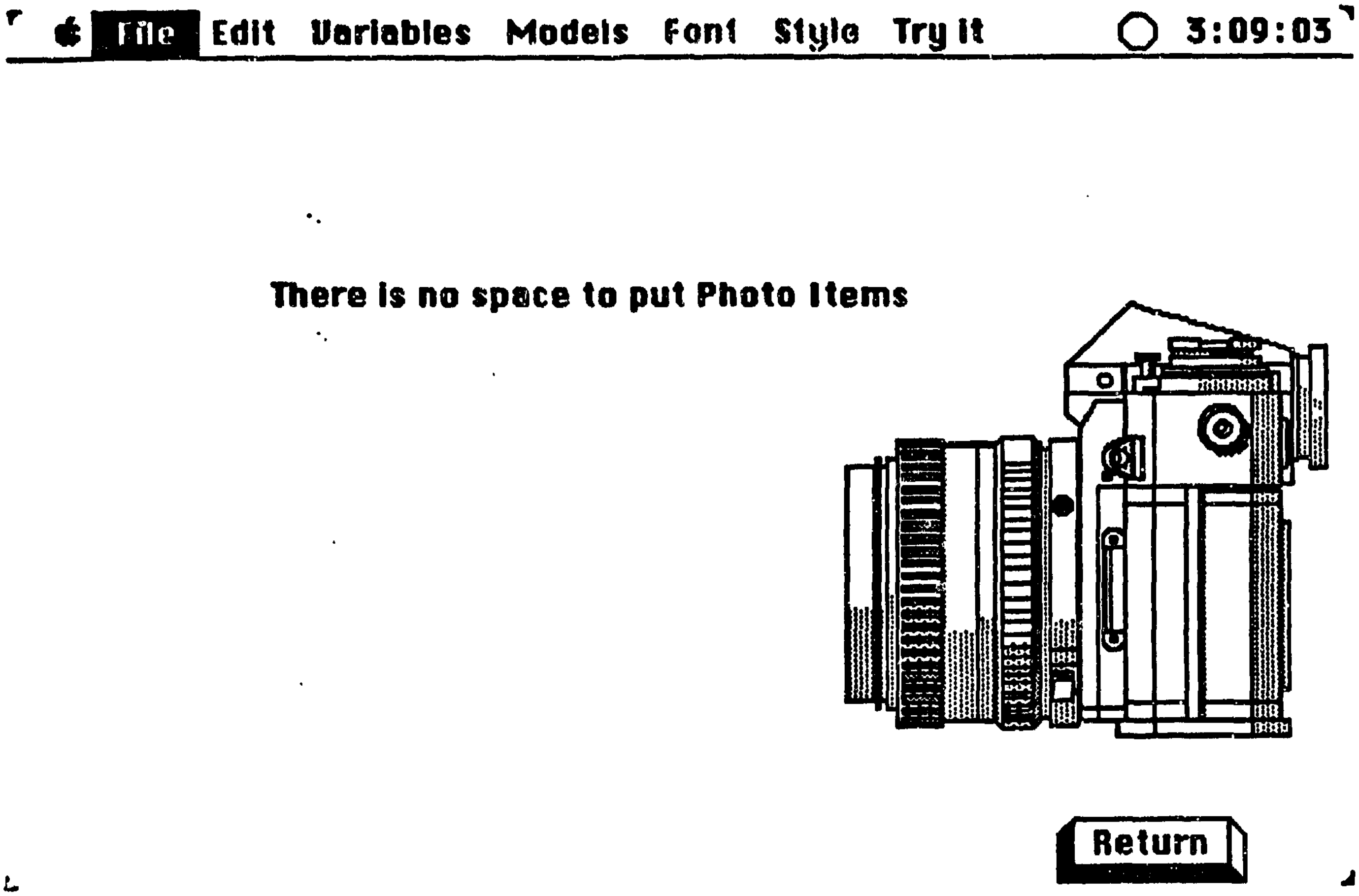
Select one of the topics below

1. Photographic terminology

2. Parts of a Camera

3. Take Some Pictures

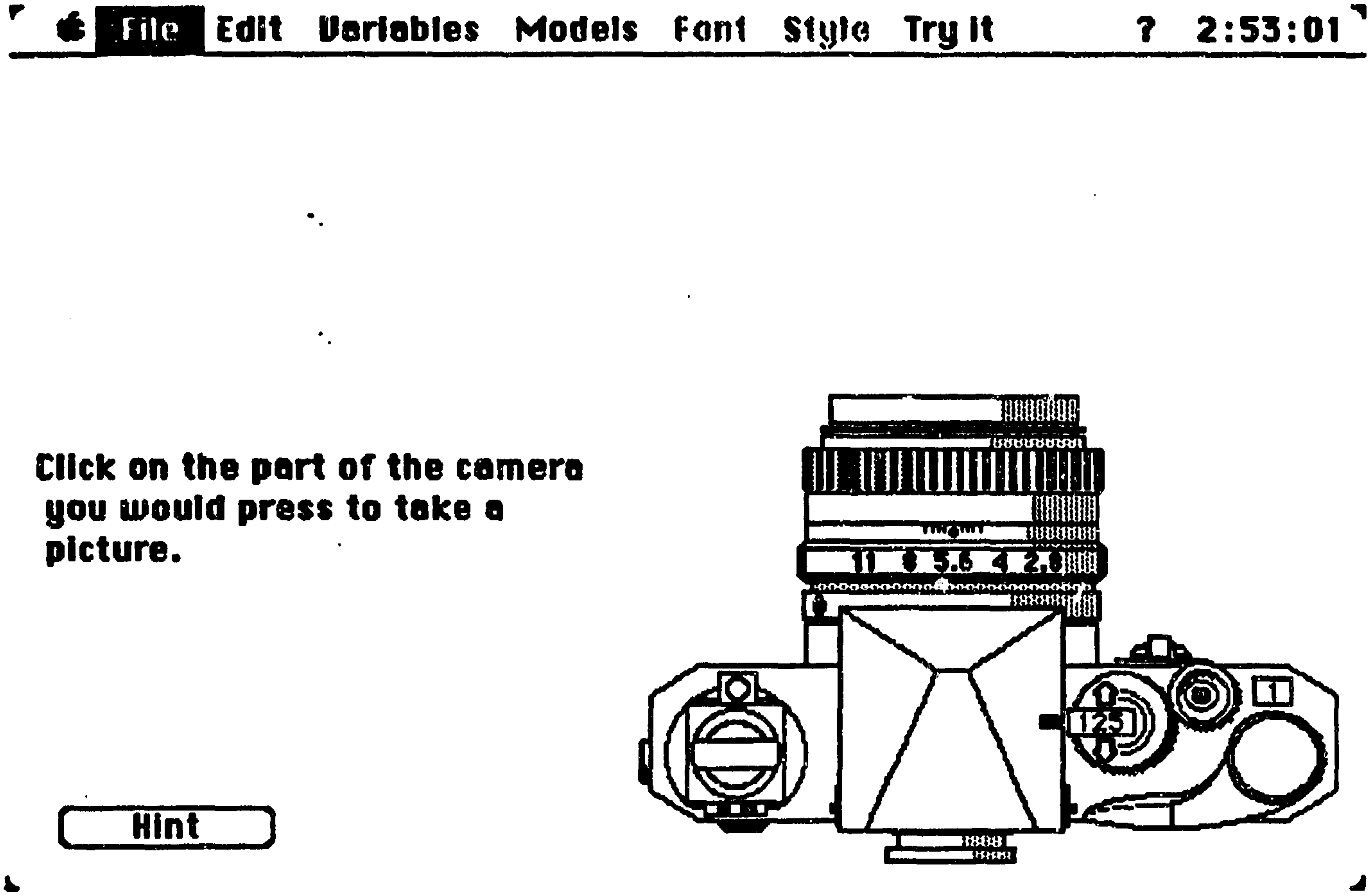
Figure 4. This is a main menu shown on the screen to ask a student to choose a topic after you click on RETURN button from the screen before.



There is no space to put Photo Items

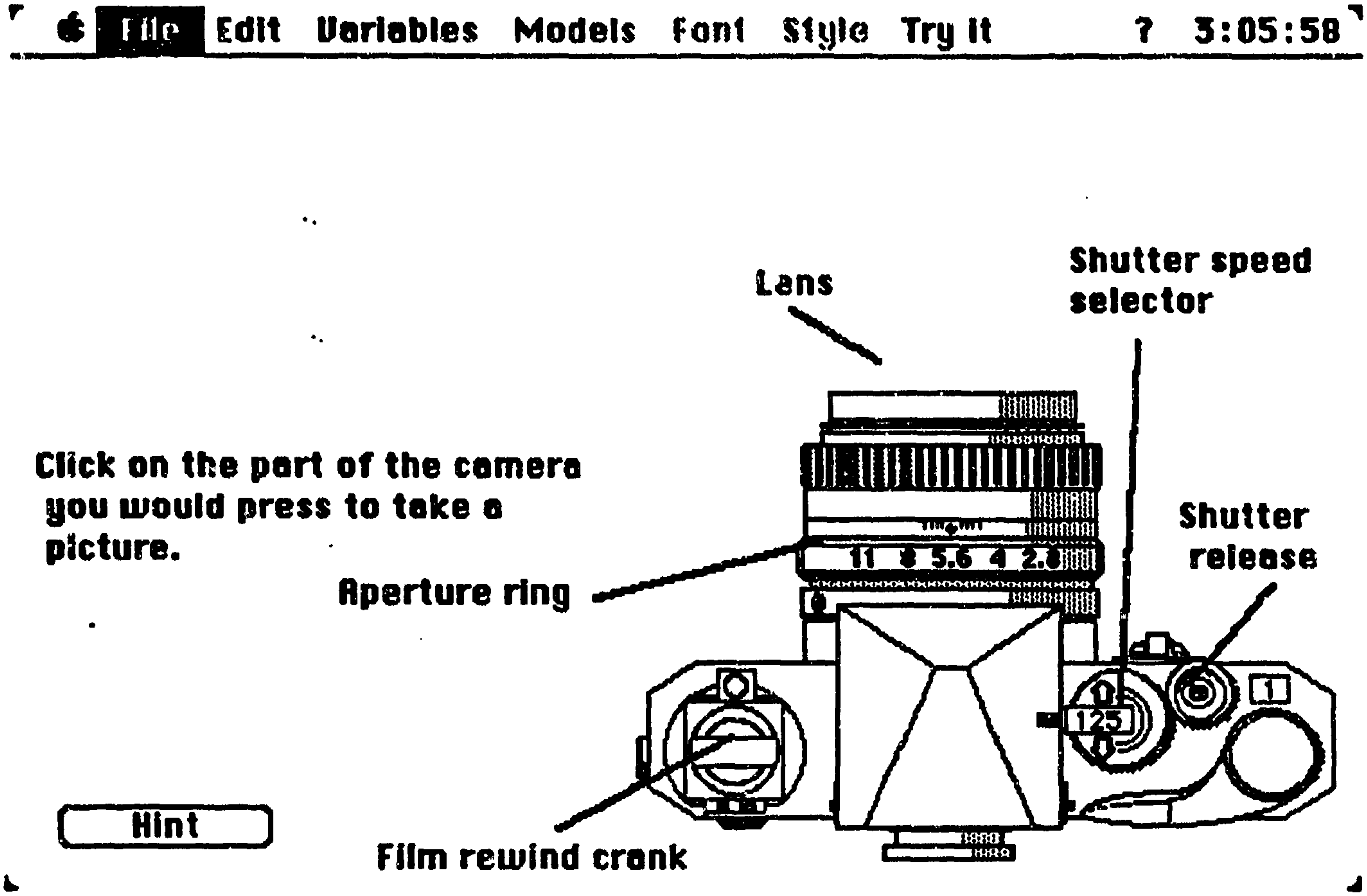
Figure 5. This is the picture shown on the screen after you choose the first choice from the main menu by click on the phase or by type number 1.

When I design this part of the courseware, it needs to import a file named Photo Items to the courseware. But there is no enough space to do it.



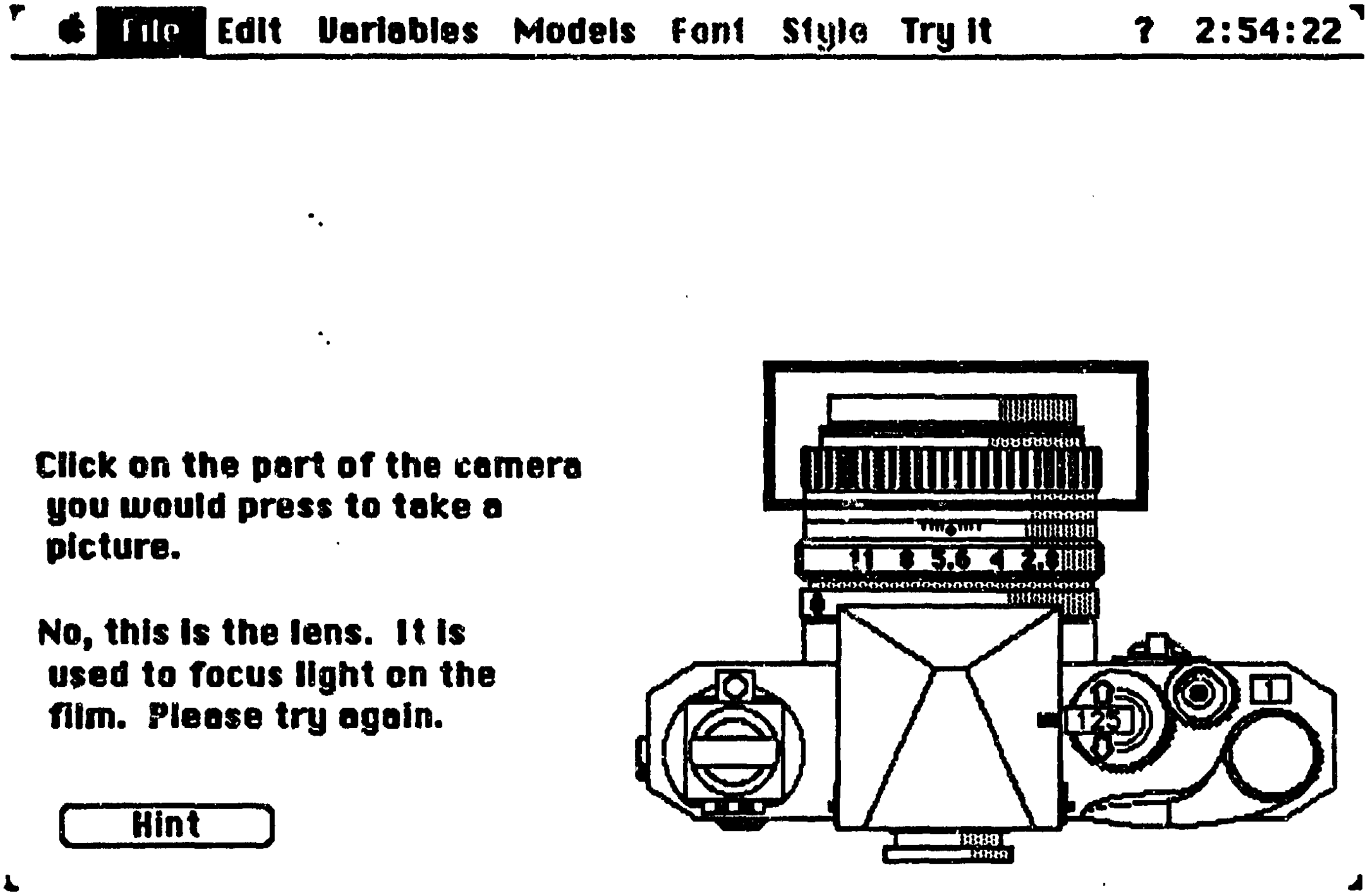
Click on the part of the camera you would press to take a picture.

Figure 6. This is the first picture shown on the screen when you choose the second choice (Parts of a Camera) from the main menu.



Click on the part of the camera you would press to take a picture.

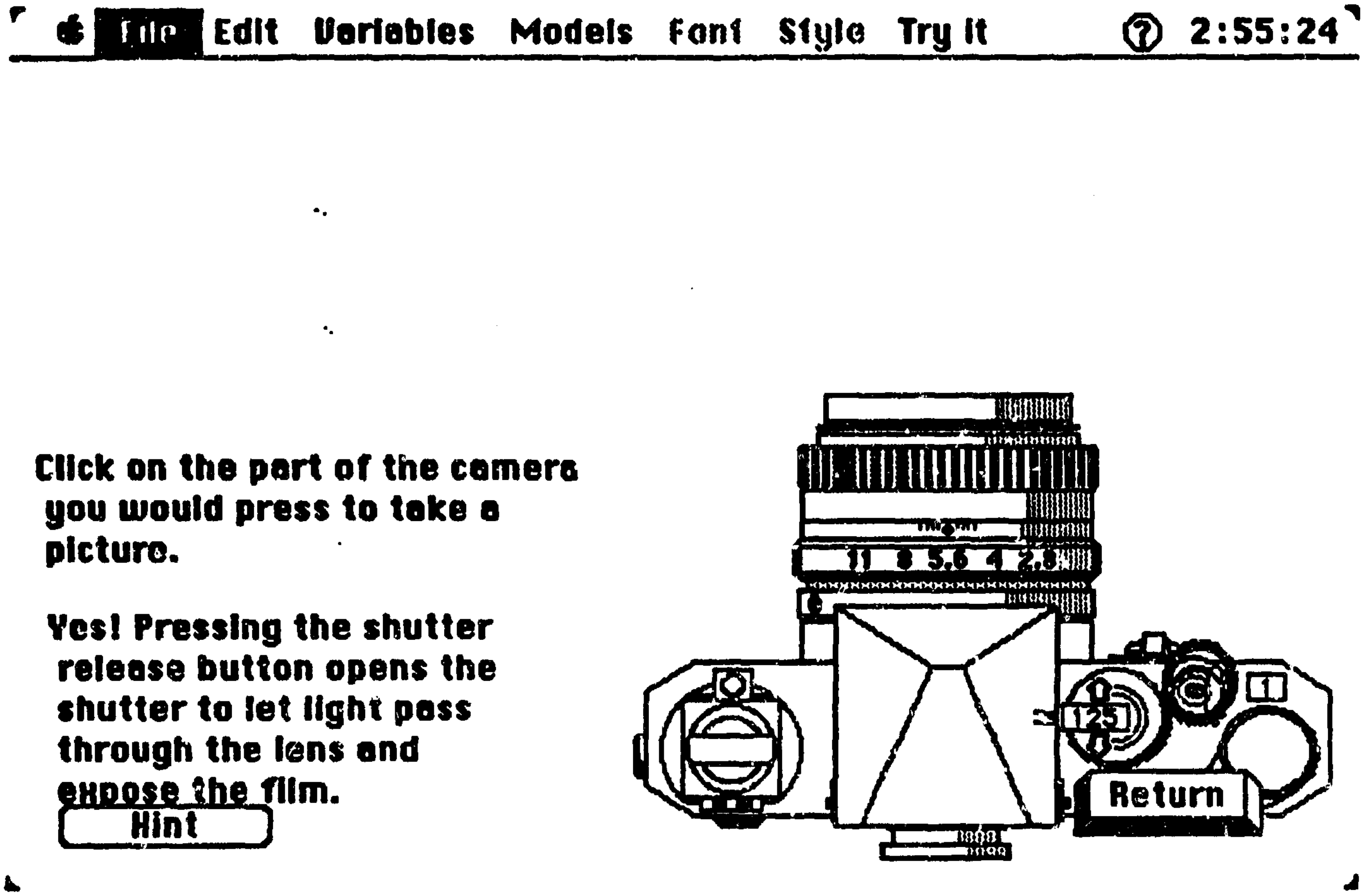
Figure 7. This picture shows how HINT button works. When you click on HINT button, several items will appear on the screen to point out parts of the camera in order to help you to answer the question.



Click on the part of the camera you would press to take a picture.

No, this is the lens. It is used to focus light on the film. Please try again.

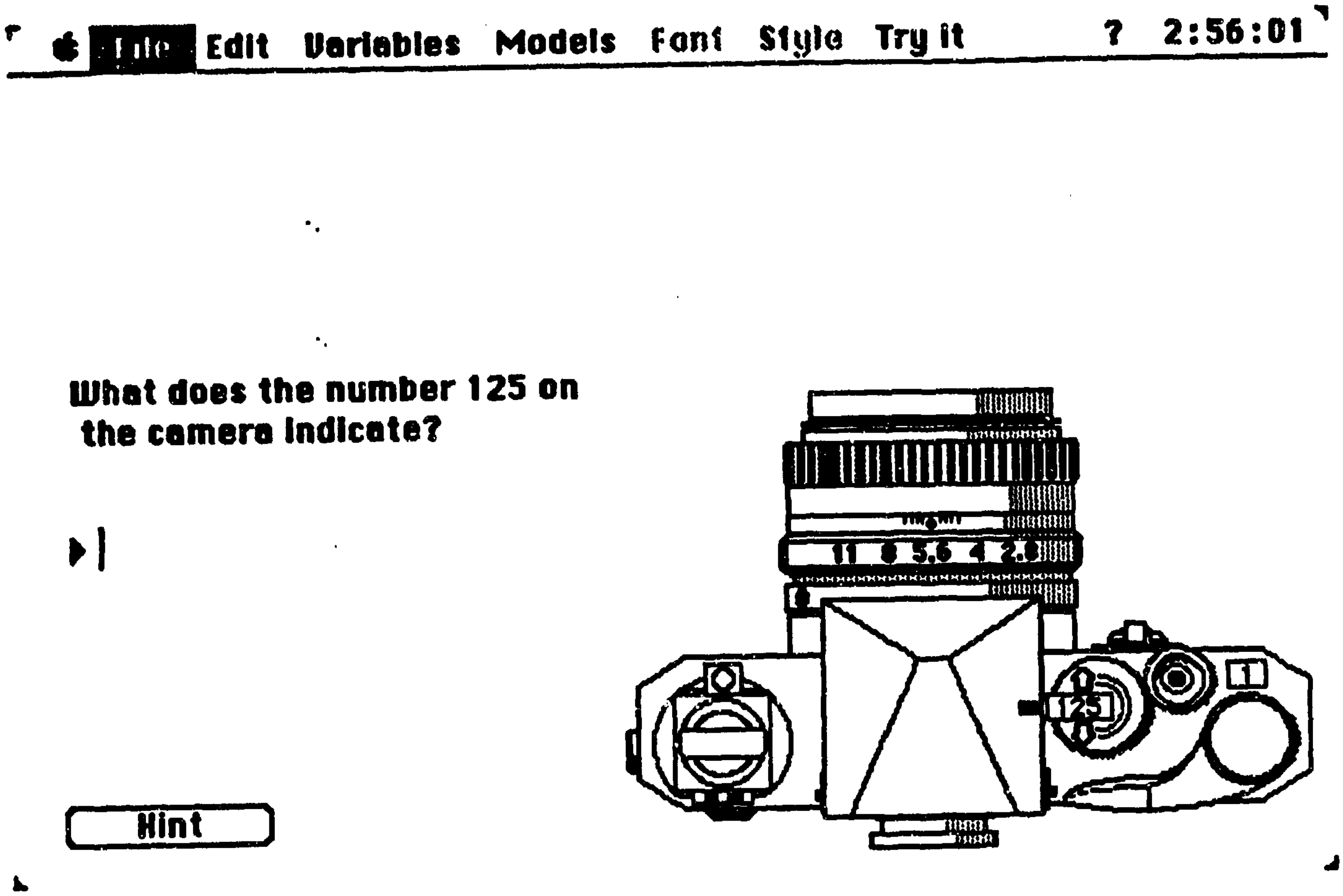
Figure 8. After you click on the part Lens, which is a wrong answer, a short comments will show on the screen to tell you why it is a wrong answer.



Click on the part of the camera you would press to take a picture.

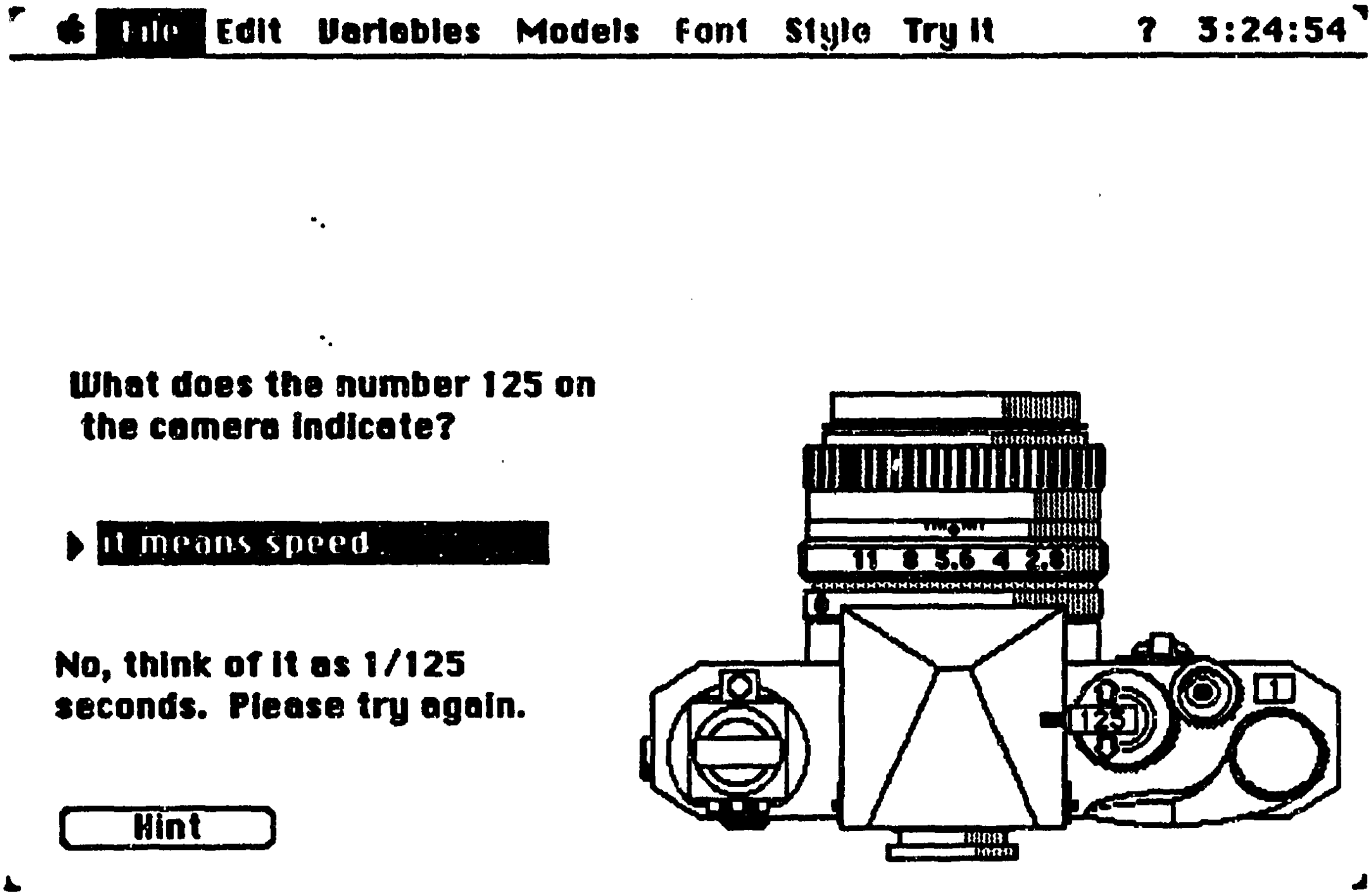
Yes! Pressing the shutter release button opens the shutter to let light pass through the lens and expose the film.

Figure 9. Congratulations! You are right this time.



What does the number 125 on the camera indicate?

Figure 1C. This is the second question from Part of a Camera. It is designed as a limited tries (two times).

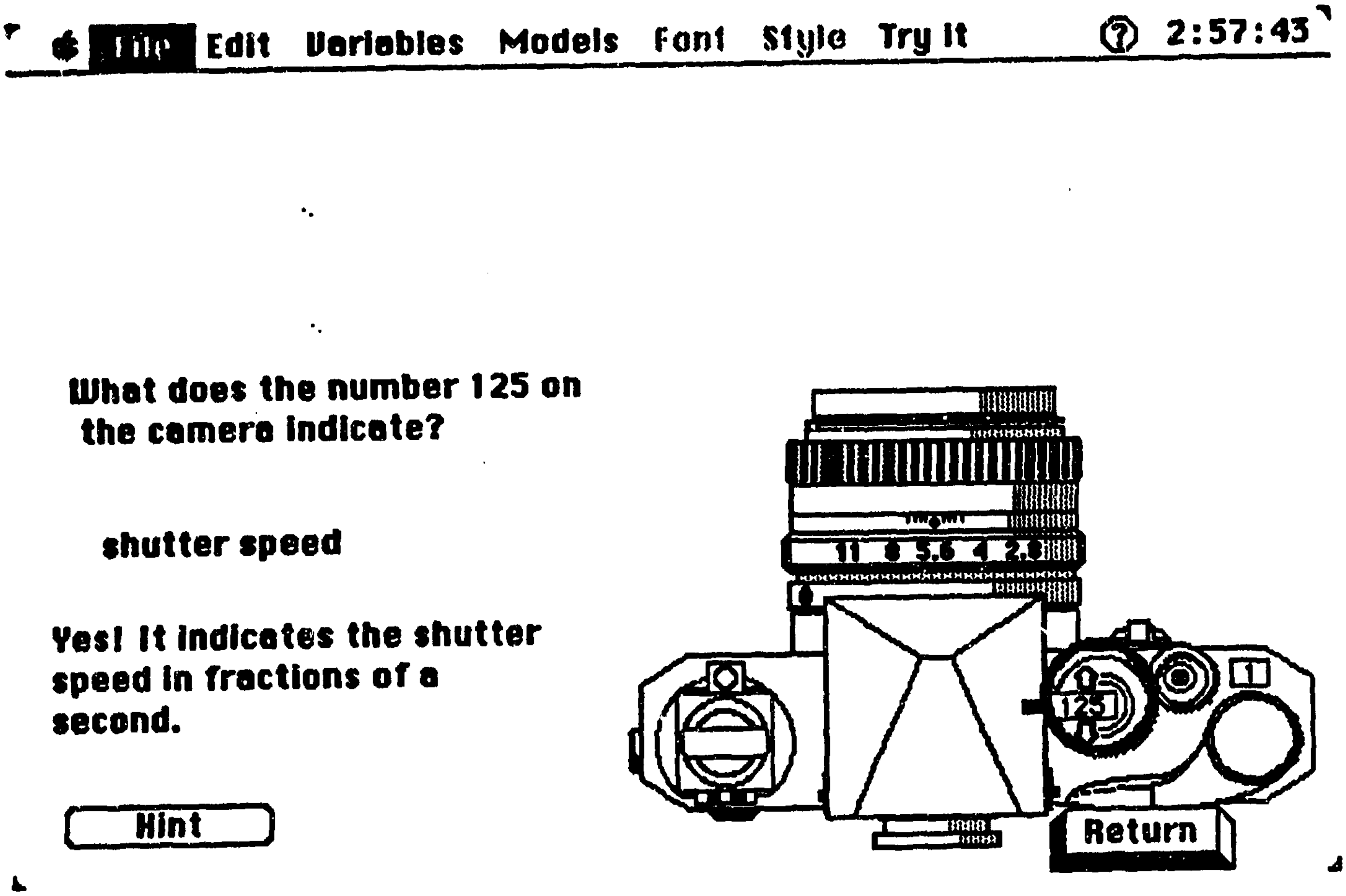


What does the number 125 on the camera indicate?

▶ it means speed.

No, think of it as 1/125 seconds. Please try again.

Figure 11. The first try is not right.

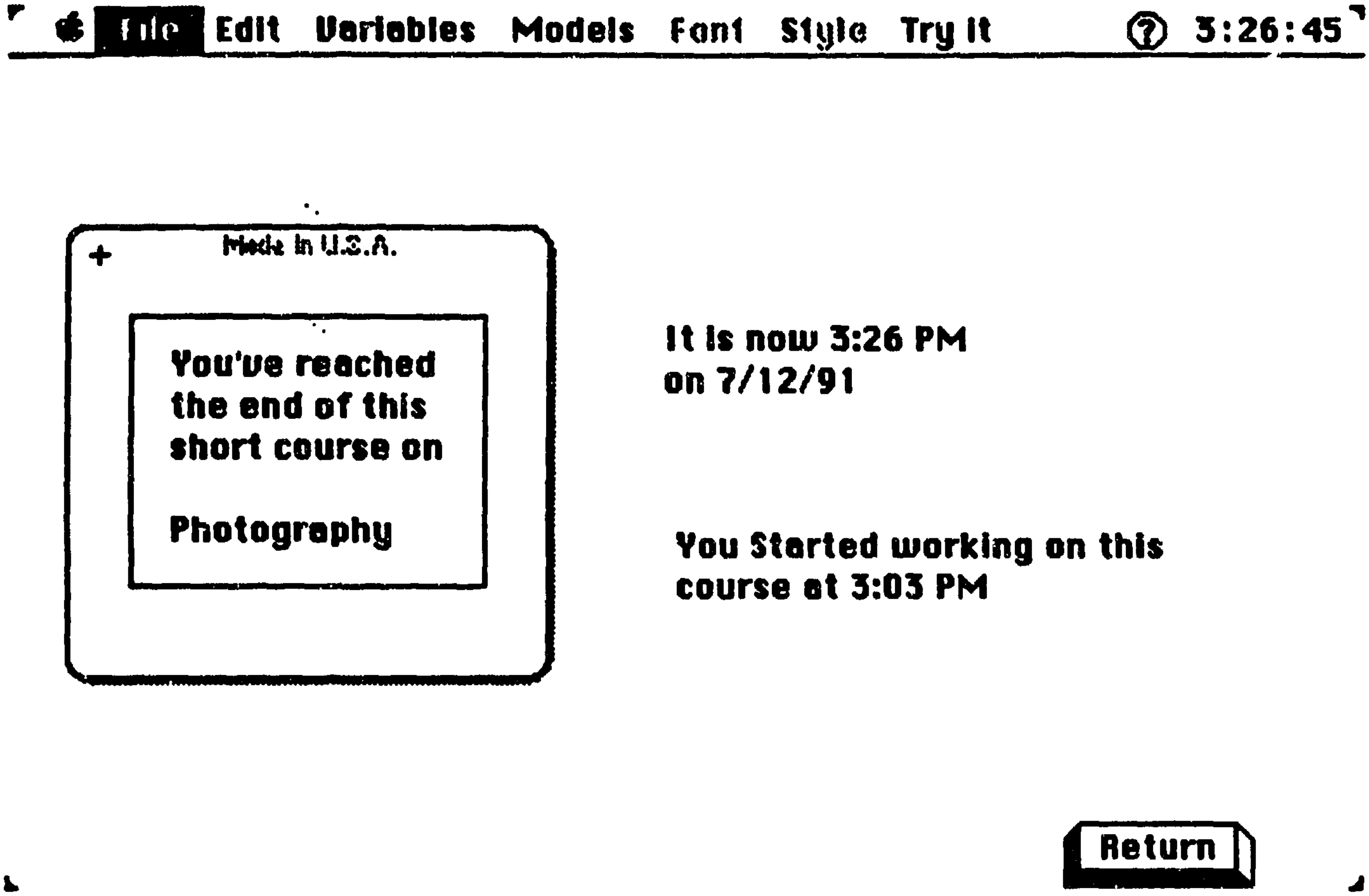


What does the number 125 on the camera indicate?

shutter speed

Yes! It indicates the shutter speed in fractions of a second.

Figure 12. Congratulations! You did it again!



You've reached the end of this short course on

Photography

It is now 3:26 PM on 7/12/91

You Started working on this course at 3:03 PM

Figure 13. This is the last picture shown on the screen after you click on the QUIT button. It shows you time and date by using the system variables and the system functions.

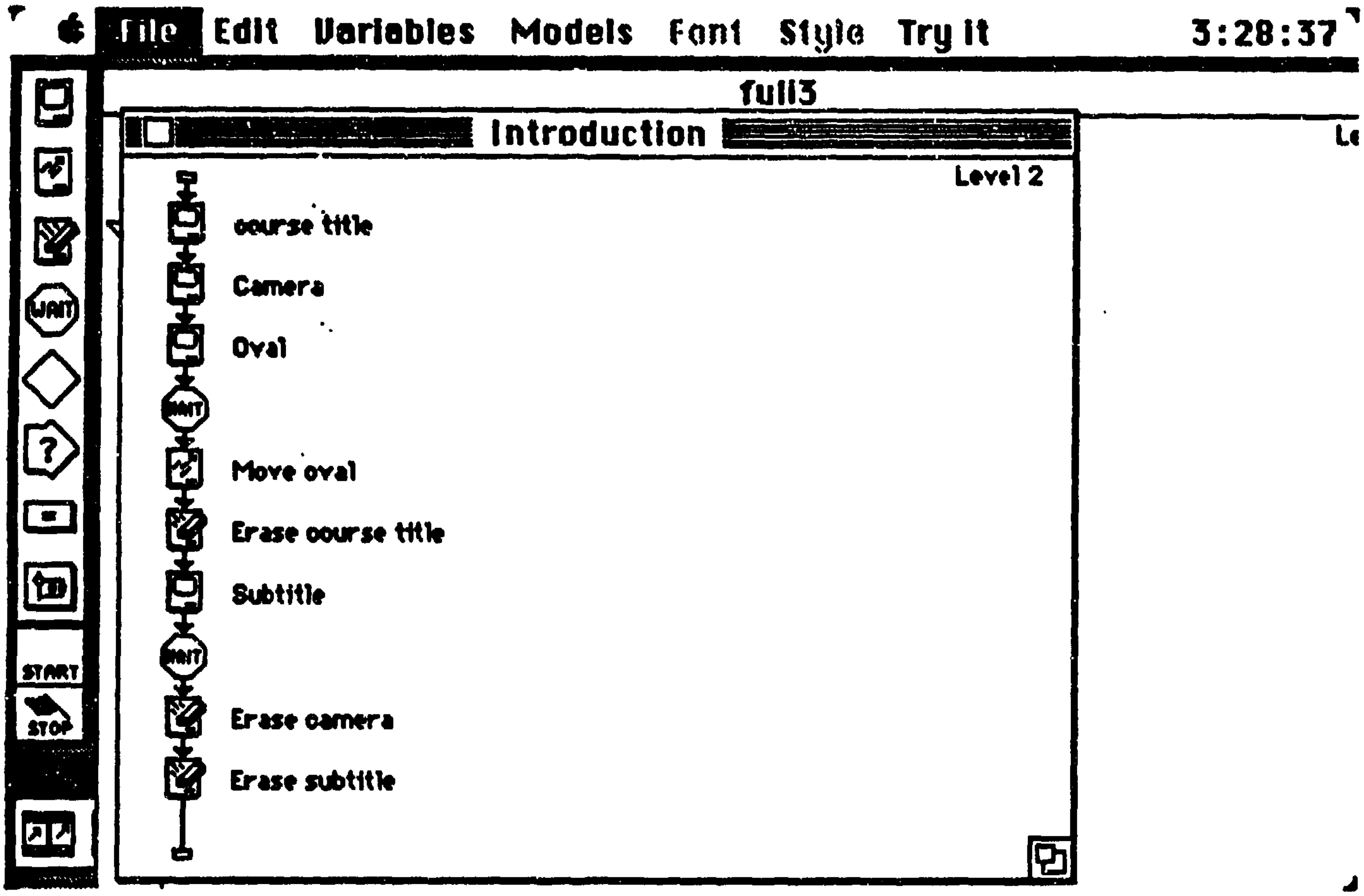


Figure 14. This is the picture that shows how the Introduction part designed by icons.

You can think it as a icon flow diagram. The basic design ideas are really like programming by icons.

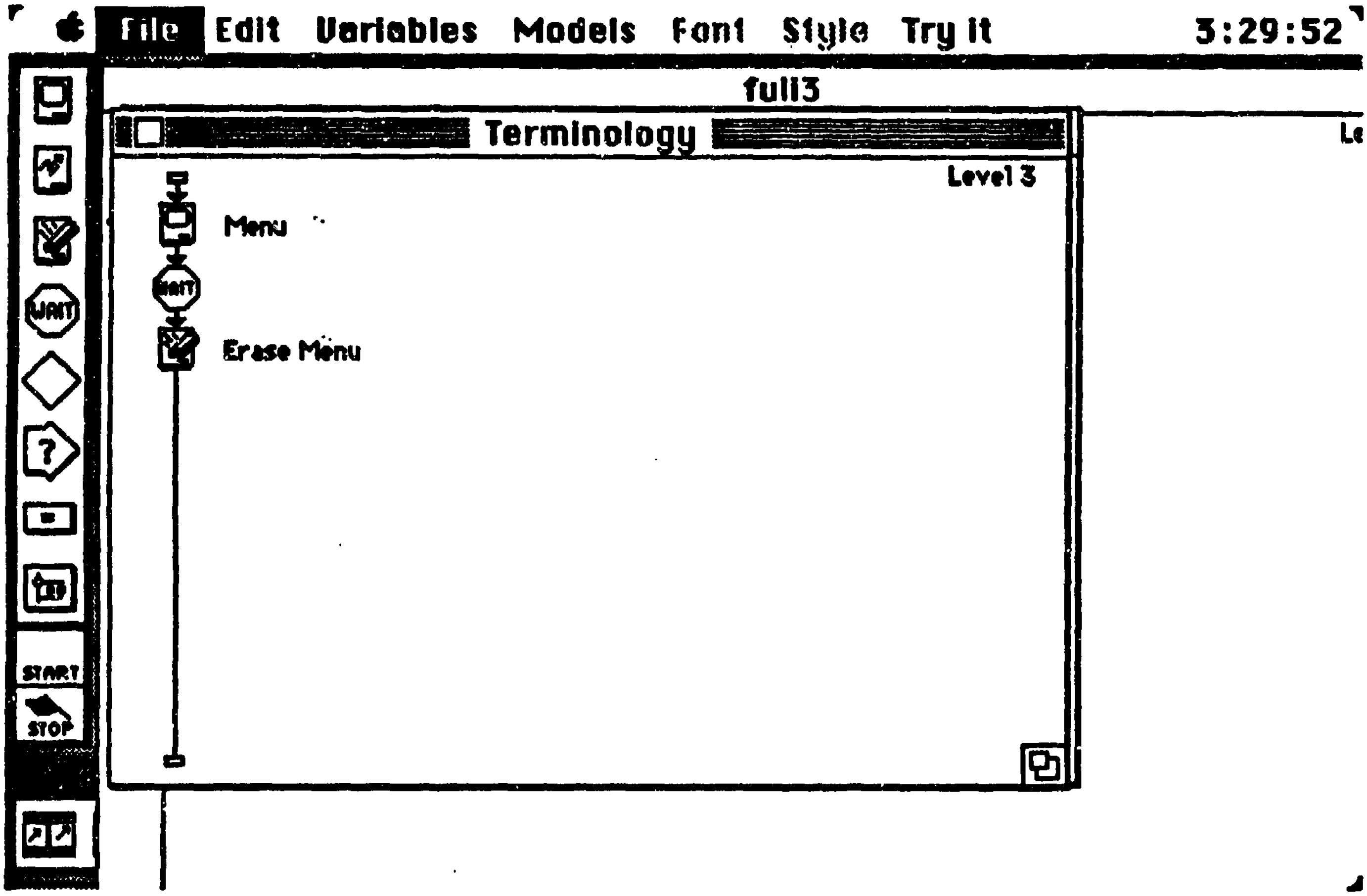


Figure 15. This is the icon flow diagram for the Terminology part (the first choice from the main menu).

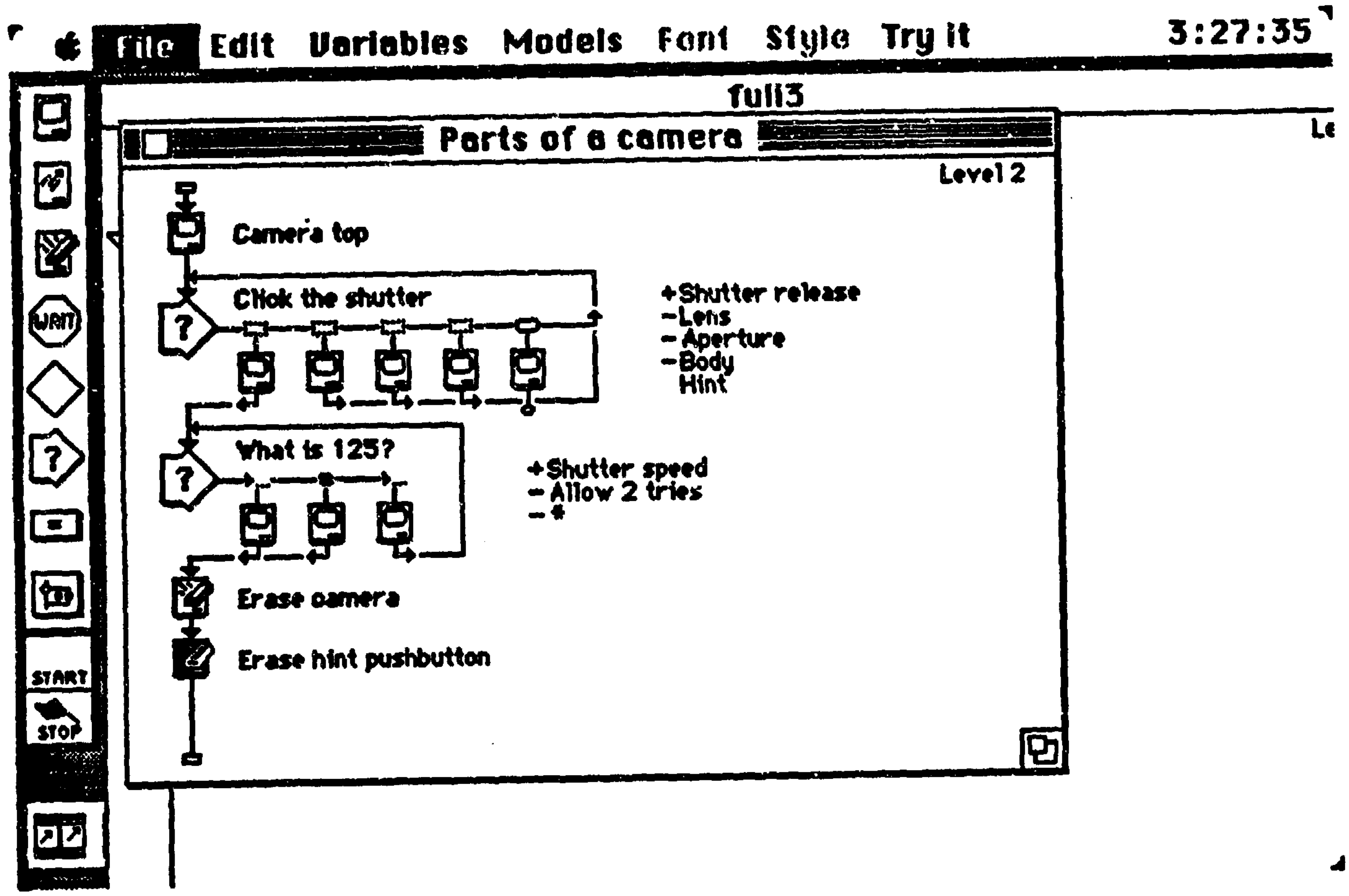
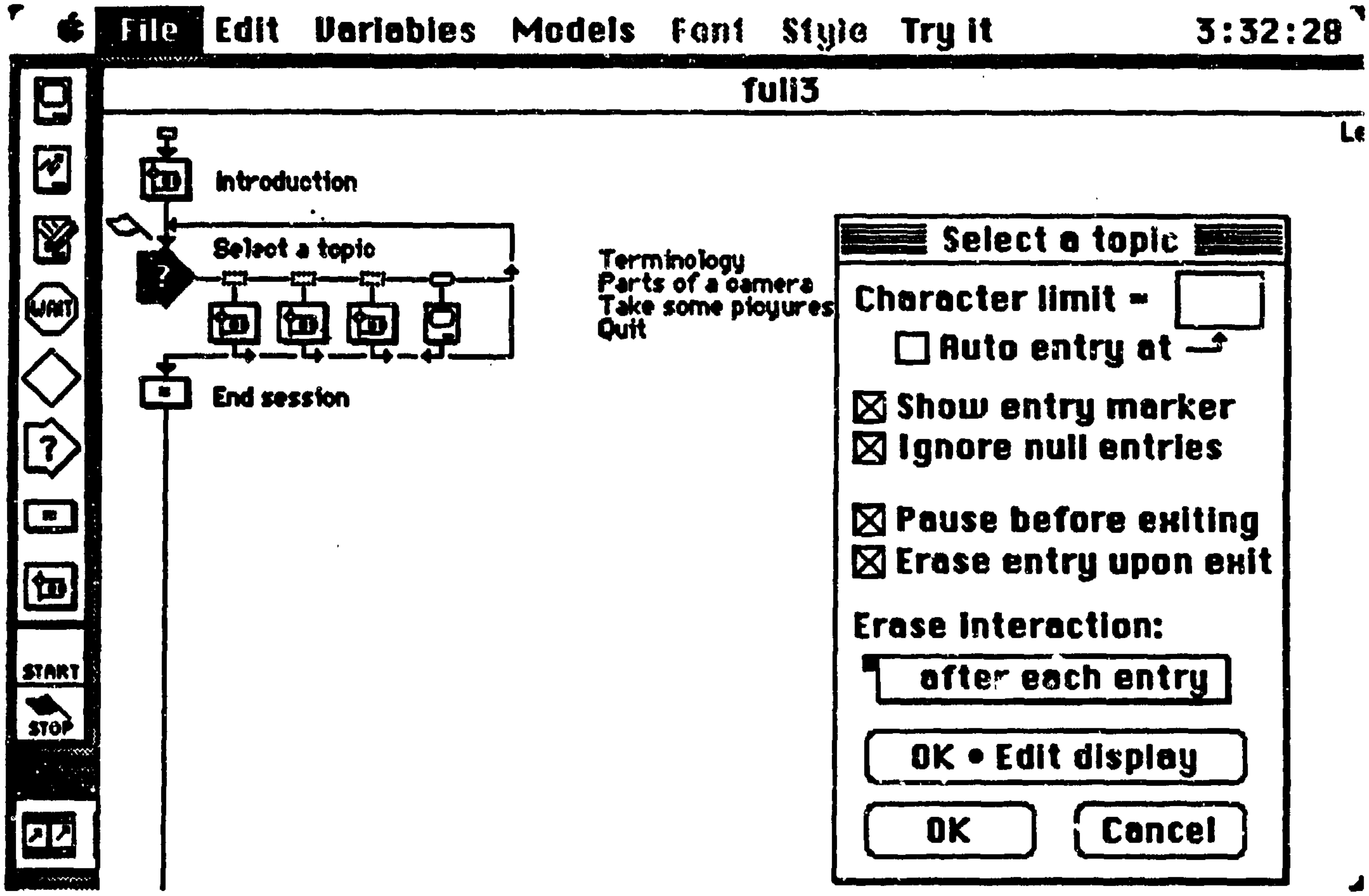


Figure 16. This is the icon flow diagram for the second choice from the main menu (Part of a Camera).

From the diagram, you can see how branch works.

You need to define the wrong answers and the right answer during the design.



Select a topic

Character limit =

☐ Auto entry at ↵

☒ Show entry marker

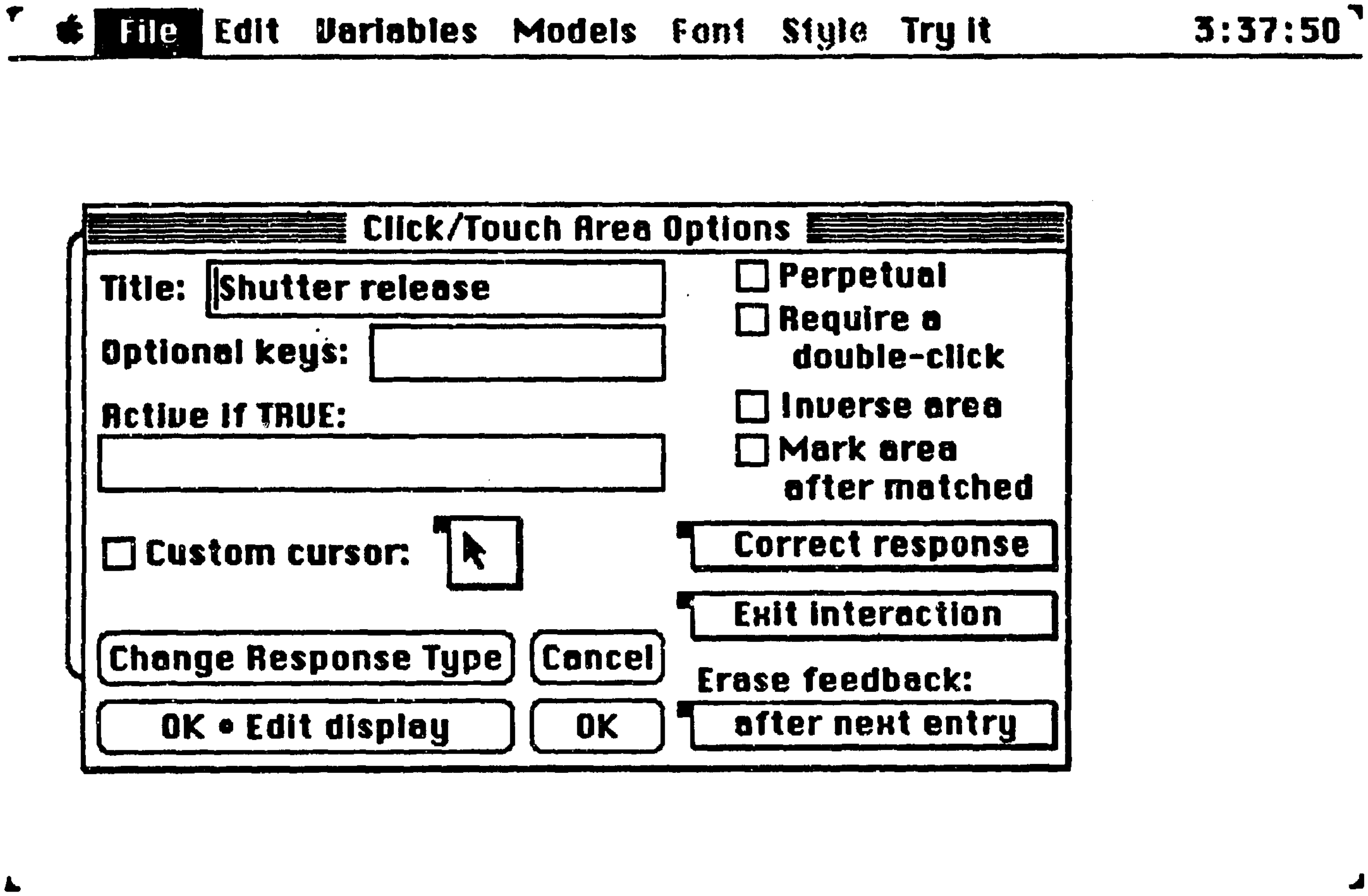
☒ Ignore null entries

☒ Pause before exiting

☒ Erase entry upon exit

Erase interaction: after each entry

Figure 17. This picture shows how to group each parts into a map icon and how to define the type of the branch.



Click/Touch Area Options

Title: Shutter release

Optional keys:

Active if TRUE:

□ Custom cursor:

□ Perpetual

□ Require a double-click

□ Inverse area

□ Mark area after matched

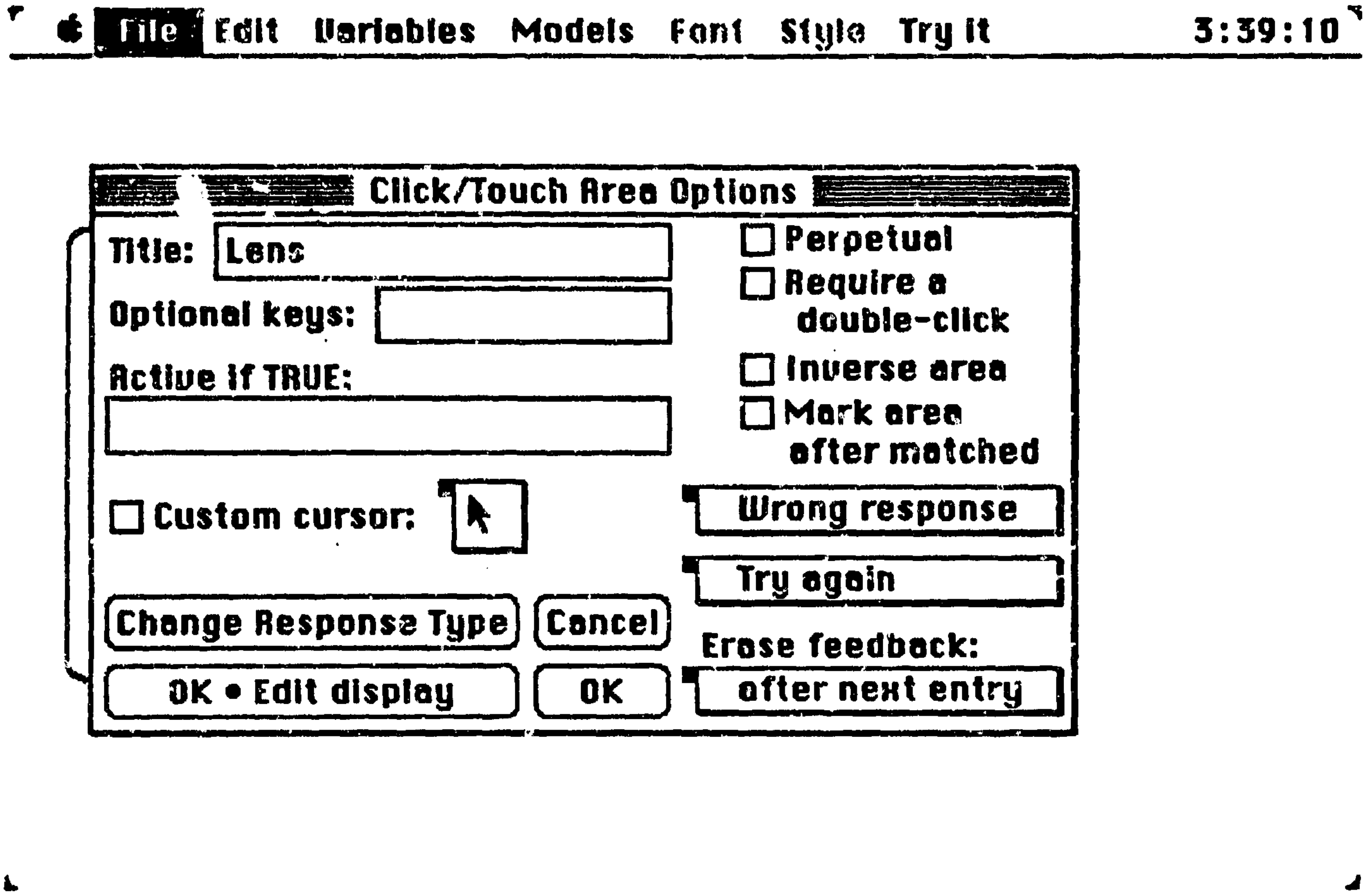
Correct response

Exit interaction

Erase feedback: after next entry

Figure 18. This is a example to show you how to define the right and the wrong answers and the action after the answer entered.

This is designed for the answer of the first question in the Part of a Camera. Shutter release is a right answer, and exit interaction is needed to go on to the next question.



Click/Touch Area Options

Title: Lens

Optional keys:

Active if TRUE:

□ Custom cursor:

□ Perpetual

□ Require a double-click

□ Inverse area

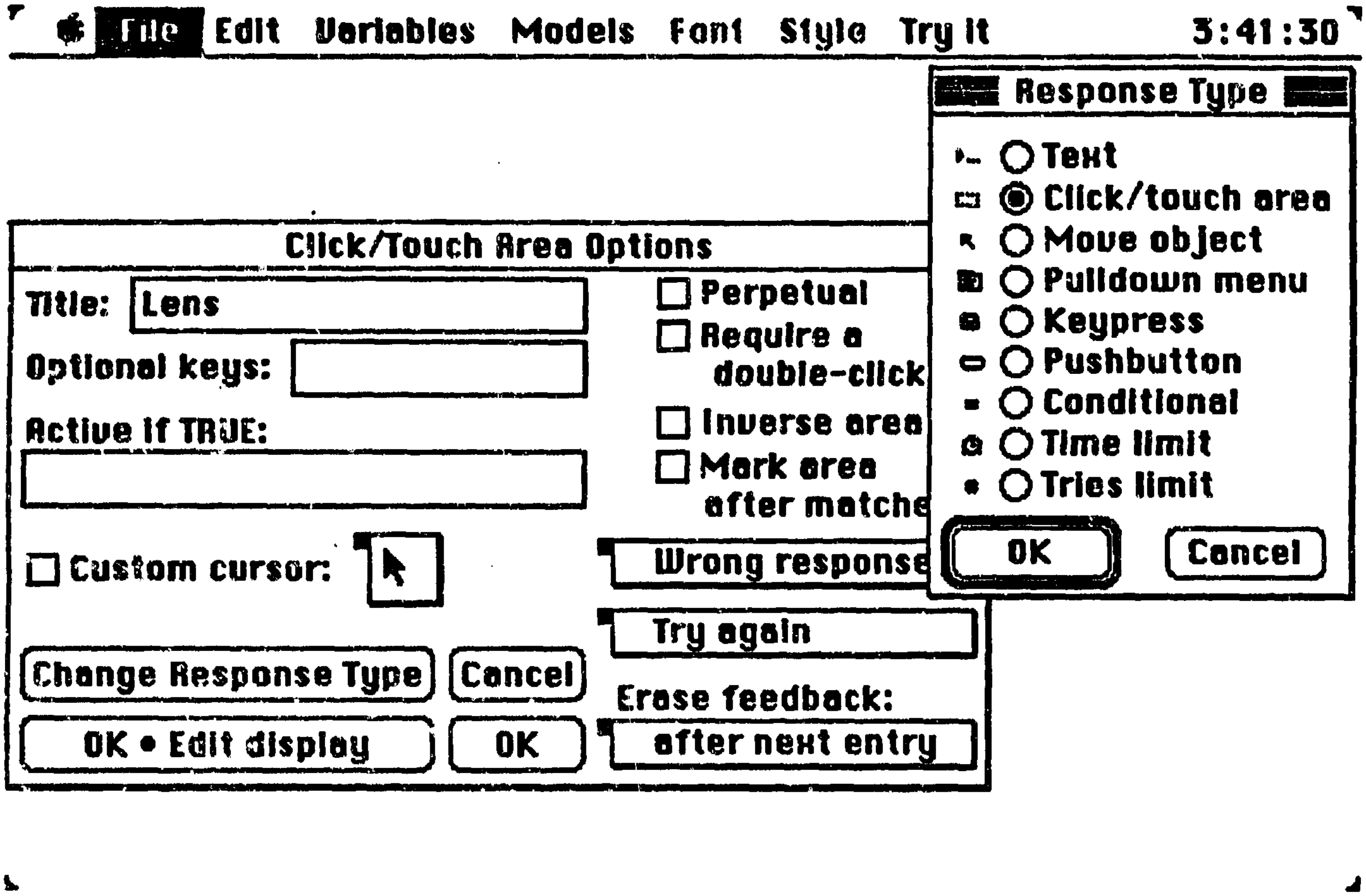
□ Mark area after matched

Wrong response

Try again

Erase feedback: after next entry

Figure 19. This picture shows the design for the wrong answer of the first question in Part of a Camera. Try again is needed to give a student a second chance to enter an answer again.



Click/Touch Area Options

Title: Lens

Optional keys:

Active if TRUE:

☐ Perpetual

☐ Require a double-click

☐ Inverse area

☐ Mark area after matche

☐ Custom cursor:

Erase feedback: after next entry

Response Type

○ Text
● Click/touch area
○ Move object
○ Pulldown menu
○ Keypress
○ Pushbutton
○ Conditional
○ Time limit
○ Tries limit

Figure 20. This picture shows you how to change the response type during the process.

When you click on the button of CHANGE RESPONSE TYPE, the list of Response Type will show on the screen then you can make a change very easily.

13.0 APPENDIX F
Review of QUEST

A BRIEF SUMMARY OF THE QUEST AUTHORING SYSTEM

by
Tracy A. Dunlap
Planning Systems Incorporated

for
US Army Research Institute
5001 Eisenhower Avenue
Alexandria, Va 22333-5600

1.0 INTRODUCTION

The QUEST Authoring System is an integrated set of programs used to create, present and manage Computer-Based Training (CBT) courseware in a DOS environment. It is an authoring system that includes an interactive authoring capability, a course administration method, various user registration functions, editors and utilities, and an authoring language. Figure 1-1 illustrates the overall QUEST Authoring System. Figure 1-2 shows the options available through QUEST's Main Menu selection screen.

2.0 INTERACTIVE AUTHORING PROGRAM

The interactive authoring program (called AUTHOR) uses dialog boxes or prompt windows to interface with the user. No programming experience is required to develop courseware using AUTHOR. The AUTHOR program is the heart of the QUEST Authoring System. The editors and all other programs in QUEST are used to support or enhance the functions in AUTHOR. AUTHOR is executed by selecting the **Author lesson** selection from the Main Menu screen, refer back to Figure 1-2.

AUTHOR operates on three levels: lesson, frame and object. A lesson created with AUTHOR is made up of a series of individual frames, and each frame can contain hundreds of objects, such as text, line drawings, shapes, animations and video sequences. Figure 2-1 helps to illustrate the relationship between these levels. Frames in this sample lesson would contain various objects that display text and graphics on the screen; the frames would also include objects that control branching between frames, play the video and allow students to make selections from a menu.

A lesson created in QUEST can be compared to a collection of slides. Each slide is like a frame in a lesson. The images, text and graphics on each slide correspond to the objects in frames. However, lessons move beyond the limitations of slide presentations. While slides can only appear in a fixed order, frames in QUEST can appear in any order.

In these lessons, students interact directly with the program by selecting menu options and answering questions (if applicable). QUEST analyzes student answers, records data on student performance and displays reports. Figure 2-2 describes some of the basic functions that can be performed at each level in AUTHOR.

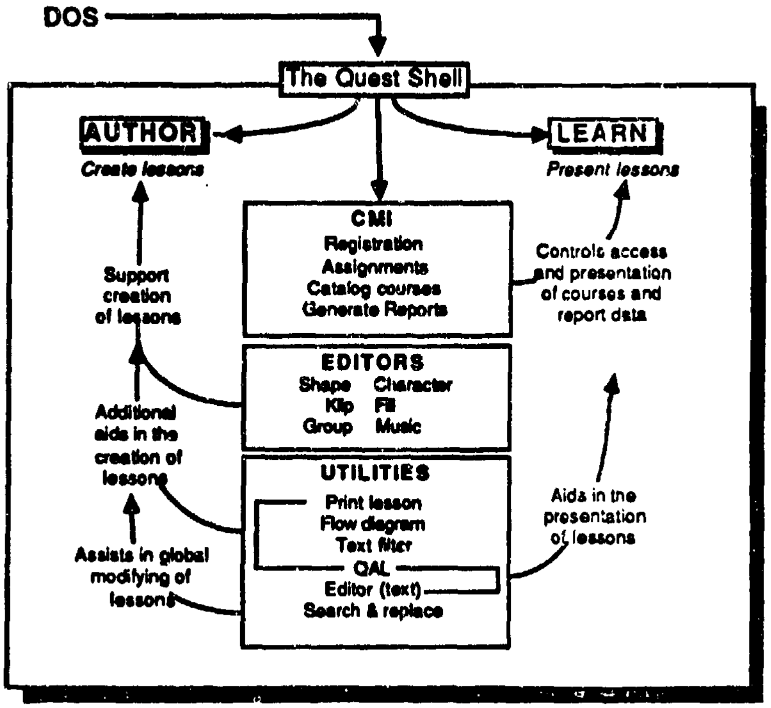
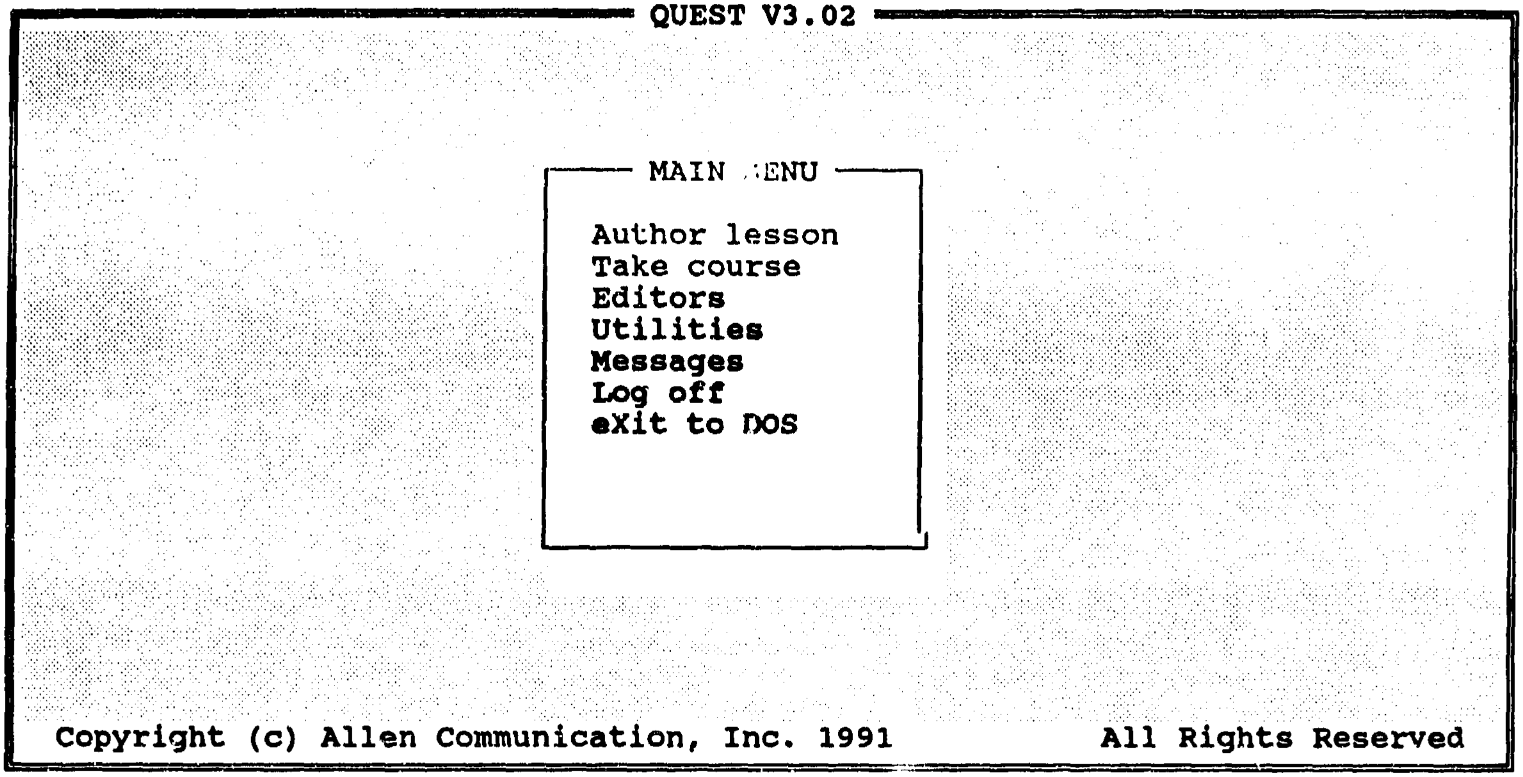


FIGURE 1-1: The QUEST Authoring System



Copyright (c) Allen Communication, Inc. 1991 All Rights Reserved

FIGURE 1-2: QUEST'S Main Menu

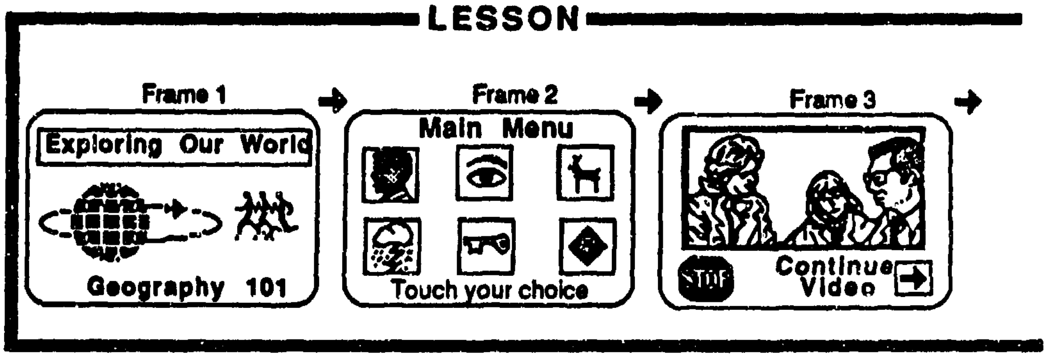


FIGURE 2-1: Level Relationships Within A Lesson

LESSON LEVEL:
Specify text fonts and graphics libraries used in the lesson;
Specify hardware setup: alternate input device (such as mouse or touchscreen), graphics adaptor (EGA,VGA), videodisc player;
Specify default resolution mode, lesson type and starting frame;
Set up lesson parameters for evaluating answers;
Define student keys.

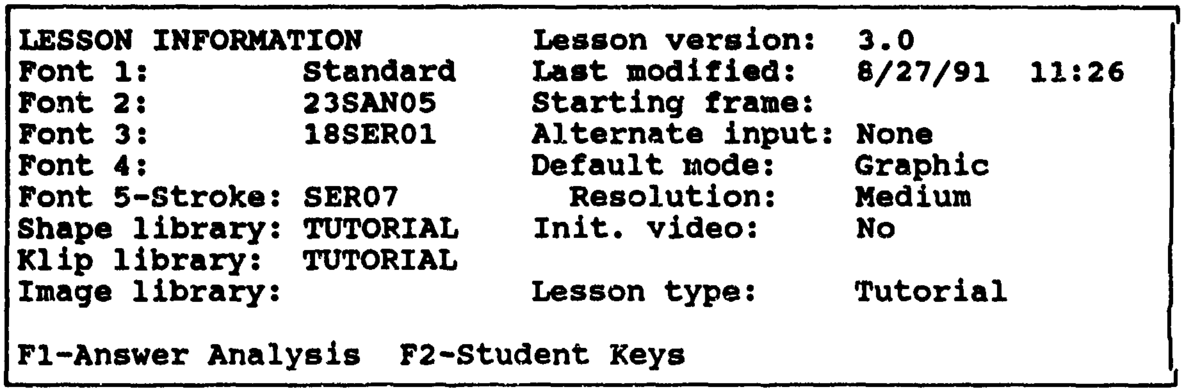
> **FRAME LEVEL:**
> Create frames;
> Copy, rename, delete, list and test existing frames.
>
> > **OBJECT LEVEL:**
> > Create text, graphics, shapes, images and klips;
> > Animate various objects;
> > Control resolution and color options;
> > Specify parameters to evaluate answers;
> > Set up the branching between frames;
> > Find and play video sequences and still frames;
> > Play digitized-audio and computer-audio files;
> > Create program objects;
> > Delete, copy, reorder and modify individual objects or groups of objects;
> > Search for specific objects;
> > Modify frame-level specifications;
> > List all objects in a frame.

FIGURE 2-2: The Basic Functions That Can Be Performed In AUTHOR

2.1 The Lesson Level

At the Lesson level, the author makes decisions that apply to an entire lesson. This includes specifying text fonts and graphics libraries to be used as the lesson is created and presented. QUEST supports a variety of text fonts. Libraries for shapes, klips and images can be created. All Lesson level functions are performed from the Lesson Information screen, refer to Figure 2-3.

Fonts and libraries must be specified at the Lesson Information screen before they can be used in the lesson. At this screen, the author specifies the starting frame and indicates which graphic device and resolution mode to be implemented in order to create and present the lesson. Parameters for evaluating student



```
LESSON INFORMATION           Lesson version:  3.0
Font 1:         Standard     Last modified:   8/27/91  11:26
Font 2:         23SAN05      Starting frame:
Font 3:         18SER01      Alternate input: None
Font 4:                      Default mode:    Graphic
Font 5-Stroke:  SER07          Resolution:    Medium
Shape library:  TUTORIAL     Init. video:     No
Klip library:   TUTORIAL
Image library:               Lesson type:     Tutorial

F1-Answer Analysis  F2-Student Keys
```

FIGURE 2-3: QUEST'S Lesson Information Screen

answers and special student definition keys are also set. For additional Lesson level capabilities, refer to Section 3.1.

2.2 The Frame Level

At the Frame level, the author develops the individual frames in the lesson. The author works from a prompt window that displays the options, refer to Figure 2-4.

Frames can be edited (created), copied, renamed, deleted, listed on the screen and tested from within AUTHOR. From the Frame prompt, the author can also access the Lesson Information screen (press I for Info). When a frame is being edited at the Frame prompt, the author will go to the Object level in AUTHOR. For additional Frame level capabilities, refer to Section 3.2.

2.3 The Object L. el

At the Object level, the author can create objects within a particular frame in the lesson. Some frames may include only a few objects, while others may include hundreds of objects; especially true when complex graphics and animations are used. Object-level functions are also selected from a prompt window, refer to Figure 2-5.

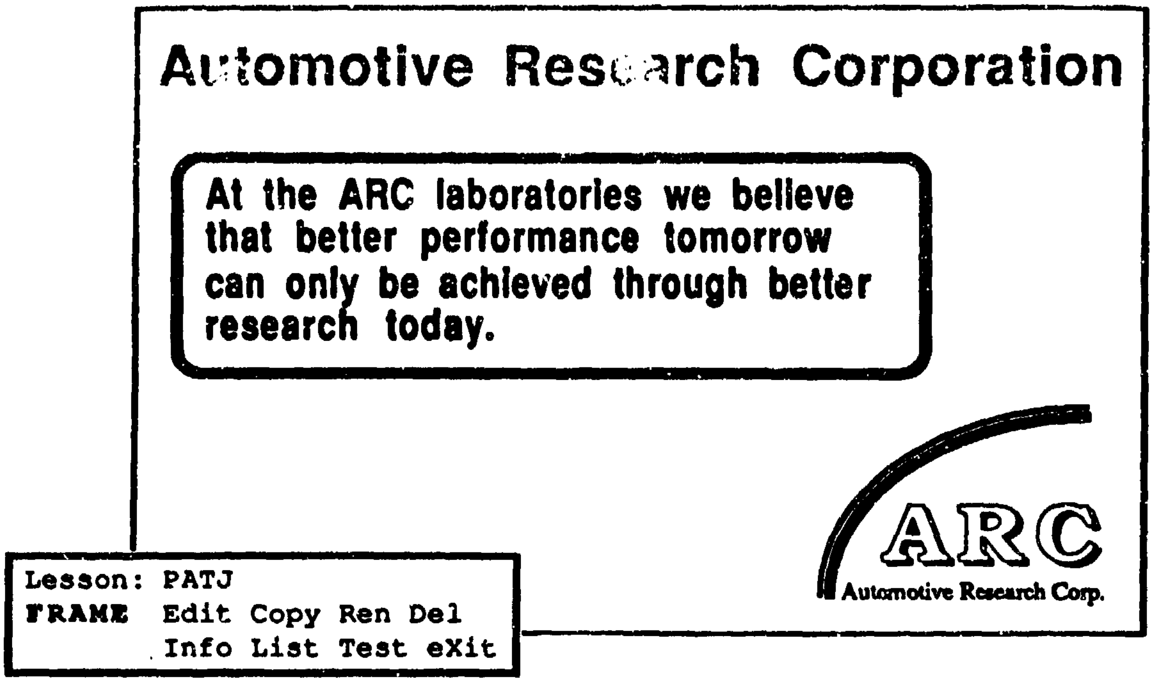
The author can select functions to create objects within a particular frame. It is on the Object level that most of the functions in AUTHOR are performed: typing in text, creating line drawings, setting up animations, playing video sequences and so on. For additional Object level capabilities, refer to Section 3.3.

2.4 The LEARN Program

The LEARN program is used to present courses that have been developed using AUTHOR. LEARN presents the course, lesson by lesson and frame by frame, according to the specifications made in AUTHOR. These programs are accessed from within the QUEST shell by selecting **Take course** from the Main Menu; refer back to Figure 1-2. For additional LEARN options, refer to Section 3.4.

2.5 Computer-Managed Instruction

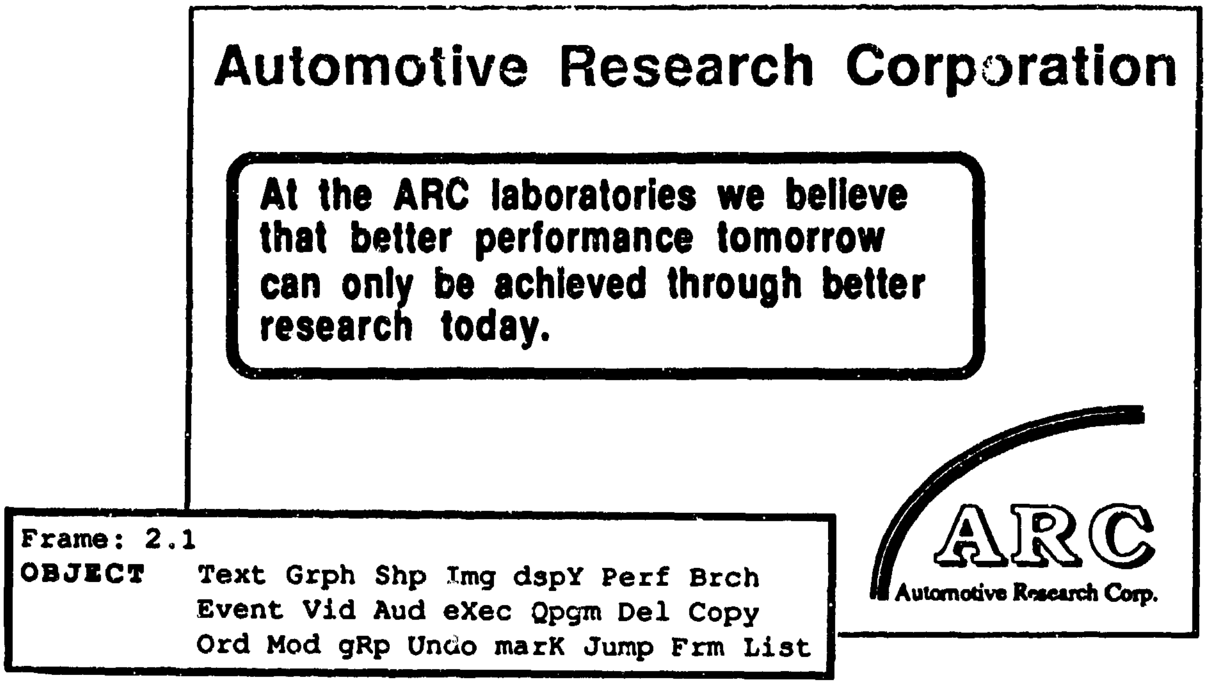
Computer-Managed Instruction (CMI) functions are also available in QUEST. The CMI functions allow the user to control access to the QUEST software by creating a log-on process which can restrict access to registered users. Access to courseware is further controlled by assigning courses to students. Student performance and course data can be recorded from which reports can



Automotive Research Corporation

At the ARC laboratories we believe that better performance tomorrow can only be achieved through better research today.

FIGURE 2-4: The FRAME Level Within AUTHOR



Automotive Research Corporation

At the ARC laboratories we believe that better performance tomorrow can only be achieved through better research today.

ARC
Automotive Research Corp.
Frame: 2.1
OBJECT
Text Grph Shp Img dspY Perf Brch
Event Vid Aud eXec Qpgm Del Copy
Ord Mod gRp Undo marK Jump Frm List

FIGURE 2-5: The OBJECT Level Within AUTHOR

be generated in order to help evaluate students and courses. These functions are retrieved by selecting **CMI functions** from the Main Menu; refer back to Figure 1-2. Additional CMI function capabilities can be found in Section 3.5.

2.6 QUEST Editors and Utilities

The utility programs provide a variety of functions to enhance courseware, increase productivity and aid courseware maintenance. This includes editors to create and maintain character fonts, shape libraries, group libraries, etc., and other programs to print lessons, generate flow diagrams, globally replace objects in lessons and capture screen images.

The authoring language (called QAL) is a QUEST utility similar to the PASCAL programming language. It includes many functions specifically for CBT. QAL can be used (as necessary) for more sophisticated applications such as complex animations and simulations.

QUEST editors and utilities can be accessed from the Main Menu (Figure 1-2). A more detailed description of the editors and utilities can be found in Sections 3.6 and 3.7, respectively.

3.0 QUEST CAPABILITIES

The following lists the capabilities of QUEST.

3.1 Lesson Level

Within the **Lesson** level, the Lesson Information screen sets up various functions and parameters, which affect the lesson and all its frames. The functions and parameters which may be changed are as follows:

- The addition of up to four fonts.
- Three graphics libraries (shapes, klips and images), which must be entered in order to access and use external graphic images within the lesson.
- The QUEST version number used to create the lesson.
- The date and time a frame was last exited.
- The lesson starting frame.
- Alternate input devices such as a mouse, touchscreen, light pen or tablet.

- Text/graphic mode and screen resolution.

- A toggle switch to accept a videodisc player.

- The choice of either creating tutorial or test lessons. Tutorial lessons can contain frames which may require student input and that the author of the lesson controls the branching between frames. Test lessons consist of a pool of questions. Each question is one frame and all frames require student input (an answer to the question). The questions are presented in random order; therefore, no branching options are available.

- Student performance answer analysis parameters may be set. Characters and words can be given special treatment when encountered in student answers. Certain characters and words can/cannot be ignored and specific word delimiters may be defined.

- Up to 23 special keys may be defined that learners can use as they work through a lesson. These keys allow learners to access a help screen, table of contents or glossary, to move forward or backward between frames and to perform many other functions.

3.2 Frame Level

Within the **Frame** level, various functions can be performed on the frames in the lesson. These functions are as follows:

- Create new frames and access existing frames.

- The option of erasing the screen or allowing the image from the previous frame to remain on the screen; thereby, enabling the user to build on what was created in previous frames.

- Specification of a frame to back up to.

- Copy frames from current or external lessons. All objects in the frame may be copied, or they may be limited to branching and performance objects or display objects only.

- The ability to rename, delete or list frames.

- Accessibility of the Lesson Information screen.

- The ability to test lesson frames without leaving AUTHOR.

- The option to save, restore or clear screen images.

3.3 Object Level

At the **Object** level, the ability to create a majority of the frame's contents is here. A frame of a lesson can be composed of several or several hundred objects. The functions (and objects) which may be implemented and/or created are as follows:

- The **Text** functions enable the user to create text objects.
  - Foreground and background colors may be changed.
  - An assortment of fonts are available: the Lesson Information screen should be modified accordingly.
  - Optional text sizing and letter/line spacing may be performed.
  - Characters or numbers can be inputted from the right to left. This is especially helpful for foreign languages, such as Hebrew, that write from right to left.
  - The same set of text attributes (options) can be saved for repeated use.
  - Pressing **Alt-H**, when editing text, brings up the Highlight prompt and its options. Material can be underlined, made bold, raised or lowered, shadowed or have its speed controlled when it is displayed on the screen.
  - Left and right margins may be set.
  - Various rectangular windows may be created to contain text.
  - A student's name or answer may be included within text, so that either one will appear in a designated place when the student takes the lesson.
  - Special characters, such as an arrow, may be inserted within text.

- The **Graphics** functions allow graphic creation by using various colors, line styles, fills and shape options.
  - The **Smooth line** option is useful for creating the effect of freehand drawings.
  - Within the **Fill** option, it is possible to mix original fill colors.

- The **Shape** options provide the ability to paste and animate shapes from a Shape Library onto the screen. Also, areas of a screen created in AUTHOR can be saved to a Shape Library for use elsewhere.
  - The **Normal** type option forces the shape to appear on the screen as it was stored in the Shape Library.
  - The **Super** type option enables the shape to be displayed on the screen with or without its background. Also, it can be made to appear either in front of or behind other objects on the screen.
  - When the Paste type is set to **Normal**, the mode field can be cycled between five modes of pasting shapes on the

screen; the **Super** type enables four separate modes. The different modes determine **HOW** the shape will appear. For example, whether the background and shape colors will mix when overlapped.
- Shapes may be animated along a defined path (path animation) or displayed in a series of shapes (cycle animation). The animation type (Normal or Super) and timing sequence used greatly affect the visual appearance of the display.

- With the **Image** option, it is possible to define, display, save, load and animate **images**, as well as display **bitmaps** and **klips**. The Image functions allow the user to work with graphics larger than those created with the Shape editor.
  - The graphics are not actually created with the image function. Images are created by cutting out a marked piece of the screen, which is composed of objects.
  - Bitmaps are full-screen images. Entire screens can be displayed using any one of 36 dissolves at a controlled speed. They replace the entire screen when used. Bitmaps cannot be animated.
  - Klips are like images in that they can be used to display a portion of the screen instead of the entire screen as with bitmaps. However, klips can be displayed using the same 36 dissolves used for full-screen bitmaps, and the display speed can be controlled. Like bitmaps, klips cannot be animated.
  - Image animation is performed in much the same way as with the Shape editor. The only difference is that when path animation is used, only one to three images can be used.
  - Up to 64 images can be stored within the Image library.

- The **Display** options allow changes to the default palette. The screen may also be cleared to a selected color.
  - In low or medium resolution, the 16-color palette cannot be changed to other colors. The arrangement of the colors in the palette can be changed, and the border color may be set.

- The **Performance** functions provide an extensive answer-judging capability.
  - There are five **types of answers** available in performance:

    Correct: Correct student responses.

    Incorrect: Incorrect student responses.

    Unexpected: Any responses that don't match the anticipated Correct, Incorrect or Neutral answers.

    Neutral: Responses that are neither correct nor incorrect; they are not scored; most often used for selection procedures, such as menu screens.

    Timed: The amount of time, that the student has to answer in, is controlled.

- Ccrrect, incorrect, unexpected and timed answers can be weighted.

- There are five **answer analysis techniques** available:

  Word: Compares words in the answer with words in the specified answer. The answer can be a single word, a series of words, whole phrases or sentences.

  Character: Matches answers character by character.

  Numeric: Compares the numeric value of a student's answer with a specified answer. By using numeric tolerances, answers can be set within a range.

  Position: Compares the position marked by the student to the positions or touch areas in the specified answer.

  Video: Compares video frames marked by students to a set range of video frames specified as the answer.

- **Input modes** control how the student enters the answer. The choice of Input modes is determined by the answer analysis technique used.

  A) For **Word**, **Character** or **Numeric** answer analysis, there are four input options.

  QAL: Allows the use of a QAL program to read the answer and to pass it to LEARN, where it is analyzed according to specifications.

  Normal: Characters are entered from left-to-right, as in a word processor or text editor.

  Math: Characters are entered from right to left. This is useful for math problems where students are required to calculate the answer from right to left.

  Length: A Normal input mode which allows the specification for a maximum number of characters that a student can enter for an answer.

  B) For **Position** answer analysis, there are different options.

  Point: A specific point on the screen is marked as the answer.

  Box: A rectangular touch area is defined as the answer.

  Circle: A circular touch area is defined as the answer.

  Near: A general answer area is defined.

- There are five different **Modifier modes**.

  Order: Account for differences in order within an answer.

Subset: Use subsets in an answer, ie, there are at least 3 out of 5 correct answers.
Extra: Allow extra words or characters in an answer.
Spell: Allow for variations in spelling.
Case: Allow for variations in Case.

- Multiple-field answers can be created. One answer can include up to 20 answer fields located at different places on the screen.
- Feedback can be created for each answer.
- The settings within a performance sequence can be saved as a template in a Performance library so that the same performance formats may be used in other frames and other lessons. Up to 100 performances can be saved.
- Performances may be edited.

- The **Branching** function controls the path or flow through lessons. Each lesson frame in tutorial-type lessons must contain a branch, whether created through the Branch or the Performance option.
  - Nine branch types are available:
    END: (End) Used to terminate a lesson.
    UBR: (Unconditional) Requires no special conditions; simplest and most common branch.
    CAL: (Call) Calls a frame or a series of frames.
    RET: (Return) Brings you back from a called frame to the specified place in the lesson.
    EXP: (Expression) Selects branching destination on the basis of an expression value.
    LSN: (Call Lesson) Accesses an external lesson.
    PGM: (Call Program) Executes external programs written in any compiled language.
    UNT: (Unit) Branches to the next unit.
    REP: (Repeat) Repeats performance in a frame.

  - Branches may be **timed** or **untimed**.

- The **Events** option controls the display of objects within a frame on the basis of time, specified keys, video frame numbers and touch input (or other alternate input devices). Event objects pause the processing of objects until the event is satisfied.

- The **Video** function helps integrate still-frame or motion videodisc images with text and graphic materials in a lesson. It allows the control of a videodisc player from a lesson, performing such functions as searching for a specified frame, playing a series of frames, etc.

- The **Audio** option enables computer audio files or digitized audio files created with a variety of digitized audio boards, to be played.

- Simple computer beeps can also be created. These beeps are stored in the frame, not in a separate file.

- The **Conditional Execution** function allows the control of execution or display of objects within one frame, changing the execution flow when certain conditions are met. When creating a conditional object, he author is indicating that if a specified condition is satisfied, the execution of objects will continue at the associated label.

- The **QAL Program Object** allows the performance of complex data manipulations or other operations not supported by AUTHOR. It does this by taking an external program written in QAL (Quest Authoring Language) and including it as an object in the frame. The QAL program becomes part of the lesson file; once the object is created, an external lesson file isn't needed.

- All objects listed in the upper-right corner of the Object prompt, with the exception of individual Branch objects, can be deleted via the **Delete** function.

- The **Copy** function allows individual objects to be copied. An object can be copied to any position on the screen and can be inserted at other positions in the list of objects as they appear in the Object prompt.

- The **Order** function allows the order of objects in a frame to be changed.

- Of the 22 functions listed on the Object prompt, the 12 listed below will create objects that can be modified using the **Modify** function:

  Text
  Graphics
  Shape
  Image
  Display
  Performance
  Branch
  Event
  Video
  Audio
  Conditional Execution
  QAL Program

  From these 12 function, 34 different types of objects can be created, all of which can be modified.

- The **Group** option allows multiple objects to be combined into a group; therefore, certain operations are much easier. The operations that can be performed are:

  New group
  Copy
  Move
  Flip
  Paste
  Order
  Scale
  Attributes
  Save
  Delete
  Rotate
  Temporary group

- The **Undo** option allows all changes made in a frame, since it was last entered, to be discarded.

- The **Mark** function helps identify (on the screen) the current object shown in the upper-right corner of the Object prompt.
  - It can help locate a specific object on the screen.
  - It can give the X and Y coordinates for the beginning point of an object.

- The **Jump** function enables the user to quickly move to any object in a large list of objects on the Object prompt, thereby eliminating having to cycle through so many in order to get to the one desired.
  - Objects can be found by object number, type or group.

- **Frame settings** can be controlled within the Object prompt.
  - The user can see the screen as he changes settings for resolution mode, dissolve options and background color display.
  - Frames can be marked as **bookmark frames**, so that if students exit a lesson or there is a power failure, they can return to the last bookmark frame, instead of starting again at the beginning of the lesson.
  - A **graphics** or **text mode frame** can be chosen. All QUEST functions are available in graphics mode. All text, graphics, image, shape and other functions, including various text sizes and fonts, can be used. In text-mode frames, no graphics and shape functions are available, and the prompts for these functions do not appear. No alternate fonts can be used. All text appears in standard font.
  - Frames can be designated as **Save Data** frames. Only certain student performance data is saved.

- The **List** prompt enables the author to either request a printed list of all objects in a frame or to see the list on the screen.

3.4 The LEARN Program

The courses created are presented with the LEARN program. The following can be input:

- The starting frame can be changed.

- Two types of **key options** can be made available to students as they take courses in LEARN:

  Defined Student Keys: Keys defined at the Student Key Definitions window at the Lesson Information screen.

  Student Control Keys: Keys automatically assigned in QUEST that allow learners to control animation, video motion and timed still sequences in the lesson.

- The current screen can be saved and subsequently restored, when the student exits the lesson to perform note taking, message and calculator functions, or if an external program is executed from within the lesson.

- Special prompts can be designed to suit a specific application.

- A score screen can display performance results, when a student terminates or completes a lesson.

- A Restart file option allows students to exit from a lesson and restart again at the last bookmark frame.

- It is possible to execute LEARN without using the QUEST shell or without logging in; the lesson is administered by using a DOS command statement.

3.5 Computer-Managed Instruction

With Computer-Managed Instruction (CMI), the following functions may be performed:

- The **Registration** function controls access to the QUEST software and to the courses created. All users must be registered before they can access QUEST programs or courses.
  - Users can be added, deleted, specific user information modified or printed.
  - User information includes:
    a) User Name (27 characters),
    b) Logon Name (13 characters),
    c) Password (9 characters),
    d) User Type (8 levels of user privilege),
    e) User ID (may be used in place of Logon Name or Password),
    f) Rank or Job (distinguishes users by military rank or responsibilities),
    g) Class Assignment (users are assigned to class(es)),
  - Users can be deleted one at a time, by class or all at once.
  - Up to 10 classes can be created for each course.

- The **Assignment** function controls student access to individual courses by making specific assignments.
  - Structured and unstructured students can only access courses that have been assigned to them.
  - Students may be assigned individually, by group or all at

once.
- Student assignments can be added, deleted, displayed or printed.

- The **Catalog** function is used to create and maintain a catalog of available courses and to help build a course structure.
  - Course information can be added, deleted, modified or printed.
  - The structure may be composed of a curriculum directory with multiple courses further consisting of lessons within courses.
  - When the course is created, directories for performance data, bitmap images, libraries and working files are automatically generated.
  - It is not a requirement to use the default directory structure.
  - Additional options such as Student Prompts, Score Screens and Restart File may be set.
  - Further divisions called **Units** can be created. One unit could include one or more lessons or several units could be within one lesson.
  - Units can be added, deleted, modified or printed.
  - Units may be shared between courses.
  - **Unit Save Data** options enable the specification of how much data will be recorded for each unit when a course is presented.

    a) **Answer Data:** Records answer data that includes: the actual answer entered by the student, the time spent in the frame, the type of response (correct, incorrect, etc.), and the score for the specified frame.

    b) **Lesson Data:** Records the following information for each frame: number of times entered, average time and the number of correct and incorrect answers.

    c) **Unit Data:** Records the following information for each student: time in each unit, number of correct and incorrect answers, score, total possible score, min/max/average score and the number of times lesson taken.

  - A customized student shell for presenting courses can be created. The shell includes logon and password prompts, case-sensitive options, colors, etc.

- **Report** functions can help monitor student performance and evaluate the effectiveness of the courseware.
  - Reports can be displayed on screen or printed. They can also be stored in ASCII text files which can be accessed with external database programs, such as DBASE III.

- Three types of reports can be generated, which can illustrate the data in either numaric or graphic (bar chart) format.

  a) **Student Reports:** Provides performance information on individual students and groups of students.

  b) **Lesson Reports:** Provides detailed information on student performance in each frame of a lesson.

  c) **Answer Reports:** Provides detailed information on the specific answers entered by students.

3.6 QUEST Editors

QUEST supports six different editing systems.

- **The Shape Editor:**
  - Allows the user to create libraries of shapes (graphics) that can be displayed in lessons.
  - Shapes can be created and modified a pixel at a time.
  - Shapes can be saved to a library from within AUTHOR and then loaded into the Shape Editor and modified.

- **The Character Editor:**
  - Provides the ability to modify an existing font to a new one and to save it under a new font name.
  - New fonts can be created without using an existing font to work from. The user is not limited to creating or modifying just the standard letters, numbers and symbols on the keyboard. Any symbol or new character that will fit in the work area displayed on the screen can be created.

- **The Klip Editor:**
  - Provides the ability to access klips stored in a Klip library, add klips from individual files into a Klip library and to modify them.
  - Possible to arrange and adjust a series of klips for animation purposes.

- **The Fill Editor:**
  - Possible to replace any or all of the 15 patterns of the Fill option available when creating graphics, shapes, etc.
  - Existing patterns can be modified or new ones created.
  - More than the 15 patterns currently in use can be stored.

- **The Group Editor:**
  - Enables the user to access groups of objects which have

been saved to a group library (using the Group option of the Object prompt in AUTHOR).
- The group may be shown on the screen, renamed, deleted or replaced with another within selected lessons.

- The **Music Editor:**
  - Provides the ability to create audio files with a piano-type sound.
  - Possible to duplicate existing melodies or create new ones to support lessons.

3.7 QUEST Utilities

Eight separate Utility functions are accessible.

- The **Print Lesson** program is used to make printouts of group, shape and klip lessons and of lesson information (including frame names, lesson and frame settings, screen images, object lists, embedded statements, text, branching and performance settings and lists of called frames). This data can also be written to a file.

- The **Flow Diagram** option allows the structure of QUEST lessons to be displayed or printed as flow charts. Each frame is represented in its relationship to the other frames in the lesson.
  - The display includes frame names, branch types, answer types and a list of orphan frames (frames that were created but not used within the lesson).
  - Provides the ability to search and delete frames, modify branching between frames and to restart the flow diagram from any frame within the lesson.

- The **Text Filter** program allows the use of text files created with the QUEST text editor, external text editors or word processors as QUEST macro files; the files can be imported into the lesson. Special characters and commands must be inserted into the ASCII text files prior to conversion. By preparing the files, special effects such as highlighting and underlining may be performed prior to conversion.

- The **Search and Replace** utility is used to search for and modify parts of lessons in a course. When lessons are modified, the user can globally replace text, graphics, lesson information and frame settings, objects and required files for an entire lesson by making just one modification.

- **QED** is a full screen text editor that is used for the Notes and Message functions in LEARN and for the Message function from the Main Menu.

- Limited to 250 lines per file, so it's not intended to be a programmer's editor or word processor.
- Automatically accesed when the Message function is selected from the Main Menu or when a student elects to take notes or send a message while taking a lesson.
- Normally accessed via the Utility Menu.
- Can access QED from the DOS prompt to edit files that are not associated with a course.

- The **QUEST Authoring Language** (QAL) is an authoring language similar to PASCAL. Even though it does not include all PASCAL capabilities, it includes many additional features that are used specifically for developing courses. QAL provides greater power for developing complex animation sequences, simulations, customized CMI functions and special answer analysis.

- The **Capture Image** program (CAPIMG) allows the user to capture screen images displayed by **almost** any program (such as PC Paintbrush). Screens can be captured and saved to a file. These files are accessed from within AUTHOR using the BMAP option from the Image prompt. Capture Image is a memory resident program. Once it is loaded on the system, it remains in memory until the computer is rebooted or turned off. It supports both EGA and VGA graphics adaptors.

- The **Calculator** program provides basic calculator functions for students to use while taking lessons.

3.8 Miscellaneous Options

- The **Messages** option provides a method to read and send messages among all QUEST users, including students, teachers and managers. Messages can also be sent among groups, such as to all students taking a particular course and from students to the instructor of the course they are taking. The Message function is especially useful for classes using a computer network.

- **Embedded statements** make it possible to define variables and assign values to the variables. These values can be displayed on the screen or used to track student performance, generate random values for numeric problems, control a student's path through a course and many other functions.

- **Macro files** can be created in QUEST which can access external text files and keystroke files.
  - The most common use of Macros is to bring text from a text file into a QUEST lesson in order to quickly and easily create the text objects in the lesson.
  - Macros can be used to save keystrokes that are repeated

often, for example, to set up the Lesson Information.

3.9 General

- AUTHOR is built around a series of prompt windows, that display available options and direct the user from one step to the next as lessons are created.

- **Function keys are used for many procedures in AUTHOR. The F1 and F2 function keys perform unique functions at individual prompt windows. The other function keys (F3 through F10) operate the same way throughout AUTHOR; they are global functions. Each of these functions operate as follows:**

  F3: Execute DOS commands and external programs from anywhere within AUTHOR.
  F4: Displays an enlarged 10 by 10 pixel area view, magnifying it 10 times.
  F5: Accesses additional cursor-control options such as color, size and location.
  F6: Manipulates the size and position of the major prompt windows.
  F7: Clears and then redisplays the actual contents of a frame; the command refreshes the screen.
  F8: Saves the object created; **Alt-F8** discards the object.
  F9: Displays context-sensitive help screens; on-screen description of the prompt window currently within. The user can also press **F9** to get more information on messages or warnings when they appear.
  F10: Toggles the mouse select flag On or Off.

4.0 QUEST LIMITATIONS

The following lists the limitations of QUEST.

- **QUEST shouldn't be used as a testing method, for the following reasons:**
  - **One frame equals one question,**
  - **Questions are randomly chosen and presented,**
  - **There is a 150 question test limit.**

- **In the Lesson level, it isn't possible to change the lesson type, once one frame has been created.**

- **When in the Object level:**
  - **Highlight text options are not available within text mode frames.**
  - **Bitmap fonts cannot be made larger or smaller.**

- When the smooth option is on, the cursor keys cannot be used between points.
- If shapes are animated using multiple shape images, then they must be consecutively stored in the Shape library.
- No more than 127 points can be marked for path animation.
- Unexpected answers must be entered last or all answers within the performance object will be incorrect.
- Limited to only true/false conditions when the Conditional Execution function is used. This could lead to a numerous amount of nested loops.
- Branches cannot be deleted. They must be modified or replaced with a performance object or another branch.

- When using Computer Managed Instruction:
  - The logon name and password are case sensitive. The only way to get around this is to create an entirely new student shell.
  - If a class name is deleted, the user is not asked to confirm the deletion.

- The Report function will not print reports until at least one unit within the course has been specified. In addition, the performance data must be saved in the Unit Window.

- A flow diagram cannot be created for a test lesson.

- When using the Search and Replace function, words or phrases that include attribute and highlight markers from the Text Filtering program, cannot be replaced. The Search and Replace function passes over these words and phrases without replacing them.

- QED editor is limited to 250 lines per file. This routine is not intended to be an actual editor.

- The Capture Image program doesn't work in ALL cases. For example, the first screen displayed in the EXAMINER package cannot be captured; therefore, the original graphic screen must be in a special format.

- It isn't recommended to load memory resident programs like CAPIMG from within AUTHOR via the F3 function. There are memory limitations.

- For the Message function, a Student or Quester cannot delete messages. Only the Course Manager or higher may do so.

- A mouse should not be used when creating macros, especially keystroke files.

- Statistical ability is extremely limited.
- If the user tries to logon to QUEST and can't, there is no way to get back to DOS. The machine must be turned off and then back on.

14.0 APPENDIX G
Review of TENCORE

A BRIEF SUMMARY OF THE TENCORE LANGUAGE AUTHORING SYSTEM

by
Fu Li
Statcom Inc.

for
US Army Research Institute
5001 Eisenhower Avenue
Alexandria, Va 22333-5600

1.0 INTRODUCTION

The TenCORE Language Authoring System is a complete programming environment specially enhanced for implementing computer-based training material. The TenCORE Author Language has complete facilities for display creation, response input and analysis, and data manipulation in a structured programming environment.

There are totally seven options from the TenCORE Main Menu. They are Edit, Information, File operations, Lessons, Manage groups, Group router, and Utilities. Each options on the TenCORE Main Menu is briefly described as the following sections.

2.0 BASIC FUNCTIONS IN THE EDIT OPTION

The Edit option is the most-used of all options on the TenCORE Author Menu. It leads to the Source Editor, from which you create new TenCORE applications and edit existing ones.

TenCORE applications are edited and stored in computer files called source files. The File Selection Display is TenCORE's mechanism for creating or editing such files.

A source file is split into sections called blocks which contain graphics, character sets, documentation, and programming. The Block Selection Display lists the blocks in a source file and lets the programmer edit them.

The basic functions in the File Selection Display and the Block Selection Display are listed as follows:

- create new source files
- see and change existing source files
- create new blocks
- see and change existing blocks
- delete unwanted blocks
- run the application from the beginning, or starting at any block
- put blocks in a different order
- copy blocks from another file
- disable or enable blocks
- print the source file
- transfer programming to and from ASCII files

2.1 File Types used in the EDIT Option

There are five file types special to TenCORE. Each of these file types has its own DOS file name extension (the part of the file name after the dot). The file types are listed as follows:

- .SRC TenCORE Source Files
  Source files contain TenCORE applications in editable form. Whenever you edit TenCORE Language source code, image blocks, or character set blocks, you are working in a TenCORE Source File.

- .BIN TenCORE Binary Files
  Binary files contain TenCORE applications in the executable form delivered to end users. TenCORE binary files are smaller and faster to execute than TenCORE source files. And because TenCORE binary files are encoded, they provide a degree of security for your applications.

- .DAT TenCORE Dataset Files
  Dataset files provide a medium for the permanent storage of raw data. The TenCORE Language includes commands for accessing data stored in datasets.

- .NAM TenCORE Nameset Files
  Nameset files provide a more sophisticated medium for permanent data storage. Namesets are organized as multiple named sets of logical records, and the TenCORE Language includes special commands for nameset management.

- .GRP TenCORE Group Files
  Group files make it possible to pre-register users of TenCORE applications by name, to establish curricula for users, and to keep data on both users and programs.

2.2 Block Types used in the EDIT Option

There are five different block types that can appear in a source file. They are briefly defined as follows:

- **Text blocks** are used for documentation or other notes to yourself or other authors. They have no significance to TenCORE and are ignored during execution.

- **Defines blocks** are used to define global variables in the program.

- **Unit blocks** are blocks which store TenCORE Language source code for TenCORE units. They can be edited and executed.

- **Image blocks** contain bit-mapped graphic images usually created with the Image Editor. The picture stored in an image block can be displayed as needed in a TenCORE program.

- **Character set blocks** contain dot patterns for individual characters which form typefaces, special alphabets or symbols. A charset can be used to print Cyrillic, Greek or other special text on the display.

2.3 Image Editor used in the EDIT Option

Image Editor is used to create and edit the Image blocks. It operates with icons, the descriptive pictures, to edit bit-mapped graphic images. Each icon represents a different function and is displayed in a menu on the left and bottom of the screen. These icons allow you to freehand draw, paint, move, cut-and-paste, and rubberstamp portions of images. They also can be used to create filled and unfilled circles, ellipses and boxes. A variety of colors, patterns, symbols, line widths, typefaces, airbrush densities, and image scaling features are included.

The advanced features are listed as follows:

- **On-Screen Help** -- At the touch of a button, you can access help for each selectable function.
- **Lasso** -- The ability to cut and paste or rubberstamp a user-defined area.
- **Color Swapping** -- From within the Color Swap menu, you can re-color the entire screen or any portion thereof.
- **Palette Editing** -- (on graphics devices with 16 or more colors) the ability to customize a palette of colors for drawing and painting.
- **Text Composition** -- the ability to directly input text files in a selected font, style, and size.
- **Image Scaling** -- the ability to interactively shrink or enlarge portions of the display.
- **Curve Fitting** -- the capability to automatically draw a curved line between plotted points.
- **Menu Switching** -- the option to move the icon menu from the left and bottom of the screen, to the right and top, or to remove it from the screen altogether.
- **Fat Bit Editing** -- the ability to magnify portions of an image and edit individual pixels.

- **Symbcl Creation** -- the ability to create symbols to be used as hatch styles for logos and to store them in a symbol library for future use.

- **Image Rotation** -- the ability to rotate images in 90 degree increments and to flip images to create mirror images.

- **Non-Destructive Image Movement** -- the ability to move a portion of an image around the screen without destroying the screen area through which it passes.

- **Snap Grids** -- the capability to place an invisible grid on the screen for "lining up" charts and graphs.

Following is a brief description of the icons used in the Image Editor. They play the totally different functions from the icons used in the Course of Action.

☒ CLEAR -- clears screen; removes icons

● EXIT -- exits Image Editor and returns to DOS

💾 DISK -- stores/retrieves PIC, TXT, PAL files

□ BLANK -- has no function.

A TEXT -- selects type style, size; places text

✂ SCISSORS -- lassos, rubberstamps, stores CUT files

P PAINT -- fills area with selected color, pattern

AIRBRUSH -- sprapaints the screen

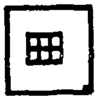
✏ PENCIL -- draws freehand lines, figures

ERASE -- erases selected screen areas

LINES -- draws straight lines; draws polygons

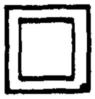
ROTATE -- rotates images 90 degrees at a time

FATBIT EDIT -- edits images pixel by pixel



GRID -- creates visible and invisible grids

CIRCLE -- draws circles and ellipses



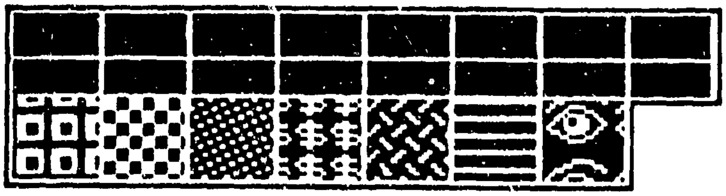
RECTANGLE -- draws rectangles

COLOR SWAP -- manipulates colors and palette

BRUSH -- controls drawing actions

LINE STYLES -- selects
smooth, dashed, dotted lines

LINE WIDTHS -- selects
thickness of drawing line



COLORS/HATCHPATTERNS -- available color/patterns

3.0 BASIC FUNCTIONS IN THE INFORMATION OPTION

The Information option leads to the On-Line Information Facility. The Information Facility contains condensed information about the TenCORE Authoring Language, including concise writeups of:

- commands
- system variables
- mathematical operators
- error message
- other programming features

These writeups can also be accessed directly while editing a TenCORE lesson. The user can request a writeup on a particular command, examine the writeup, and return to the point you were editing.

4.0 BASIC FUNCTIONS IN THE FILE OPERATIONS OPTION

The File operations option leads to the File Management Menu. The File Management Menu displays length and last-edit information about files and the following functions:

- create new files
- make copies of files
- rename files
- lengthen or shorten files
- delete files
- execute DOS commands

5.0 BASIC FUNCTIONS IN THE LESSONS OPTION

The lessons option allows you to try out lessons and to see them just as an end-user would. It has two primary functions:

- executing ready-made programs without leaving the TenCORE Author environment

- trying out end-user binary versions of programs under development to make sure they perform correctly in student mode

6.0 BASIC FUNCTIONS IN THE MANAGE GROUPS OPTION

The Manage Groups option provides access to the TenCORE Group Activity manager. The Group Activity Manager is a facility for:

- pre-registering users by name and group
- setting up curricula for groups
- collecting data on users (programs completed, scores, etc.)
- collecting data on applications (such as missing files, execution errors, etc.)

The Group Activity manager gives users convenient access to catalogs of on-line activities, and administrators a means for tracking the progress and status of each user and the usage of each activity. It can be used on stand-alone micro computers and local area networks.

7.0 BASIC FUNCTIONS IN THE GROUP ROUTER OPTION

The Group Router option allows you to use or test entire curricula (set up using the Group Activity Manager) in much the same way that the Lessons option allows use or testing of individual applications.

8.0 BASIC FUNCTIONS IN THE UTILITIES OPTION

The Utilities option provides access to the following functions:

- print a file
- memory usage
  The memory usage display shows the amount of memory in use by TenCORE, together with other information about memory allocation.
- change standard character sets
  This function provides a way to change the standard character sets used on graphics screens to characters of your own design.
- disk inspection
  This function allows the byte-by-byte inspection and editing of TenCORE files as raw hexadecimal data. It should be used with caution, as changing critical data could cause a file to become unrecognizable to TenCORE.
- call a DOS command

This function allows the execution of DOS commands, or even a temporary return to the DOS prompt, without the necessity of restarting TenCORE to resume authoring. On completion of the DOS command or exit from the DOS prompt, execution returns automatically to TenCORE.

9.0 THE TENCORE CAPABILITIES

Based on the learning, there are some comments of "good things" and "bed things" about this system. The capabilities of each options are summarized in the section 2.0 to 8.0. Now let us look at the "good things", in general, of the system:

- According to the programming ability, the system provides very powerful functions to build a different kind of applications for the end user.

- TenCORE provides character sets editor to let you create and edit characters singly or in groups. The Single Character Editor is best for designing text-type characters such as foreign alphabets. The Multiple Character Editor can be used to edit any combination of characters -- a useful tool for working on entire alphabets and for creating character-based graphics. The two editors can be used interchangeable on any charset.

- The Charset Editor in the TenCORE is more powerful than the one in the MicroCAT based on the functions provided by each system.

- Graphics ability is provided by the system from the Graphics Editor for the unit and to edit Image Blocks in the EDIT option. The Graphics Editor is a program generator, writing the program for you from the display you create. Image Block is a type of block that stores icons to be used to represent graphics objects. You can use these icons to create any image you need.

- The Group Activity Manager gives information of both, the user and the activity, by the time the user spent in each activity, the completion status, the most recent session, and the score of each activity from each user. These information will help the designer and the user to improve their work.

- On-screen HELP provides the convenient way for the programmer and the end user.

- File management functions are provided by the system. These functions are convenient for the programmer.

- The system provides the security function in the performance of the courseware. There is a requirement to ask a password to get into any lessons. The binary file is used for the student (not source file) that also prevents any changes from an unauthorized user.

10.0 THE TENCORE LIMITATIONS

Now, let us look at the "bed things", limitations, in the system from the programmer and the end user views of the point:

- The quality of the on-screen displays for the text and image editing is not as good as the menu screen designed.
- Color functions in the program are not working very well.
- There is no good support for the text generation and editing.
- There is a need to have a tutorial lessons (samples) to show how this system works from the beginning to the end.
- There are too many different ways used for the same function' in the different process, such as function EXIT, sometimes the function key F1 should be used, and sometimes function key F6 or F9 are used. Even the commands are show on the screen, it is still confused by the user during the implementation.
- The function Import cannot deal with the graphics.
- There is no statistics analysis method provided for the results of each activity. It only shows the score for each activity from each user.
- There is no "Bookmarking" ability. Then the user can go to any activity in the file (group) from the activity router menu.
- The print out ability provided in the system cannot print out the source file of image and character set blocks. It can only print out the text, defines, and unit blocks.

11.0 THE COMPARISON WITH THE COURSE OF ACTION AND THE MICROCAT

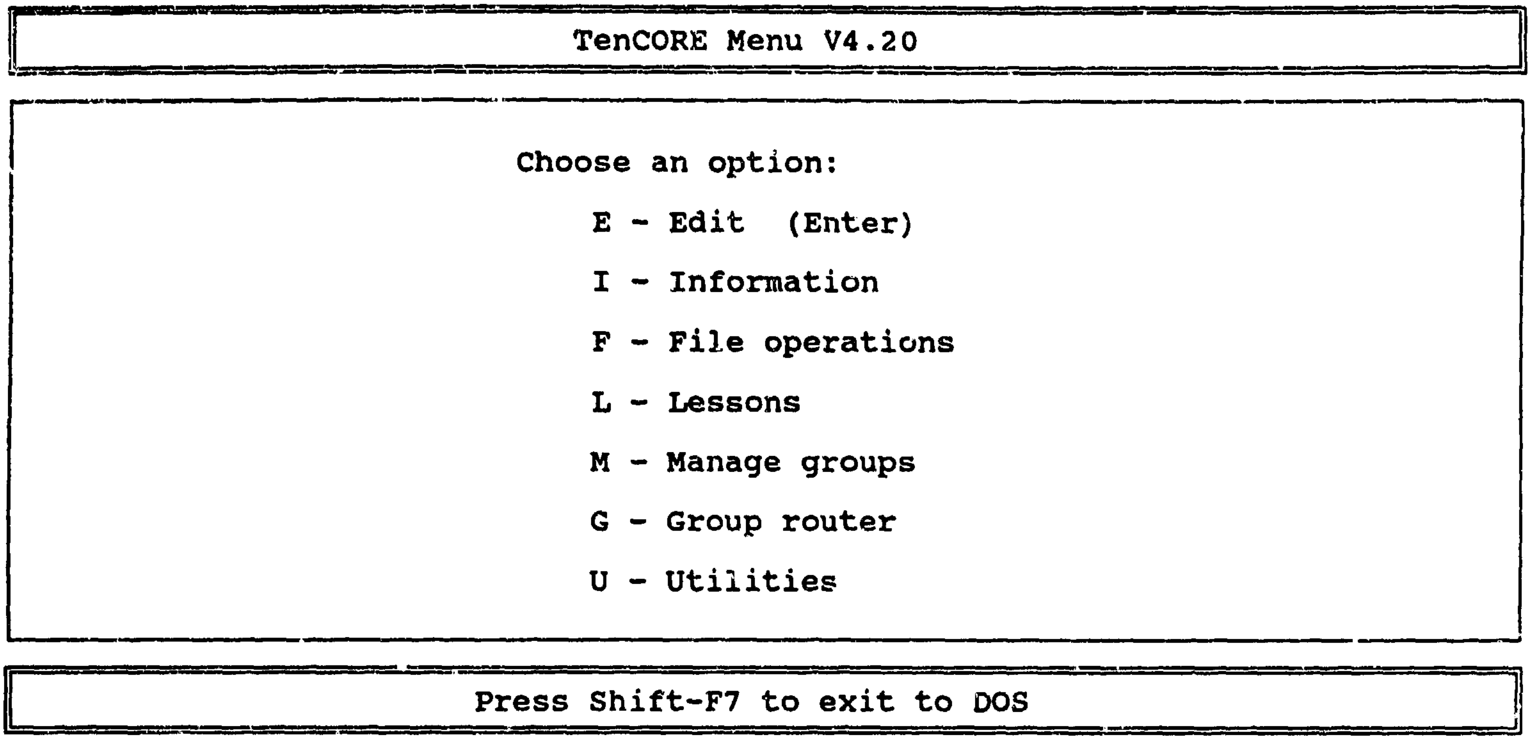
The TenCORE and the Course of Action are both use icon functions, but there is a big difference between them. In the

TenCORE, icons are used only for the image editor, e.g., to simplify the program of the graphics part needed in a lesson. In the Course of Action, icons are used as more powerful function keys. These icons are actually used to "write" a program. Using icons with pull-down menus, you can easily organize a lesson. The Course of Action also can use these icons to show the overall diagram of the lesson that you create.

The TenCORE and the MicroCAT are both having special character editing abilities, but the TenCORE provides more functions than the MicroCAT does. The TenCORE can edit multiple character set simultaneously. The graphic ability in the TenCORE is better than the one in the MicroCAT, plus the TenCORE provides more user-friendly interface than the MicroCAT does. Then the MicroCAT supports the statistics analysis methods to evaluate the results of the test that the TenCORE cannot.

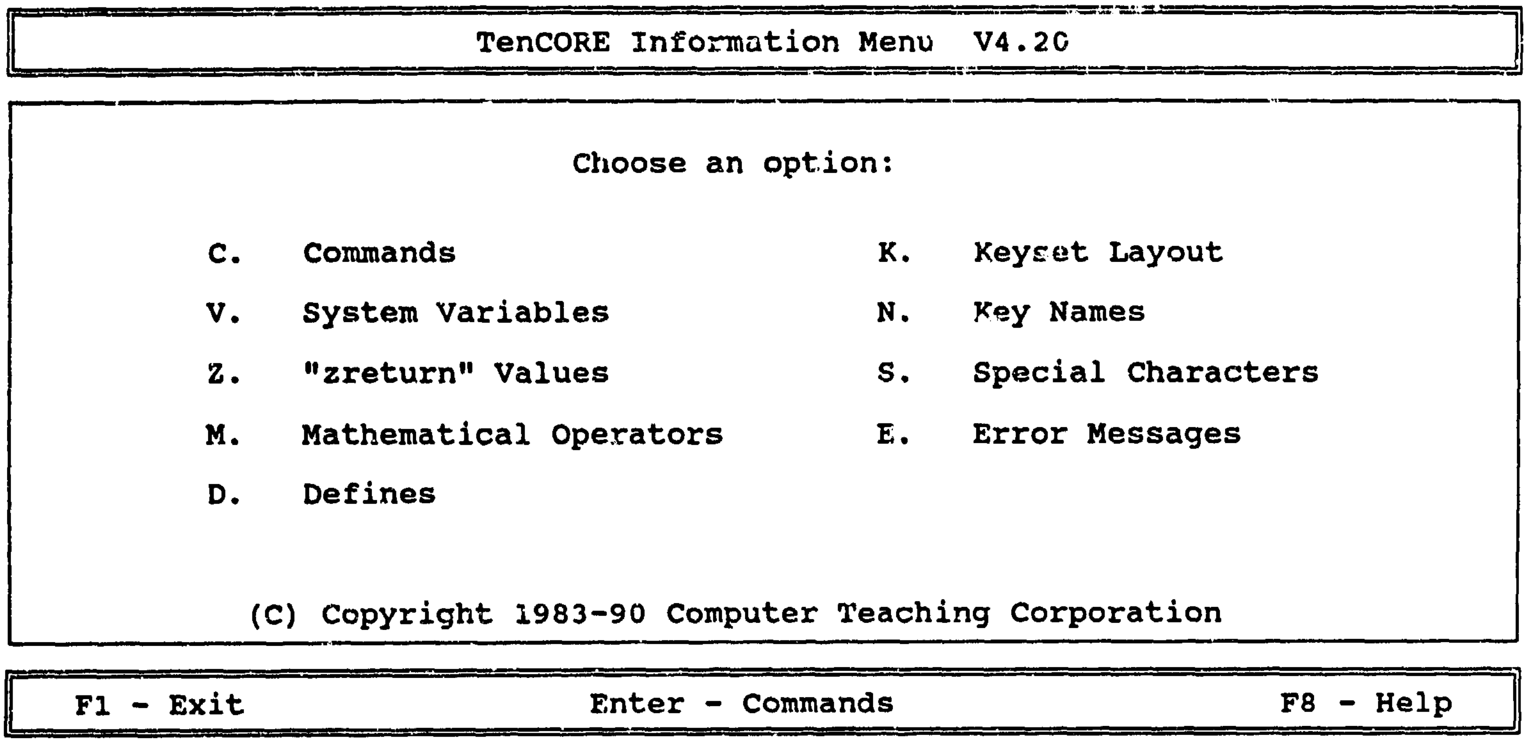
APPENDIXES

Some of the Menus used in the TenCORE System



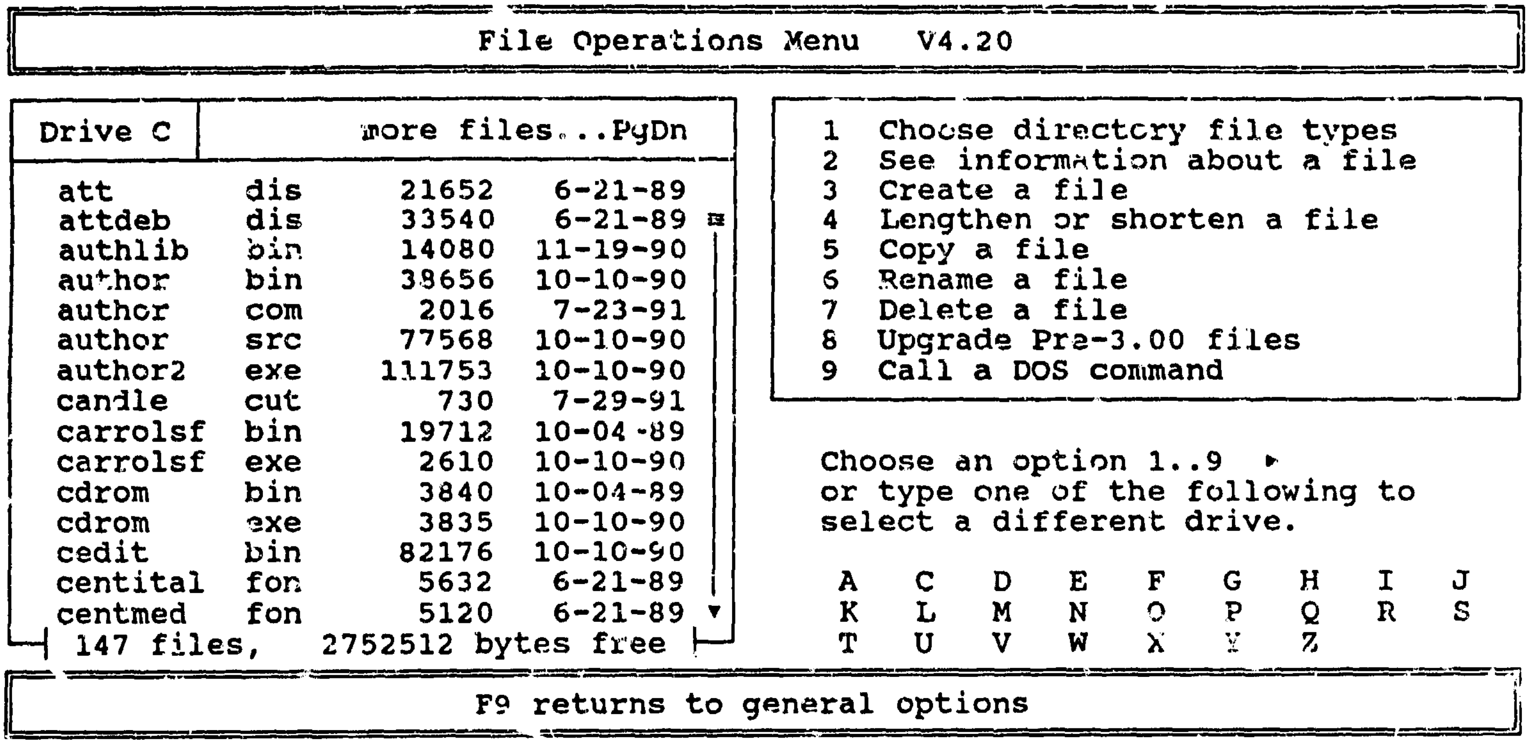
```
TenCORE Menu V4.20

Choose an option:
     E - Edit  (Enter)
     I - Information
     F - File operations
     L - Lessons
     M - Manage groups
     G - Group router
     U - Utilities

Press Shift-F7 to exit to DOS
```

Figure 1. TenCORE Main Menu



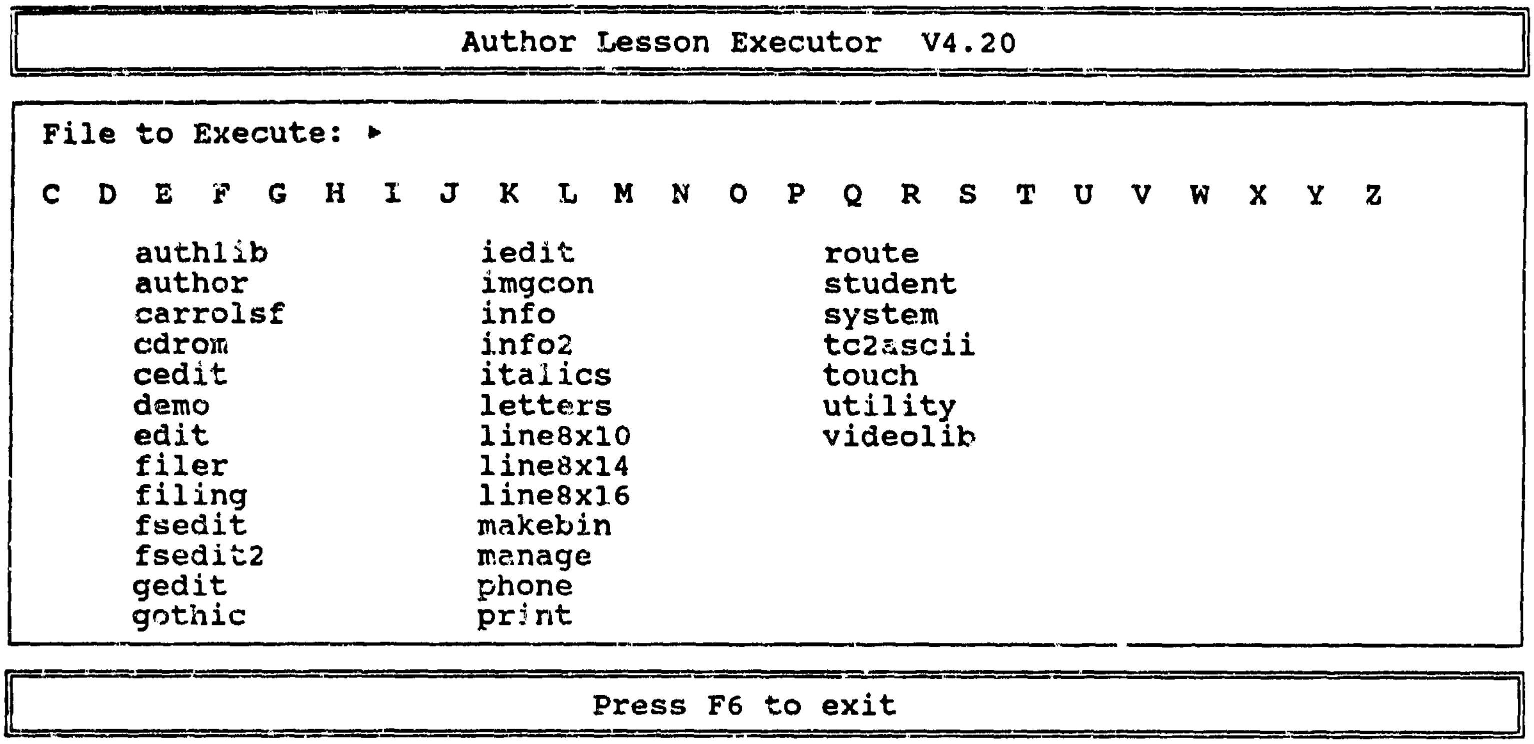
```
                     TenCORE Information Menu  V4.20

                          Choose an option:

      C.    Commands                        K.    Keyset Layout
      V.    System Variables                N.    Key Names
      Z.    "zreturn" Values                S.    Special Characters
      M.    Mathematical Operators          E.    Error Messages
      D.    Defines

          (C) Copyright 1983-90 Computer Teaching Corporation

  F1 - Exit                 Enter - Commands                 F8 - Help
```

Figure 2. TenCORE Information Menu.



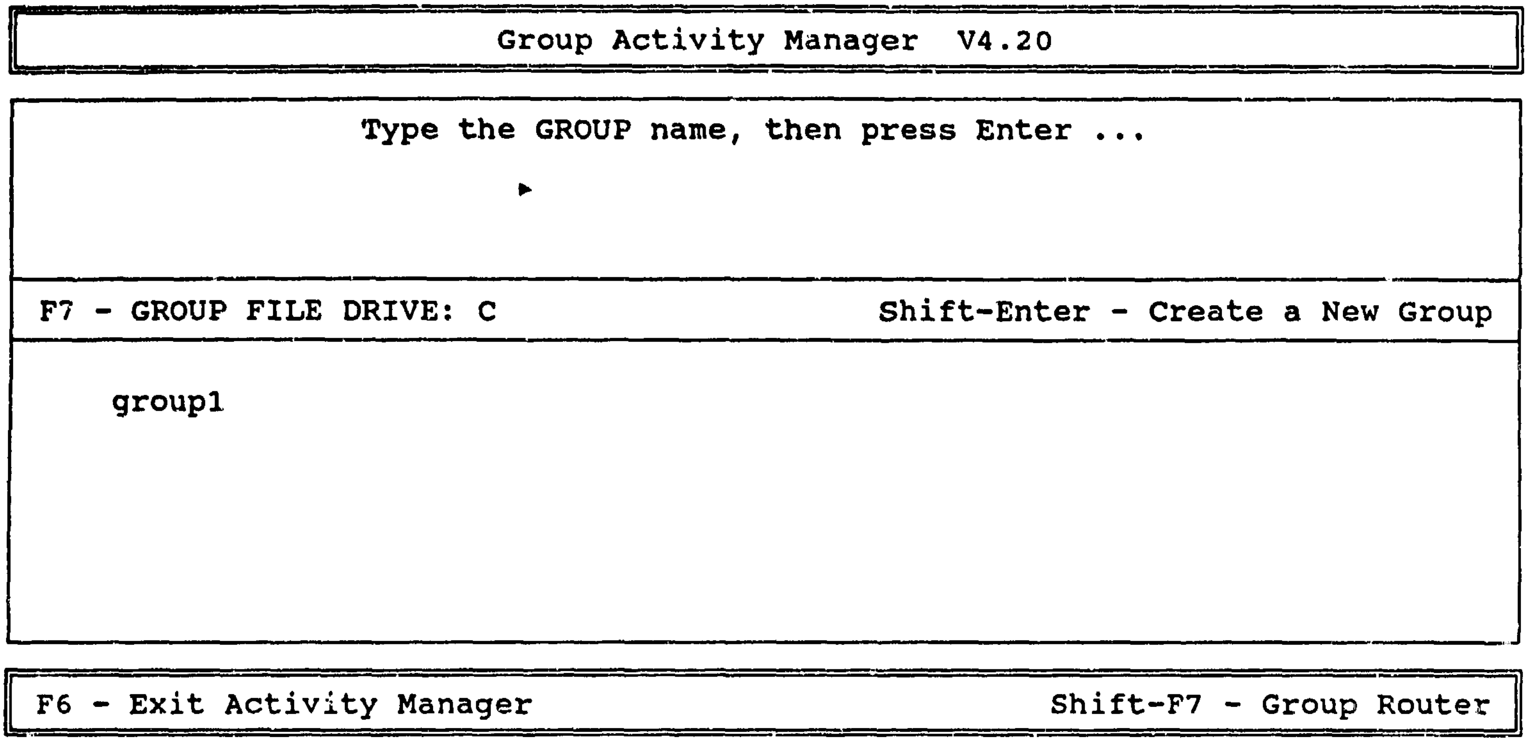
```
                          File Operations Menu   V4.20

 Drive C           more files...PgDn         1  Choose directory file types
                                              2  See information about a file
  att        dis     21652    6-21-89        3  Create a file
  attdeb     dis     33540    6-21-89        4  Lengthen or shorten a file
  authlib    bin     14080   11-19-90        5  Copy a file
  author     bin     38656   10-10-90        6  Rename a file
  author     com      2016    7-23-91        7  Delete a file
  author     src     77568   10-10-90        8  Upgrade Pre-3.00 files
  author2    exe    111753   10-10-90        9  Call a DOS command
  candle     cut       730    7-29-91
  carrolsf   bin     19712   10-04-89
  carrolsf   exe      2610   10-10-90       Choose an option 1..9 ▸
  cdrom      bin      3840   10-04-89       or type one of the following to
  cdrom      exe      3835   10-10-90       select a different drive.
  cedit      bin     82176   10-10-90
  centital   fon      5632    6-21-89         A   C   D   E   F   G   H   I   J
  centmed    fon      5120    6-21-89 ▼       K   L   M   N   O   P   Q   R   S
    147 files,   2752512 bytes free           T   U   V   W   X   Y   Z

                          F9 returns to general options
```

Figure 3. TenCORE File Operation Menu



```
                    Author Lesson Executor  V4.20

File to Execute: ►

C  D  E  F  G  H  I  J  K  L  M  N  O  P  Q  R  S  T  U  V  W  X  Y  Z

     authlib            iedit              route
     author             imgcon             student
     carrolsf           info               system
     cdrom              info2              tc2ascii
     cedit              italics            touch
     demo               letters            utility
     edit               line8x10           videolib
     filer              line8x14
     filing             line8x16
     fsedit             makebin
     fsedit2            manage
     gedit              phone
     gothic             print

                          Press F6 to exit
```

Figure 4. TenCORE Lessons Menu



Group Activity Manager V4.20

Type the GROUP name, then press Enter ...

▶

F7 - GROUP FILE DRIVE: C

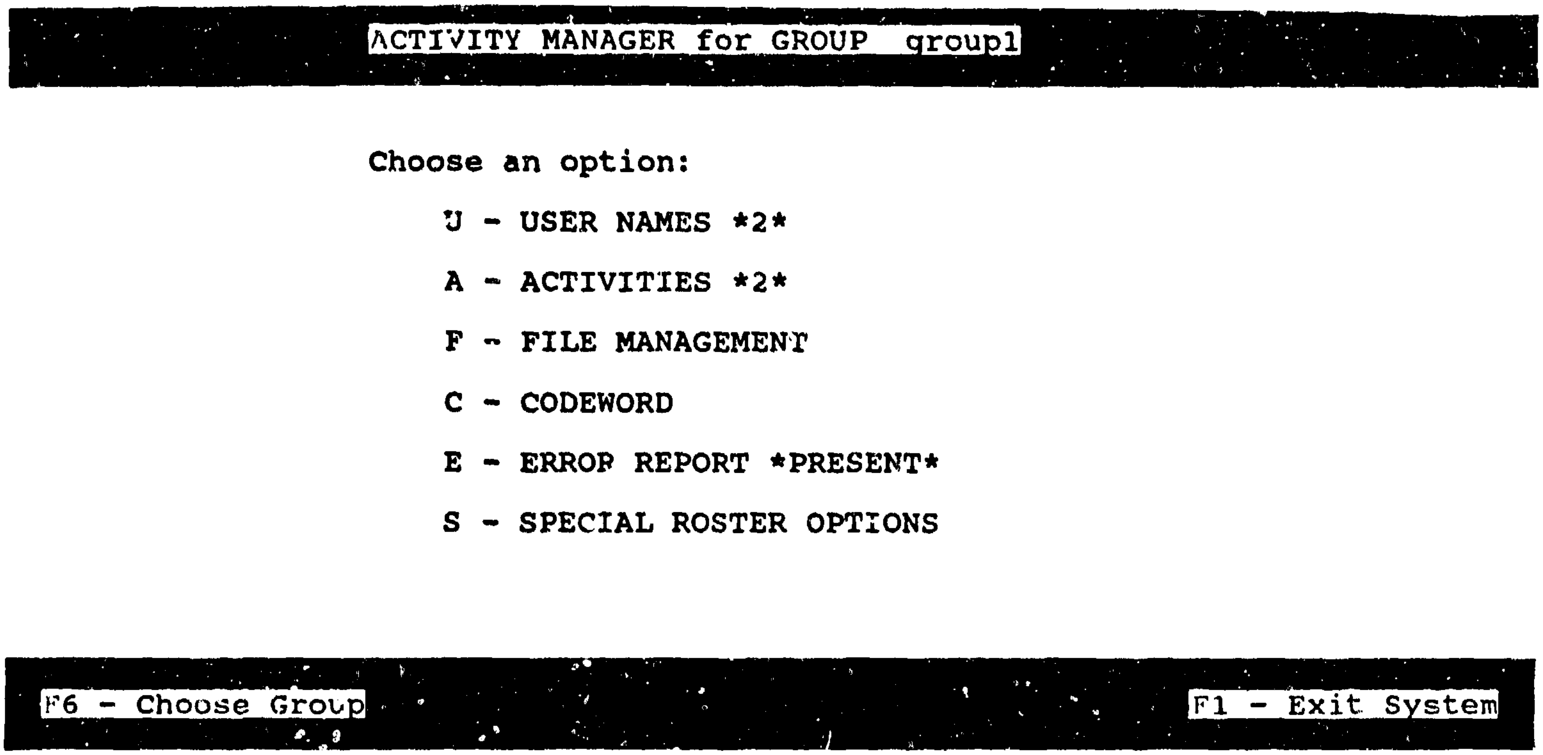
Shift-Enter - Create a New Group

group1

F6 - Exit Activity Manager

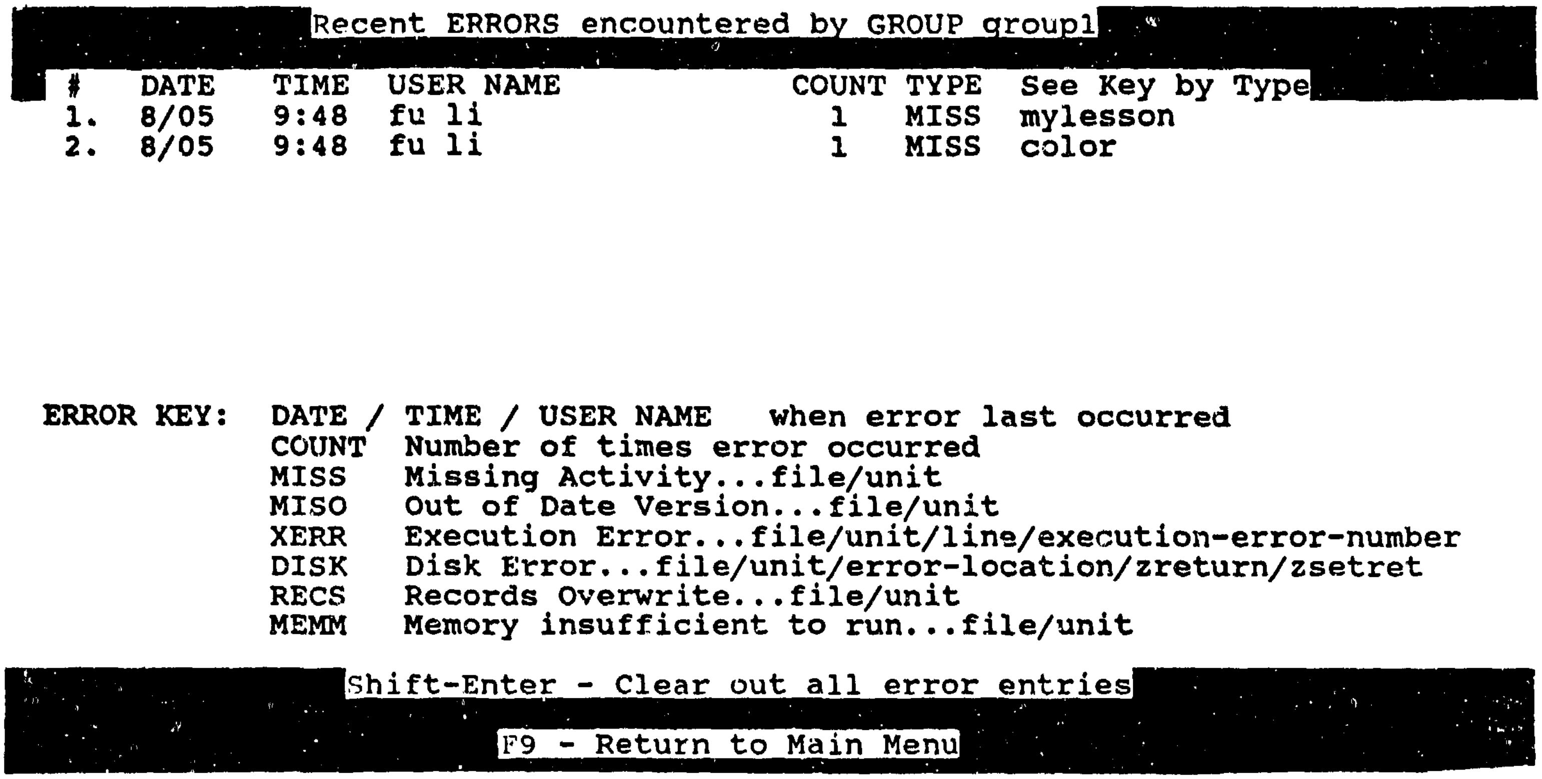
Shift-F7 - Group Router

Figure 5. TenCORE Manage Groups Menu



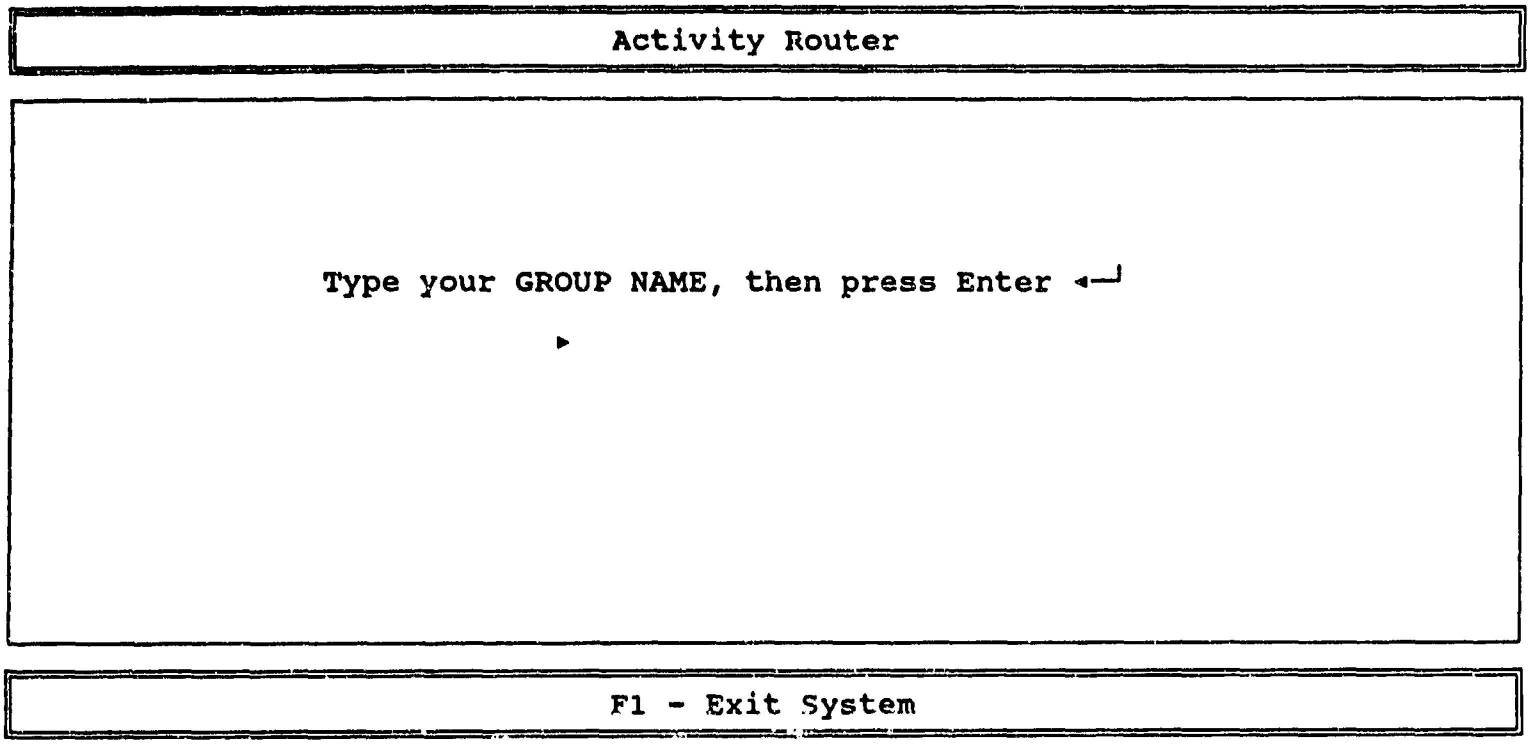
```
ACTIVITY MANAGER for GROUP  group1

Choose an option:

    U - USER NAMES *2*

    A - ACTIVITIES *2*

    F - FILE MANAGEMENT

    C - CODEWORD

    E - ERROR REPORT *PRESENT*

    S - SPECIAL ROSTER OPTIONS

F6 - Choose Group                                F1 - Exit System
```

Figure 6. Functions in the Manage Groups Option



```
                  Recent ERRORS encountered by GROUP group1

 #   DATE    TIME   USER NAME               COUNT TYPE  See Key by Type
 1.  8/05    9:48   fu li                     1   MISS  mylesson
 2.  8/05    9:48   fu li                     1   MISS  color






ERROR KEY:  DATE / TIME / USER NAME   when error last occurred
            COUNT  Number of times error occurred
            MISS   Missing Activity...file/unit
            MISO   Out of Date Version...file/unit
            XERR   Execution Error...file/unit/line/execution-error-number
            DISK   Disk Error...file/unit/error-location/zreturn/zsetret
            RECS   Records Overwrite...file/unit
            MEMM   Memory insufficient to run...file/unit

                  Shift-Enter - Clear out all error entries

                          F9 - Return to Main Menu
```

Figure 7. Error Report of the Activities "Mylesson" and "Color" for the user Fu Li



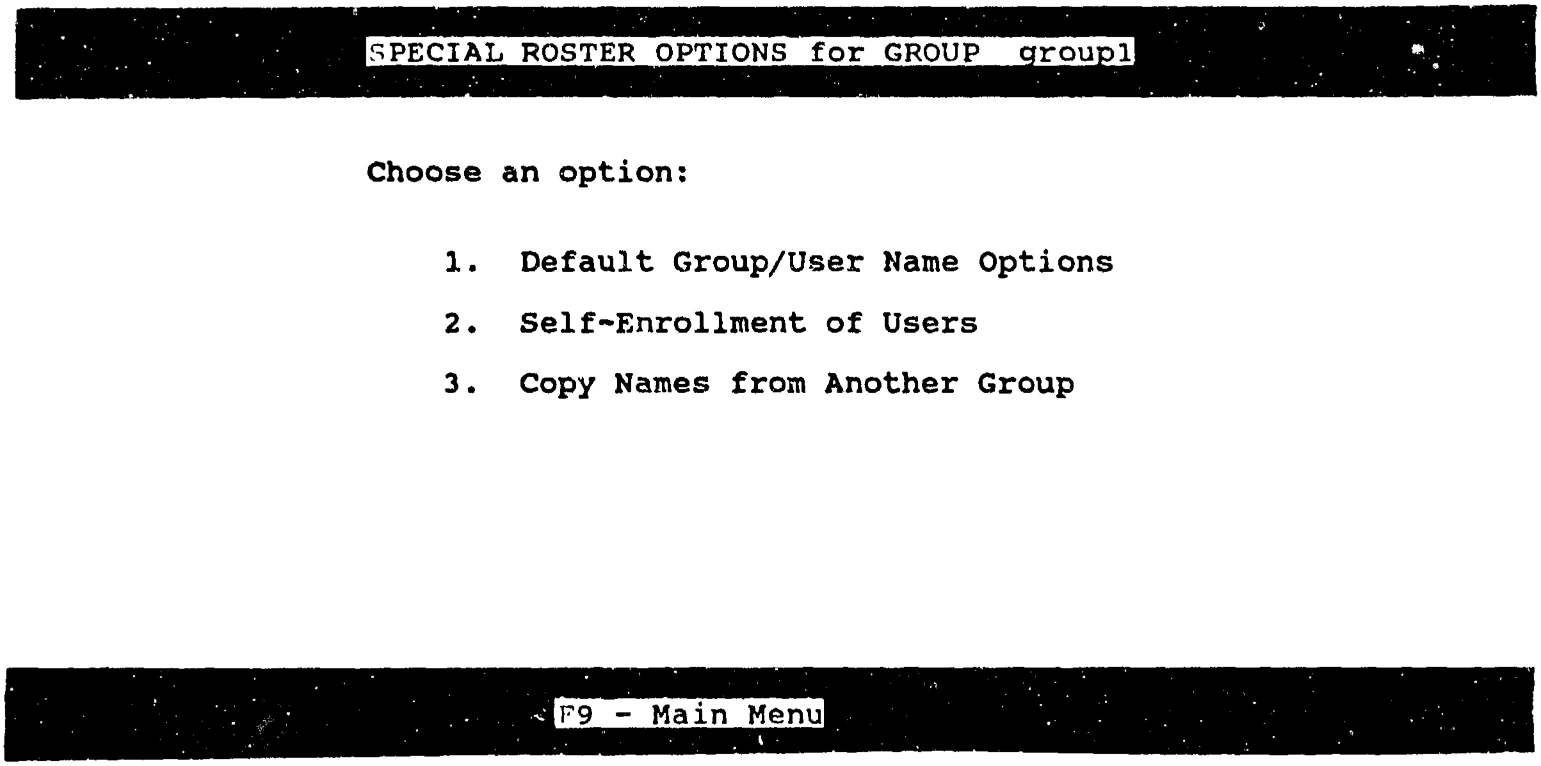
Activity Router

Type your GROUP NAME, then press Enter ↵

▶

F1 - Exit System

Figure 8. TenCORE Group Router Menu



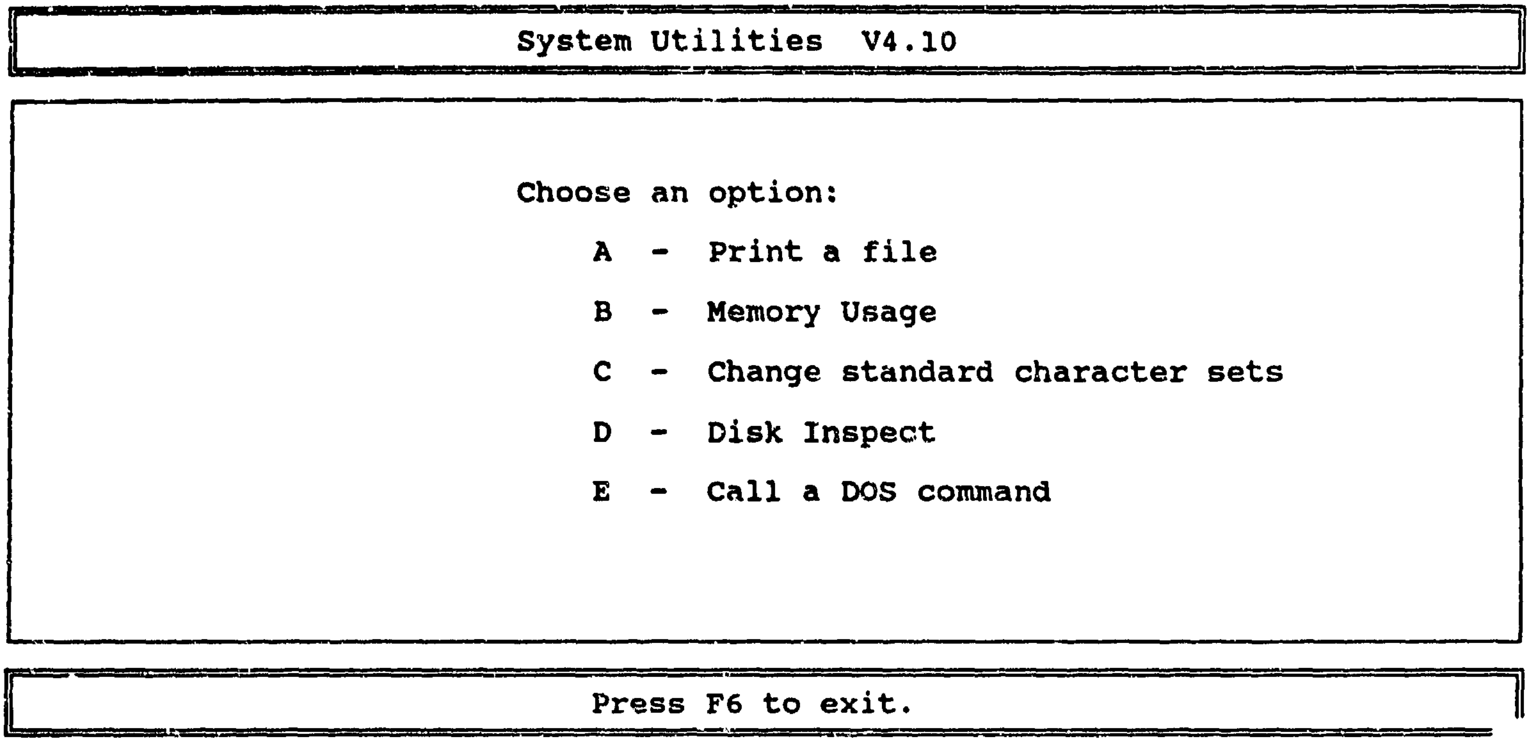
SPECIAL ROSTER OPTIONS for GROUP group1

Choose an option:

1. Default Group/User Name Options
2. Self-Enrollment of Users
3. Copy Names from Another Group

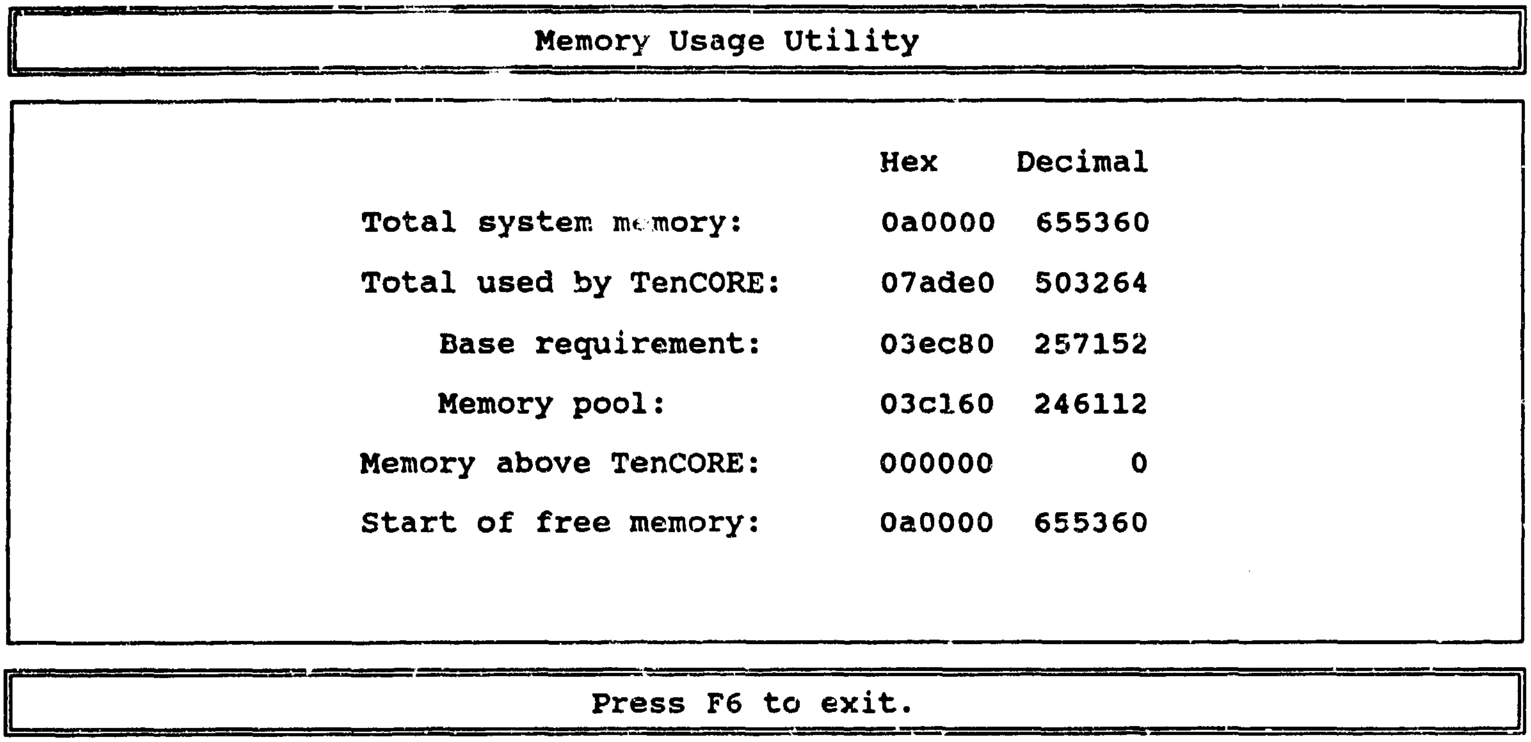
F9 - Main Menu

Figure 9. Functions in the Group Router Option



```
System Utilities  V4.10

Choose an option:

    A  -  Print a file

    B  -  Memory Usage

    C  -  Change standard character sets

    D  -  Disk Inspect

    E  -  Call a DOS command

Press F6 to exit.
```

Figure 10.   TenCORE Utilities Menu



```
                    Memory Usage Utility

                                      Hex     Decimal
        Total system memory:          0a0000   655360
        Total used by TenCORE:        07ade0   503264
            Base requirement:         03ec80   257152
            Memory pool:              03c160   246112
        Memory above TenCORE:         000000        0
        Start of free memory:         0a0000   655360

                      Press F6 to exit.
```

Figure 11. Example of the Memory Usage report

Print out of the source file named "Color" with two units named "Title" and "Factors"

```
-- color / title    -- Edited: 7/29/91   4:15 pm -- Printed: 8/5/91  10:07 am

initial startup
next    factors
*
screen  cga,medium
spacing variable
*
color   green+
at      2:6
box     28,9
*
color   brown+
at      3:7
box     28,9
*
at      4:8
erase   28,9
color   green+
box     3:8 ; 12:36 ; 1
*
chars   narrow
size    2
at      6:12
write   Colors
        and
        Culture
size    1
*
at      18:12
write   Press Enter to continue.

```

```
next    factors
back    title
chars   narrow
spacing variable
color   brown+
at      4:13
write   many factors affect a person's
        color associations.  Some factors
        seem to be physiological, while
        others are definitely cultural.

        For example, look at the circles
        to the left.

        What color would you expect to
        see in the topmost circle?
color   green+
at      8:6
circle  7,fill
color   brown+
at      6:6
circle  7
arrow   15:14
putlow
answer  red
.       write   Many people think so.
.       color   red+
at      17:13
write   Many people expect red.
answer  green
.       color   green+
answer  yellow
.       color   brown+
ok
.       color   red+
endarrow
at      4:6
circle  7,fill
box     25,170;45,110; 1
box     33,110;37, 2
```

15.0 APPENDIX H
Review of Hypercard and HyperTalk

A BRIEF SUMMARY OF
HYPERCARD AND HYPERTALK

by

Tracy A. Dunlap
Planning Systems Incorporated

and

Fu Li
Statcom Inc.

for
US Army Research Institute
5001 Eisenhower Avenue
Alexandria, Va 22333-5600

1.0 INTRODUCTION

HyperCard is a software package produced by Apple Computer, Inc. HypėrCard provides a unique information environment for the Apple Macintosh computer. It can be used to search and store information such as words, charts, pictures, digitized photographs and any other subject that suits the user. Any piece of information in HyperCard can connect to any other piece of information; therefore, the user can find out what he wants to know in as much or as little detail required.

HyperTalk is a language that is built into the HyperCard software. With HyperTalk, the user can write his own instructions, called scripts, for HyperCard to carry out. The user can create, customize, and personalize HyperCard stacks without learning how to write scripts, but scripting with HyperTalk allows even more control over what the computer can do.

2.0 HYPERCARD DEFINITIONS

In order to understand how HyperCard works, it is necessary to know a few basic terms; specifically **stack**, **card**, **background**, **button** and **field**. These terms are referred to as **objects**. Objects are elements that can receive and send commands or act on commands according to instructions written using HyperTalk.

A **stack** is a named collection of cards. Cards in a stack are usually (but not necessarily) based on the same theme. They usually share the same look and contain similar information. An icon for each stack appears in the Finder.

A **card** is an information holder. On a Macintosh Plus or SE, a card fills the entire screen. On larger-screen Macintoshes (like the Macintosh II) and on larger monitors, a card appears in a window. Cards can contain both text and pictures.

The **background** is a common storage area. In other words, the background is a portion of the card/stack that contains pictures, buttons or fields which are on **EVERY** card in the stack.

Buttons can be used to travel through cards within the same stack or among different stacks. In actuality, the buttons "link" the cards and/or stacks together. By clicking buttons, the user can do a lot more than go to other cards and stacks. For example, the user can perform a specific operation.

Fields are the boxes in which text is written.

HyperCard has many tools that the user can implement to look for and create information. **Browsing** is a term for moving through cards and stacks. The **browse tool** is a little hand which moves around whei the user repositions the mouse. The browse tool is the tool used most often.

3.0 OPERATION OF HYPERCARD

When HyperCard begins, the first thing the user sees is the **HOME CARD**, refer to Figure 3-1. It is a visual directory that serves as a starting point for important cards and stacks. It displays basic stack examples that HyperCard can do. A short description for some of these stacks are listed as follows:

- Intro: This is the on-line Introduction that helps the user to learn how HyperCard works and what each of the functions can do.
- Help: This stack briefly explains all of HyperCard's features. It's a useful information source and an excellent example showing the kinds of things that the user can do with HyperCard.
- Address: This stack can hold addresses and phone numbers.
- Art Ideas: This stack contains illustrations that the user can select to include within the cards created.
- Card Ideas: This stack contains alternative card formats that can be used when creating cards.
- Button Ideas: This stack contains alternative button formats.
- Stack Ideas: This stack contains alternative stack formats.

The creation of cards (and stacks) is based on the Macintosh pull down menu system. By using this method, a card can be created in "layers". The card may contain any number of layers, depending on the amount of detail required. The different layers and a brief description of each are as follows:

- Background Field: Text, buttons or pictures, which appear on every card as a given background.
- Shared Fields: Text which appears on more than one, but not on all cards.
- Shared Buttons: Identical function buttons which appear on more that one card -- for example, the Home Button.

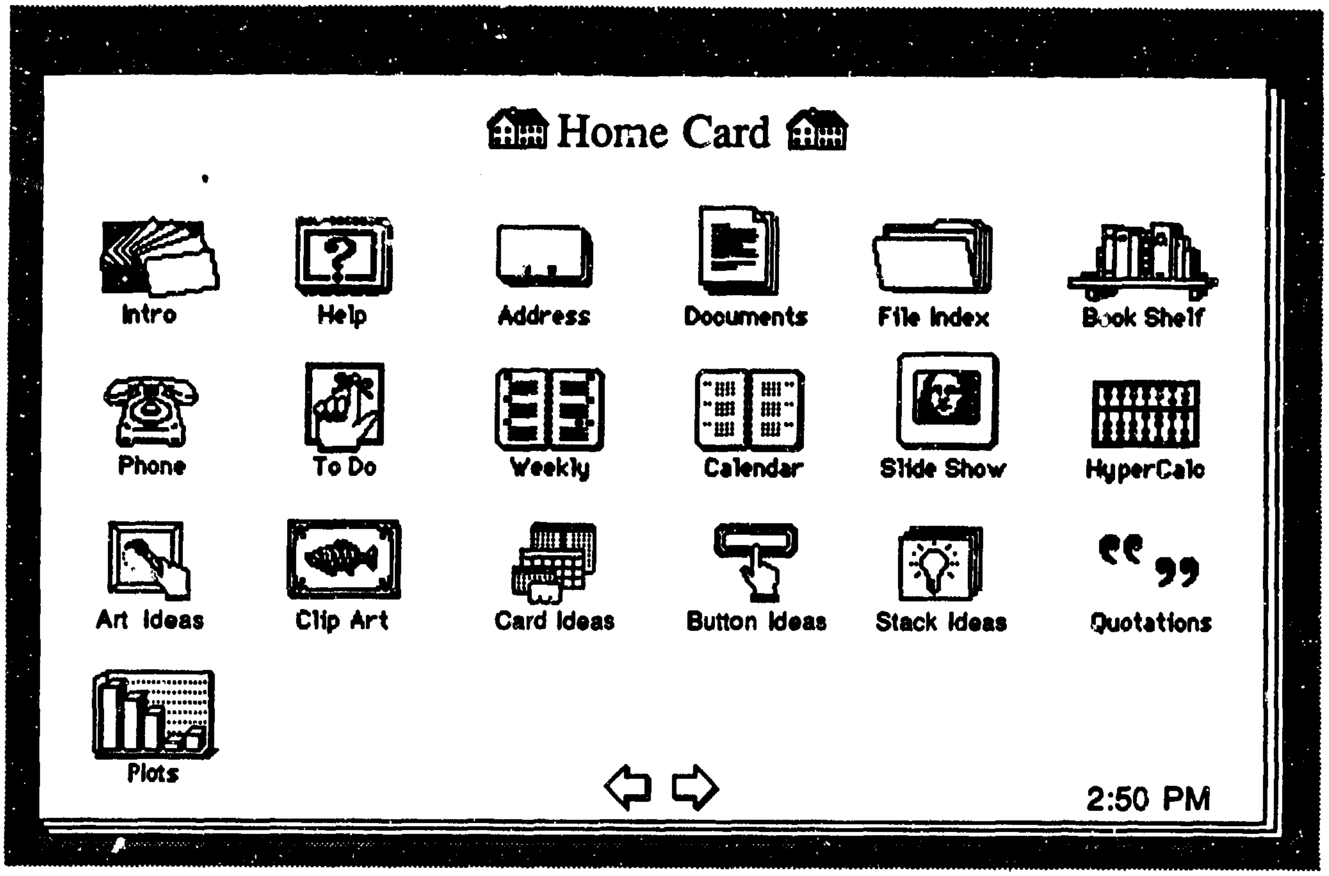


FIGURE 3-1: HyperCard's HOME CARD

- Card Pictures: Graphics which are specific to the individual card.
- Card Fields: Text which is specific to the individual card.
- Card Buttons: Buttons which are specific to the individual card.

Figure 3-2 illustrates a sample card with each of the layers identified.

In order to create, delete, edit or view a card and/or stack, specific user capabilities are required. This is accomplished by accessing one of the five different user levels on the User's Preferences card, refer to Figure 3-3. Each of the levels incorporates everything from the previous level and adds new capabilities. The levels are as follows:

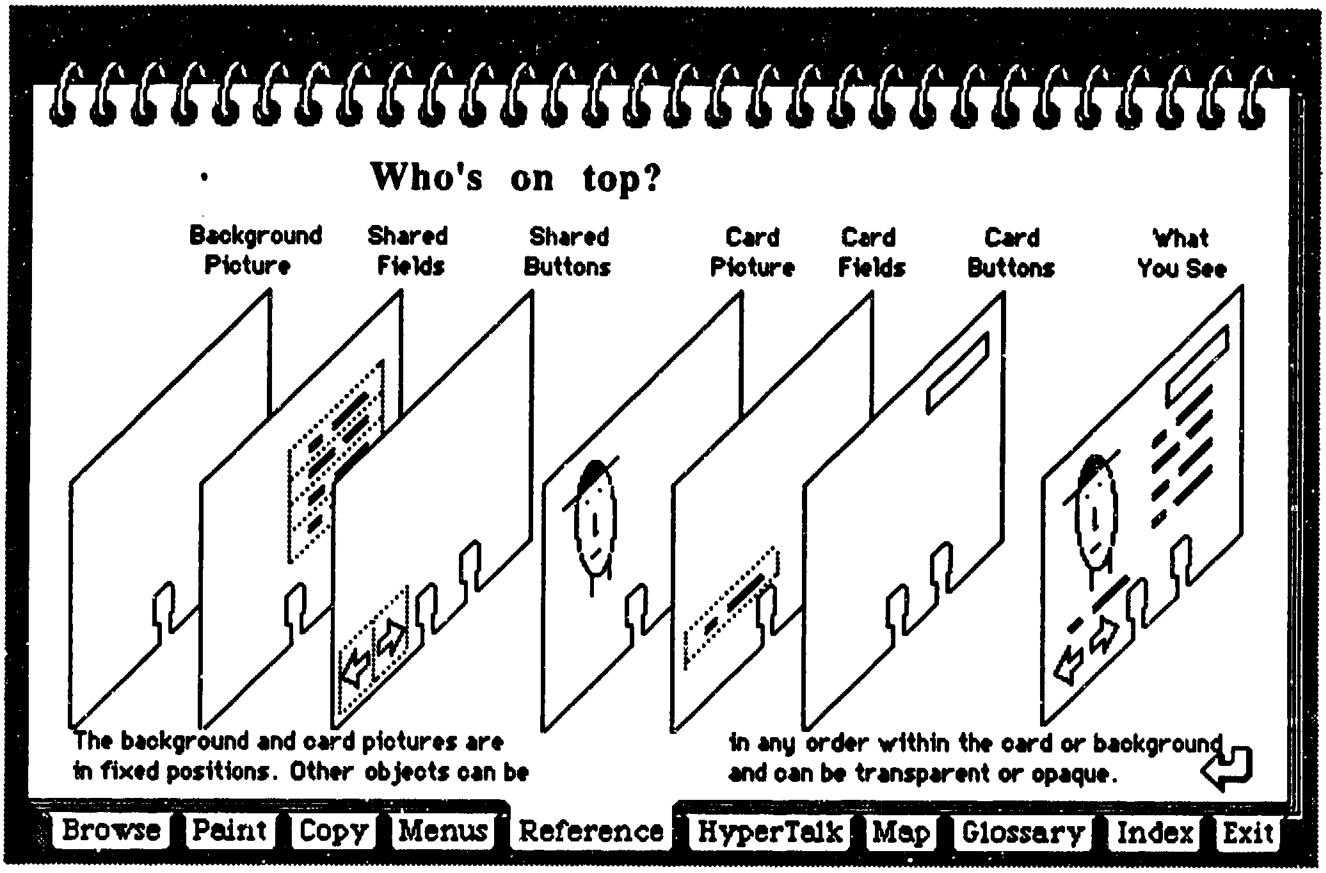
- Browsing: Viewing capability only.
- Typing: Same as Browsing, plus the ability to edit text within existing fields.
- Painting: Same as Typing, plus graphics and extended editing capabilities.
- Authoring: Same as Painting, plus the use of button and field tools, i.e. card/stack creation.
- Scripting: Same as Authoring, plus the ability to write and edit scripts (HyperTalk).

If cards in a stack are created at the same time, they appear in the order of their creation. If a new card is created while the user is within the middle of the stack, that's where the card goes. In short, the cards are "ringed" together; when the end of the stack is reached, the next card is the first card in the stack. Even though cards are ringed together, each can be accessed at different points within the stack through the use of buttons and/or the pull down menu system.

There are many different ways to create, edit or delete cards and stacks. The next few steps illustrate a simple method on how to create a new stack.

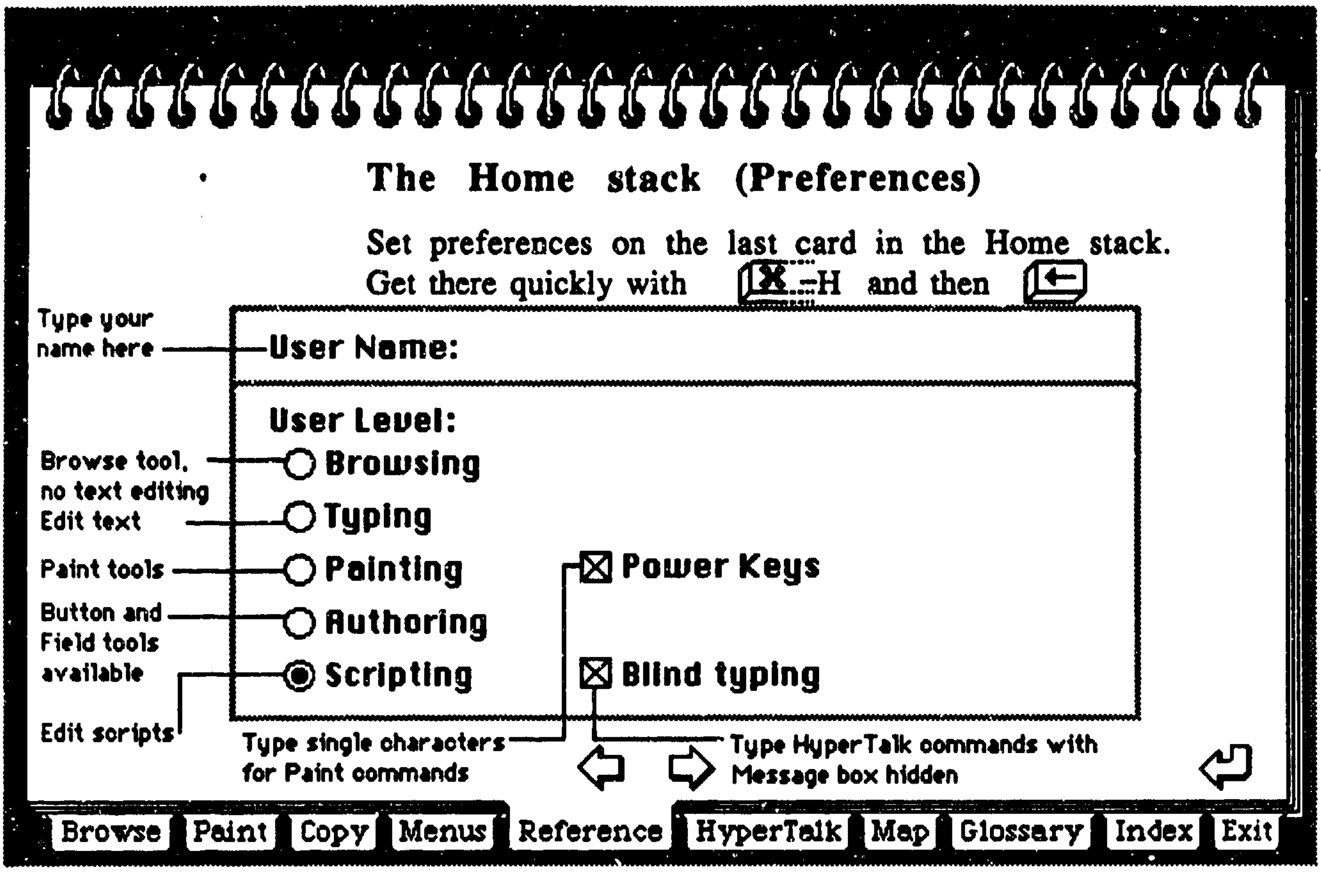
A. Change the user level to AUTHORING on the User Preferences card:

   1) Choose **HOME** from the **GO** menu.
      -- This takes the user to the HOME card within the HOME



The background and card pictures are in fixed positions. Other objects can be in any order within the card or background and can be transparent or opaque.

FIGURE 3-2: HyperCard's Card Layering System



The Home stack (Preferences)

Set preferences on the last card in the Home stack.
Get there quickly with ⌘-H and then ←

User Name:

User Level:

○ Browsing

○ Typing

○ Painting ☒ Power Keys

○ Authoring

● Scripting ☒ Blind typing

Type your name here
Browse tool, no text editing
Edit text
Paint tools
Button and Field tools available
Edit scripts
Type single characters for Paint commands
Type HyperTalk commands with Message box hidden
Browse
Paint
Copy
Menus
Reference
HyperTalk
Map
Glossary
Index
Exit

FIGURE 3-3: HyperCard's User Preferences Card

stack.

2) Choose **LAST** from the **GO** menu.
-- This moves the user to the USER PREFERENCES card, the last card in the HOME stack.

3) Click the button labeled **AUTHORING**.

B. Create the new stack:

1) Choose **NEW STACK** from the **FILE** menu.
-- A dialog box appears asking the user to name the new stack and to check whether the new stack will use the same background format as the current stack. This does not include any of the stack's data, just the stack's template. Refer to Figure 3-4.

2) Click the check box to remove the check mark from **COPY CURRENT BACKGROUND**.
-- The new background will be completely blank.

3) Type in the new stack's name as **PRACTICE STACK** and click **NEW**.
-- The stack is now created. It is ready for the user to modify and fill with information on the new stack's cards.

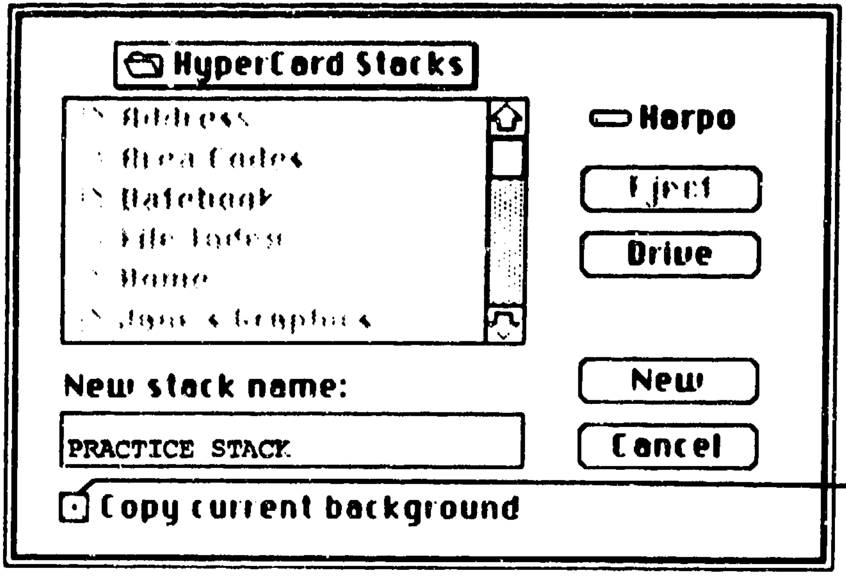
When a new stack is created, it automatically gets three things: the stack itself, a background and the first card. If the **COPY CURRENT BACKGROUND** option was checked, the stack would contain the background pictures, fields or buttons of the card the user was currently on when the **NEW STACK** command was executed. Otherwise, as in this case, the result is a blank card.

4.0 HYPERTALK DEFINITIONS

As mentioned before, HyperTalk is an extension of HyperCard. It allows the user to customize HyperCard's actions, by writing sets of instructions called **scripts**. The user does not need any prior programming experience to use HyperTalk. The only prerequisite is that he should be familiar with HyperCard and to know how to manipulate stacks. The following lists some terminology commonly used in HyperTalk:

- Objects: Any HyperCard element that can receive and send messages. Objects are stacks, cards, backgrounds, fields and buttons.

- Script Editor: A large dialog box containing a window in which the user can type and edit a script. The top line of the script editor



New stack name: PRACTICE STACK

☐ Copy current background

You use the current stack's design for the new stack if this box is checked.

FIGURE 3-4: Dialog Box For A New Stack

identifies the object to which the script belongs. Keyboard commands are used to edit text in the script editor.

- Scripts: A collection of HyperTalk instructions associated with a HyperCard object. The object's script editor is used to add to and revise its script. Every object has a script, even though some scripts are empty; that is, they contain nothing.

- Message: A string of characters sent to an object. Messages that come from the system, such as mouse clicks, keyboard actions or menu commands are called **system messages**. Messages can also be sent from the **Message Box** or from **handlers**.

- Handler: A set of HyperTalk instructions specific to a message. A handler is written in the object's script and contains instructions for HyperCard to carry out when a particular message is received. A handler must begin with the keyword **on** and end with the keyword **end**. Both keywords must be followed by the name of the message.

- Message Box: This is a single line container, that can be used to send messages to HyperCard or to objects. The Message box can also be used to give a one-line command to HyperCard and to search for text. The Message box can be displayed at any time by pressing **Command-M** or by choosing **MESSAGE** from the **GO** menu.

- Object Hierarchy: The order in which a message is passed between objects. For example, when an on-screen button is clicked, the action generates a **mouseUp** system message. The **mouseUp** message always goes first to the button that was clicked. If that button's script doesn't have a handler for **mouseUp**, the message is passed to the card, then to the background, then to the stack, then to the Home stack and finally to HyperCard itself.

- Container: A place in computer's memory where a value can be stored, such as text or numbers. Examples of containers are HyperCard fields, the Message box and variables.

- Variable: An entity that has a changing value. HyperCard has built-in variables such as **it** and **the selection**. A variable can be created to hold some value (either numbers or text) simply by using its name with the **put** command and placing the value into it.

5.0 OPERATION OF HYPERTALK

In order to understand HyperTalk, a brief example illustrating it use is necessary. This example shows the creation of a **Home** button, which is placed onto a card within the stack previously created using HyperCard. The button is then given a command through HyperTalk.

A. Change the user level to SCRIPTING on the User Preferences card:

1) Choose **HOME** from the **GO** menu.
-- This takes the user to the HOME card within the HOME stack.

2) Choose **LAST** from the **GO** menu.
-- This moves the user to the USER PREFERENCES card, the last card in the HOME stack.

3) Click the button labeled **SCRIPTING**.

B. Create a button in the Background:

1) Choose **BACKGROUND** from the **EDIT** menu.
-- The user is now working in the background.

2) Choose the button tool from the **TOOLS** menu, refer to Figure 5-1.
-- The pointing hand or **Browse** tool on the screen changes to an arrow pointer.

3) With the pointer anywhere on the card, hold down the Command key and press the mouse button at the same time. Drag the mouse to create a small square button.
-- The arrow pointer changes to a crosshair. By performing the following actions, a new button is created.

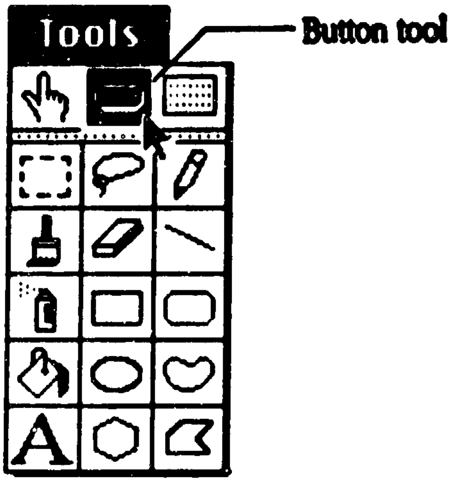


FIGURE 5-1: Choosing The Button Tool

4) Move the button anywhere on the card by dragging its center.
   -- Because the button is in the background, it will appear in its final position on every card in the stack. This way, the user can always go **Home**.

C. Customize the button.

1) Double-click the button to see its **Button Info** dialog box.
   -- The Button Info dialog box tells information about the selected button. Refer to HyperCard Capabilities, Section 6.1 and Figure 5-2.

2) Type **HOME** in the **Button Name** box and don't press return.
   -- This gives the button a name.

3) Click the **AUTO HILITE** check box in order to select it.
   -- The "Auto Hilite" option causes the button to become highlighted when it's clicked.

4) Click the **ICON** button.
   -- Another dialog box appears in which an icon can be selected as a button.

5) Choose one of the house icons shown by clicking and then click **OK**.
   -- All of the dialog boxes disappear. The new button now has the house icon **on** it.

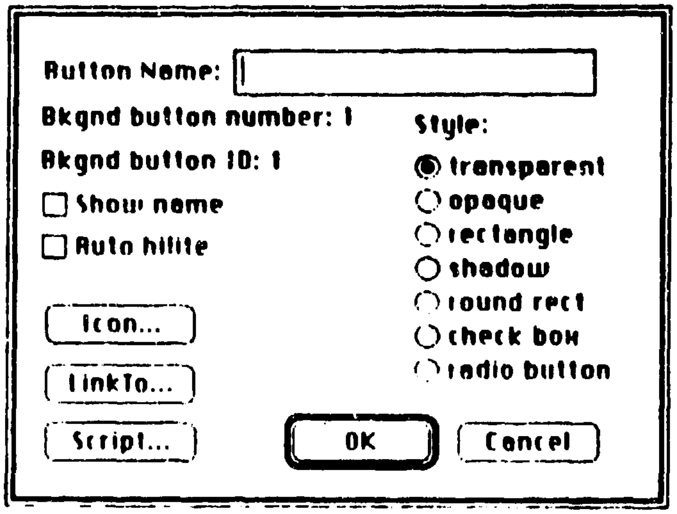
D) Customize the button further by adding a script within the script editor.

1) Double-click the **HOME** button.
   -- The Button Info dialog box appears again. Refer back to Figure 5-2.

2) Click the **SCRIPT** button.
   -- A large dialog box with two lines of text already in the window, appears. This box is the script editor for the **HOME** button. Refer to Figure 5-3.

3) Type **go Home** between the two existing lines and click **OK**.
   -- The button's script is edited.

Figure 5-4 constitutes the completed script for the **HOME** button. These instructions describe what should happen when the **HOME** button is clicked. This code sequence is also known as a



Button Name:

Bkgnd button number: 1

Bkgnd button ID: 1

□ Show name

□ Auto hilite

Style:

● transparent

○ opaque

○ rectangle

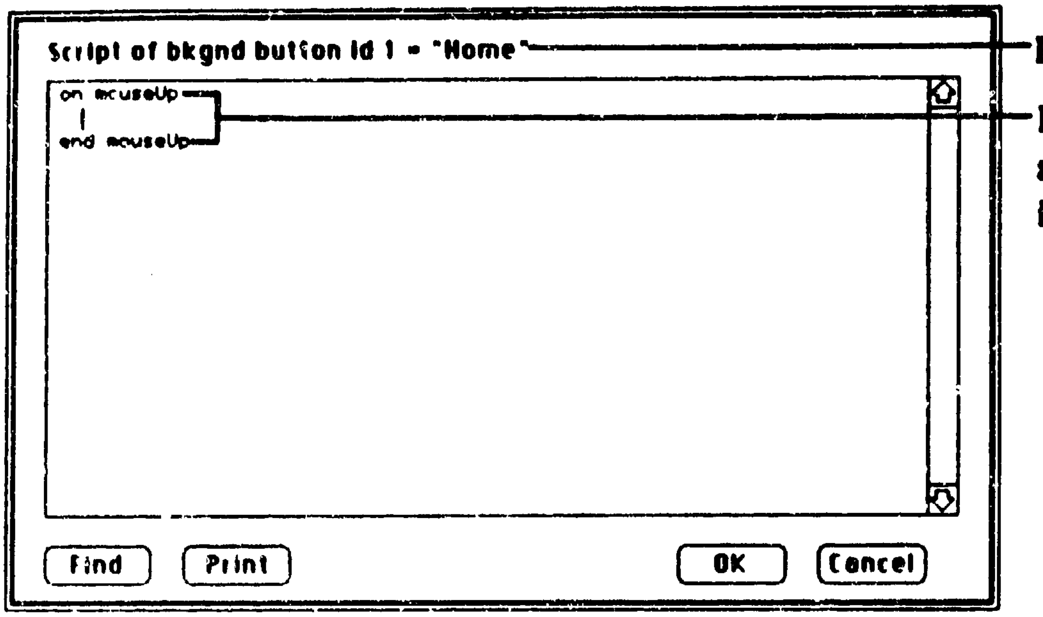
○ shadow

○ round rect

○ check box

○ radio button

FIGURE 5-2: Button Info Dialog Box



Script of bkgnd button id 1 = "Home"

```
on mouseUp
  |
end mouseUp
```

Identification line

Lines that appear automatically in button scripts

FIGURE 5-3: The Script Editor

handler.

```
on mouseUp
  go Home
end mouseUp
```

Figure 5-4: The Completed Script For The **HOME** Button.

Whenever the mouse button is depressed or released, electrical signals are sent. The HyperCard software interprets these signals from the system and translates them into HyperTalk **system messages**.

MouseUp is a system message that means the mouse button has been released; an on-screen HyperCard button receives this message when the mouse button is clicked. Whether something happens when the on-screen button receives the **mouseUp** message depends on whether the button's script contains any instructions for that message. The last line, **end mouseUp**, indicates the end of the instructions. Translated into English, the instructions say:

"When this button is clicked, go to the first card of the Home stack. That's all."

To go back to the stack where the **HOME** button was created, the **Message box** can be used. This is an alternative method for sending messages.

E) To get back to the stack using the Message box method:

1) Choose **MESSAGE** from the **GO** menu or press **Command-M**.
   -- The Message box appears; Figure 5-5.

2) Type **go to stack "Practice Stack"** and press return.
   -- The stack "Practice Stack" appears on the screen.

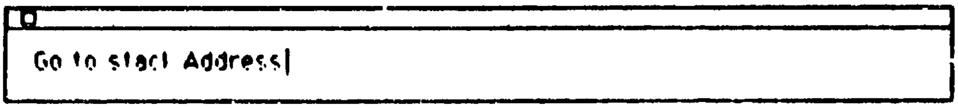
3) Close the Message box by clicking its close box or pressing **Command-M** again.

6.0 HYPERCARD/HYPERTALK CAPABILITIES

6.1 HyperCard Capabilities

The following lists the capabilities of HyperCard:

- On-line Introduction and Help screens are available.
- HyperCard is based on a pull down menu system in order to execute commands.



Go to stack Address

FIGURE 5-5: HyperCard's Message Box

- Stack and card security measures can be established through thé use of passwords and the User Preference card.

- New stacks and cards can be created either from existing or non-existing stacks and cards. Once these have been created, the text, button and graphical editing capabilities are endless.

- It is possible to create graphics directly via the **PAINT** option, to copy graphics from other stacks or cards, or to import graphics from other files which could implement laser-scanning methods. Likewise, it is possible to export graphics into separate MacPaint documents.

- It is possible to compact stacks in order to conserve working space.

- Stacks and cards can be printed out using a wide range of formats.

- Cards may contain any number of common (background) fields, which can appear on other cards within the same stack.

- It is possible to branch or "link" to other cards and stacks through the use of "buttons" which implement system commands.

- It is possible to search through a stack by "keying-in" on a specific text field.

- Buttons can resemble actual buttons or their specific functions may be represented through the use of icons.

- General button information may be brought up at any time. This information includes:
  - Button Name,
  - Show Name Option
    (When this box is checked, the button's name appears on the button.),
  - Button Number
    (Buttons numbered in order of creation.),
  - Button Layer
    (Current layer button is on.),
  - Button Identification Number
    (Unique button id, which is permanent and unchangable),
  - Style Option
    (Allows for different button style options.),
  - Auto Hilite Option
    (When this box is checked, a button is momentarily highlighted when it is clicked.),

- Icon Option
  (Allows the use of an icon as a button.),
- LinkTo Option
  (Creates the link for the button.),
- Script Option
  (Allows access into the button's script editor.).

- General Background information may be brought up at any time. This information includes:
  - Background Name,
  - Background Number,
  - Background Identification Number.

- General card information may be brought up at any time. This information includes:
  - Card Name,
  - Card Number,
  - Card Identification Number,
  - Total Number of Fields on Card; this includes both text and graphic fields,
  - Total Number of Card Buttons.

- General stack information may be brought up at any time. This information includes:
  - Stack Name,
  - Stack Location (drive and directory),
  - Total Number of Cards Within Stack,
  - Total Number of Common Backgrounds Within Stack,
  - Disk Size of Stack,
  - Current Amount of Free Storage on Disk.

6.2 HyperTalk Capabilities

The following lists the capabilities of HyperTalk:

- Any object (stack, card, background, button or field) can contain a script for a particular action. For example, buttons can be instructed to route to other cards, or text within fields can be updated automatically just by using the scripting capability.

- When any object is copied, such as a button or field, the script that the object already has, is also copied.

- HyperTalk can produce special effects in both sight and sound. These effects may include:
  - A **BEEP** sound,
  - An actual song,
  - Specialized visual effects, such as **SCROLL UP/DOWN** or **ZOOM IN/OUT**

- Conditional statements, such as **IF/THEN** and **IF/THEN/ELSE** structures can be used to direct the flow of operations.

- Message Handlers (a set of HyperTalk instructions specific to a particular message received) can be written for objects.

- Calculations can be performed within containers. The calculations may include variables, arithmetic expressions or specific HyperTalk functions.

- Scripts can be written to produce animated effects on the screen. Such examples include moving figures or shuffling through cards.

7.0 HYPERCARD/HYPERTALK LIMITATIONS

7.1 HyperCard Limitations

The following lists the limitations of HyperCard:

- HyperCard will not operate without a **HOME** stack. This stack should never be deleted.

- It is necessary that the **HOME** stack is either in the same folder as HyperCard, or in the HyperCard stacks folder. Otherwise, the user has to specify where the **HOME** stack is each time HyperCard begins.

- The most common mistake in creating a field or button is putting it on the card level when it's supposed to be on the background level. A check to determine which level the object is in, is to flip to the next card after the object has been created. If the object is on that card also, then the object is in the background level.

- If a background text field is deleted, then the text within that field will be irrevocably deleted on every card that uses the field.

- When the user tabs to a field that already contains text, then all of the text will be selected. Typing at this point will wipe out all old text in the field and will be replaced with what is typed. To preserve old text, move the pointer to where to begin typing and click the mouse.

- The **BROWSE** tool, not the **PAINT** text tool (option for graphical text writing) should be used for entering text in fields. Regular text (via **BROWSE** tool) is always editable in an unlocked (unprotected) field. **PAINT** text is not editable once it has been typed in.

- Set text properties, such as bold, underline, font type, etc., affect all text within the field being set. A single word in a field cannot have a particular text property, unless it's the only word in the field.

- Some stacks are designed so that when the stack is locked (security measure), the user can type into a field or paint a picture on a card or background; however, the changes disappear as soon as the user leaves the current card.

- There is no way to find out the stack password once it has been set. The password must never be forgotten!

- Changing the name of a stack will break any links made between any card within the renamed stack and any other card in any stack. Therefore, it becomes necessary to re-establish every link.

- HyperCard has no **SAVE** command. HyperCard automatically saves what was created, whenever the user switches from one card to another. To save the stack under a different name, the **SAVE A COPY** command must be used.

- If the **LANDSCAPE** (horizontal) print option is chosen for a LaserWriter, then the **FAST LASER PRINTING** option must be chosen also. If not, printing a stack will be extremely slow.

- **WITH THE CURRENT BACKGROUND** print option must be checked in order for the column format to work.

- If any part of a picture moves off the screen after implementing the **ROTATION** command, then that part is lost completely. Try **REVERT** to retrieve the lost portion.

- A graphics area, that has been selected using the "lasso" method, should be completely enclosed. If not, then the **INVERT** prints white the unenclosed section of the lassoed area; the whole image seems to disappear.

- It isn't possible to export a button's action. Only the image is what is exported, i.e., only what appears on the screen.

- A maximum of ten screens may be exported into a maximum of ten files, respectively.

7.2 HyperTalk Limitations

The following lists the limitations of HyperTalk:

- If a specific name is within a HyperTalk statement to be acted upon, then the quotation marks around the name should be used. For example,

  go to stack "Home" (operation routes to stack called Home).

  Although it is possible in many cases to omit the quotation marks and still have a working statement, as a general rule it's best to include the marks. Without them, HyperTalk is sometimes able to carry out the command as intended, but not always.

- HyperTalk's visual effects aren't visible when **COLOR** or **MULTIPLE GRAYS** screen options are selected.

- In order to hear sounds, the speaker volume within the control panel must be set to a value greater than 0.

- In HyperTalk, it is necessary to include the **card** or **cd** adjectives in front of the **field** container, in order to specify a card field. If the adjective is left out, then a background field is assumed. Conversely, the **background**, **bkgnd** or **bg** adjectives in front of the **button** container must be used to specify a background button. Otherwise, HyperTalk assumes a card button.

- Abbreviations **cd** and **bg** are not available in HyperCard versions earlier than 1.2.

16.0 APPENDIX I

CRITERION-REFERENCED TESTING
BIBLIOGRAPHY AND TOPICAL OVERVIEWS

J.D. Dressel
US Army Research Institute
5001 Eisenhower Avenue
Alexandria, Va 22333-5600

General

A student's performance on a criterion-referenced test indicates how well that student can perform the well-defined objectives of that test. The objectives of the test could cover a task, a duty position, or a period of instruction. The test content can be broad or narrow; the breadth is not important. What is important is the defining of the test objectives which drive the construction of the test. The test objectives form the domain or body of knowledge/skills of the test, which is the criterion to which the test items are referenced. On the basis of comparison of the student's test score to a minimum standard, the student is either classified as being a "master" (having mastered the content of the test) or a "nonmaster" (lacking the minimal competence required).

The purpose of a criterion-referenced test is to determine if the student has mastered the content area (domain) of the test. For example, assume the test is referenced to a criterion of job skills which form a military occupational speciality (MOS). If the student passes this test, then the assumption is made that he/she has mastered the MOS. No assumption can be made as how this student compares to other students within the class. This is a concern of norm-referenced test which have a different function from criterion-referenced tests. Basically a norm-referenced test will indicate how well a student performed the test as compared to the performance of his/her classmates. A criterion-referenced test will indicate if the student can perform the job (objectives) for which the test was developed.

The following topics of concern in criterion-referenced test development will be presented:

A. Content domain specification

B. Test item writing

C. Test construction/administration

D. Item analysis

E. Standard setting

F. Reliability

G. Validity

Content Domain Specification

Content domain specification refers to determining which content areas, subjects, or tasks should form the body of the test. Test developers call the field of knowledges/performances covered by the test, the domain of the test. The domain is divided into clearly defined objectives and individual test items are written for each objective. Content domain specification basically concerns detailing exactly what the test should test.

Testing for competence for a specific job requires the performance of a front-end analysis. This process would require that observers record the actual tasks the job-holders perform and note the features of acceptable performance during the completion of each task. This group of tasks would therefore define the job. A panel of highly knowledgeable people in this job area could select tasks which are critical for successful job performance. These critical tasks could then be sampled by constructing test item in a uniform/standardized manner for the test.

Developers of tests for an Army military occupation speciality (MOS) have some of this work done for them. Training and Doctrine Command (TRADOC) has lists of critical tasks recorded for each MOS. The test developer could have subject matter experts (SME) examine the list to ensure its current accuracy.

More information will be provided under the heading of Test Item Writing on how to construct individual test items. What is important to remember here is that the domain of the test must be clearly defined and stated. Only in this way can the results of the test be meaningfully interpreted. It must be clear exactly what the successful test-taker as "mastered".

Test Item Writing

There are two principal forms of test items. Student performance on written test items can indicate the student's knowledge underlying task performance while hands-on performance measures can indicate whether the student can perform the task.

Due to physical, time and scoring constraints, most classroom testing employs written tests. There are four common types of written test items. These items are

1) multiple-choice

2) true/false

3) matching

4) constructed response.

By far the most popular written test item is the multiple-choice item. Multiple-choice items consist of a statement or question stem and typically four answer choices or options. There is one correct choice; the remaining three incorrect options are called foils.

Approach the writing of multiple-choice items by following these three general rules:

1) Be sure the question is clearly stated and requires the student to respond.

2) Write the correct choice.

3) Write the foils in the same style as the correct choice.

Some frequent errors in constructing multiple-choice items include:

1) long question stems,

2) grammatically incorrect question stems and options,

3) correct option longer that foils,

4) foils belonging to a set or category different from the correct response,

5) clue in question stem,

6) use of negative/confusing statement.

7) non-random order of correct options.

The test developer can also consider the use of true/false test items. Here a statement is presented and the student judges it to be either true or false. While the probability of correctly guessing a single test item is high (50%) the probability of guessing correctly an entire series of question is quite low; for example correctly guessing the answers to twenty true/false items in a test of thirty items would occure about twice in a million occasions.

Follow these general rules when writing true/false items:

1) a single test item should test a simple idea or bit of information,

2) make positive statements,

3) avoid long statements

4) deal with clear-cut facts not disputable issues.

Matching questions are actually a form of multiple-chice question with more then four possible answers. Matching questions can cover a large topical area very efficiently. The rules for writing multiple choice questions also apply to writing matching questions.

Constructed-response items require the student to recall or create the answer to the question rather than select it from the options presented. There are three forms of constructed-response items:

1) completion: where the student fills in the blanks of a statement,.

2) short essay: where the student writes several sentences on the topic questioned,

3) extended essay: where the student writes extensively creating a position drawing upon an entire unit of instruction to answer the question.

Due to scoring restraints, military test developers, especially for enlisted MOSs at the entry-skill level rarely use the constructed-responses item formats.

Hand-on performance testing can either measure the process (the performance) or the product (what was created) to indicate student's competence or skill. Three methods are frequently used to assess skill:

1) observation

2) checklists

3) rating scales

Observation is used when: the student's responses is either correct or incorrect, the student either achieved the objective or he/she failed, the student bench-pressed 150 pounds or he/she did not. Observation is therefore used when only a single outcome is recorded.

Checklists are used to record the performance on a series of observational responses usually required in a specified sequence.

Rating scales are used to record the performance along a continuum from good to bad, from high to low. Rating scales are used to rate somewhat abstract qualities or characteristics which may vary gradually. Of these three methods of measuring skill, the creatien, use and interpretation of the rating scale requires the greatest care.

In conclusion of this brief overview, it should be noted that regardless of the form, the test item should be developed within context of the domain specification. In order to be of any value, the test item must examine some aspect of the specified topical content.

Test Development

Development of a criterion referenced test is a relatively straightforth operation. While each step must be considered and performed with care, the developer must also consider the relative importance of the test against the time and effort to be spent for its development. For example, developing a test measuring the students' performance on a simple hour of instructional material would not require the same rigorous development as an end-of-course test. The commonly accepted test development steps are presented below. Descriptions of these steps are provided in the appropriate sections of this overview.

1. Define the purpose of the test
2. Review the individual objectives
3. Draft test items to fit the objectives
4. Review of test items by content (SME) and test specialists
5. Edit test items
6. Tryout (field test) test
7. Revise test items
8. Assemble test
9. Select standard
10. Pilot test revised test
11. Prepare administration manuals
12. Collect task item statistics

Item Analysis

Tha first step in conducting an item analysis is to again review the test items to make certain they reflect the content area you wish to examine. This review should be performed by some content matter expert other than the test developer. Any test items that don't seen appropriate after this review should be deleted.

Next, the draft test items should be field tested with one or more groups of students comparable to those student who will take the actual test. The results of this testing will provide information on how well the items function. Three general testing approaches are commonly used.

1) Preinstruction-postinstruction, in which the same group of students take the test, then receive the instruction, and finally are retested after instruction. Students' item performance is compared before and after instruction.

2) Uninstructed - instructed groups, here two different groups of students receive the test. One group has not received the instruction, while the other groups has. Again, the item performance of the groups is compared.

3) Contrasting groups, where the members of each group are individually selected on the basis of either being a master or nonmaster of the content matarial. The two groups take the test and item performance is compared.

Immediately after field testing the items, the tested students can be asked to provide feedback on the test items. Generally the students are asked:

1) Were there any confusing items?

2) Were there any words in the items which you did not know?

3) Was there any difficulty in understanding what you were asked to do?

4) Were there any items without a correct answer?

5) Were there any items with more than one correct answer?

The student feedback can then be considered and used in conjunction with the item statistics (which follow) to revise the test items.

Item difficulty is indicated by the percentage of students who correctly answered the item. The item difficulty index values can range from 0 (an extremely difficult item) to 100 (a very easy item). The difficulty index must be determined for both the instructed and noninstructed students. Difficulty index values can give an indication of the influence of instruction or even the need for instruction.

An item discrimination index refers to how well a test item indicates to which group (instructed, master vs. uninstructed, nonmaster) a student belongs. There are several different forms of this index. However, basically each index operates on the different proportions of students from each group getting the item correct. Index values range from +1.00 to -1.00; a value of +1.00 would indicate that all the instructed students correctly answered the question while none of the uninstructed students' correctly answered the question. A value of +.25 would indicate that 25% more instructed students correctly answered the question than uninstructed students. Any test item with a negative (-) discrimination value (more nonmasters correctly answering the item than masters) should be examined closely.

Another part of item statistics is the choice response analysis for multiple-choice test items. Here the test developer compares the response pattern for each item from both of the field-tested groups (uninstructed, nonmaster vs. instructed, master). If any of the three following conditions do not exist, then the test item probably needs revision.

1) No distractor/foil should receive as many responses from the instructed group as the correct answer.

2) All distractor should receive some (5-10%) responses from each group.

3) Each distractor/foil should be selected by more students from the nonmaster group than from the master group.

Apart from the arena of item statistics, all items should be reviewed to detect any racial, ethnic, sex, or cultural bias. Bais is present if membership in any group would hinder performance regardless of the individual's ability.

Standard Setting

A standard is used to classify students as either having mastered a set of objectives or not having mastered those objectives. A standard therefore represents a point on a scale of performance. Scoring above this point indicates competence while scoring below this point indicates a deficiency. While this concept of a magical point of mastery may seem untenable, it is essential for the decision-making role of criterion-referenced tests. There are many methods for standard setting and all require human judgement. The goal of that judgement is to minimize the incorrect classification of students.

Three factors should be considered when setting standards. Briefly these factors are:

1) Analysis of decision context which considers such things as: the consequences of the decision to fail a student, the opportunities for retesting, the availability or remedial training and the consequences of false classification.

2) Clarity of target competencies which allows standard-setters to determine meaningful minimum competence standards. This is a re-iteration of the purpose of the test: it should provide an unambiguous descrition of the skills being measured.

3) Presence of revelant performance data: refers to pretesting selected groups of students before setting a standard. This would involve administering the criterion-referenced test to groups of uninstructed students, instructed students and previously instructed students. Then from the distribution of group test scores, making a judgement upon where the standard should be set.

Four common approaches to setting standards will be briefly described.

Informed judgment is a method whereby judges set test standards based upon the presence of relevant field test data. A panel of judges arrives at a decision specifying the standard. Borderline group is a method whereby students, who are thought to be "at or near the borderline" regarding competence of the targeted skill, are selected to take the test. The median test performance of this group then becomes the standard.

Constrasting groups is a method whereby two groups of students are selected to take the test. One group consists of students who are judged (by their teachers) to be clearly masters of the targeted skill, while the other group is composed of students who are selected on the basis of clearly not possessing the targeted skill. Frequency distribution curves for the test performance of two groups are plotted and the point of intersection is used to when determining the standard. A method very similar to Contrasting groups in Criterion groups; here, however intact classes of instructed students act as masters while uninstructed groups act as nonmaster for standard setting purposes.

A final approach to standard setting is Nedelsky's method where the individual test items are evaluated. The multiple-choice test are judged on the probability of a minimally competent student guessing the correct answer. The correct by - guessing probabilty of each test item is determined; these item probabilites are then added to generate the total test probability. This could be used as the test standard.

In conclusion it should be noted, that standards should be periodically raviewed. Course content can change, remedial instruction can change, student flow can vary, all of which could prompt a review of the current test standard.

Reliability

Reliability refers to the consistency of measurement. If you carefully measured a board in the morning, upon remeasurment in the afternoon,• evening or next year you would have the same measurement. The concept of test reliability is the same. Without any additional instruction, you would expect a student's test score to be largely the same upon retesting. However, unlike the board measurement, it is highly unlikely that the two test scores will be identical.

Although there are many types of reliability, one is central to criterion-referenced testing. This reliability refers to consistent classification of a student as being either a master or nonmaster of the tested objectives without any additional training/instruction. The simplist form of this index was introduced by Hambleton and Novick (1973). It requires two administrations of the criterion-reference test to the students without any test feedback. The index is simply the sum of the two proportions of students who received the same master/nonmaster classification after each test. For example, 50 students took the test once and about a week later took the test again. Assume, twenty students passed the test both times and twenty-five students failed the test both times. Then the reliability coefficient (^po) equals 20/50 +25/50 or .90.

Other methods which require only one test administration do exist. However these methods (Huynh and Subkoviak) both require more complex statistical computations and assumptions which the test developer may wish to avoid.

Validity

The validity of a criterion-referenced test concerns the accuracy with which the scores from the test can be used to achieve the stated purpose of the test. Validity refers to the appropriateness of the decisions which were made, based upon the test results.

Validity does not refer to the test itself but does refer to the use or interpretation of the test score. Validity is never proven conclusively; however, data (the amount of which is a function of the importance of the test) is collected which can indicate whether the test appears to serve its intended function.

There are three principal types of validity of interest to the test developer. These are item, content, and criterion validity.

Item validity involves comparing each individual test item with the domain specifications or objectives of the test. This comparision is performed by a group of subject matter experts. Any items not clearly matched with an objective are deleted.

Content validity concerns not only item validity but also how well the test item represents the domain/objectives. As a group, the test items should form a representative sample of the specified domain/objectives. Again, a panel of subject matter experts can review the assembled items.

Criterion-related validity concerns how well the criterion-reference test score/classification predict future performance. To be meaningful, these future performances should largely represent actual applications of the criterion behaviors about which the test was created. Criterion-prediction data, using various forms of correlation between test scores and performance, can be gathered in experimental or test development evaluation situations. In either case, the initial nonmasters will require remedial training.

This implies however that ongoing applications of predicting job or future success from test scores is not usually feasible. This would require that all students advance to: the next stage of instruction, graduation, or development (whatever the setting of the predicted behavior) without regard to their past test performance. This would be injurious to the student's education and could be physically dangerous in some situations.

Another form of criterion-related validity which poses fewer hazards, is decision-validity. Decision-validity gives one indication of the accuracy of mastery classification decision based upon test scores and the current standard. Quite simply, it is the sum of the percentages of correct classifications of masters and nonmasters using either the Constrasting groups or the Criterion groups techniques previously described in the standards setting action of this paper.

BIBLIOGRAPHY OF CRITERION REFERENCED TESTING AND EVALUATION METHODOLOGY

1. General

Campbell, C.P., and Allender, B.R. (1988) Procedures for constructing and using criterion-referenced performance tests. *C.V.A./A.C.F.P. Journal*, *23*(3), 2-9.

Campbell, C.P., and Hatcher, T.G. (1989) Testing that is performance-based and criterion-referenced. *Performance and Instruction*, May/June, 1-9.
This article is an overview which describes the rationale and multiple uses of performance-based (hands-on) criterion-reference testing. A description of the development and validation of these tests is presented.

Cantor, J.A., and Hobson, E.N. (1986) *The development of a model for construction of criterion-referenced system achievement tests for the strategic weapon system training program.* Paper presented at the 70th Annual Meeting of the American Educational Research Association. [ED 268 178]
This paper describes a consensus/committee approach to the development of content area, item selection, and cut-score determination as an approach to CRT development within the context of a naval weapons training program.

Cantor, J.A., and Walker, L. (1985) Criterion-referenced testing for the U.S. Navy's nuclear submarine fleet. *Proceedings of the 27th Annual Conference of the Military Test Association, 832-837.* Discusses the procedure for the development of a criterion- referenced testing program currently in place in the naval submarine fleet.

Hambleton, R.K. (1990) Criterion-referenced testing methods and practices. in T.B. Gutkin & C.R. Reynolds (Eds.) *The Handbook of School Psychology*. New York: Wiley.

Hambleton, R.K. (1985) Criterion-referenced assessment of individual differenced. In C. Reynolds & V. L. Wilson (Eds.) *Methodologies and statistical advances in the study of individual differences*. New York: Plenum Press.

Hambleton, R.K. (1978) Criterion-referenced testing and measurement: A review of technical issues and developments. *Review of Educational Research*, Winter, *48*(1), 1-47.

Nasca, D. (1988) An educator's field guide to CRT development and use in objective-based programs. [ED 293 878]
Author attempts tc clarify CRT terminology and offers practitioners a basic view of CRT use and development. An extensive reference list is presented.

Nitko, A.J. (1985) Defining "Criterion Referenced Test". In R.A. Berk (Ed.) A Guide to Criterion-Referenced Test Construction. Baltimore, MD: The Johns Hopkins University Press.

Nitko, A.J., and Pettie, A. (1989) The sixteen quality indicators: standards for evaluating criterion-referenced tests. Paper presented at the Annual Meeting of the American Educational Association. [ED 306 293].
Authors present 16 indices developed to assess the quality of SQTs. Indices support guidance provided in Army doctrine for SQT development. Authors feel these indicators can be used in the development of any CRT.

Rudolph, S.A. (1990) Test design and minimum cutoff scores. Proceedings of the 32nd Annual Conference of the Military Testing Association. 204.209.
Paper discusses a structured approach to test design to assure tests are reliable, valid, and of equal difficulty in retest situations. Approach considers criticality of objectives, number of test items and their difficulty, and the establishment of validity and cut-off scores.

Shaycoft, M.F. (1979) Handbook of criterion-referenced testing. New York, NY: Garland Stmp Press.

Schrock, S., Mansukhani, R., Coscarelli, W., and Palmer, S. (1986) An overview of criterion-referenced test development. Performance and Instruction Journal, August, 3-7.
This article presents the major stages in the design and development of CRTs while noting the differences in construction of these tests and norm-referenced tests.

Wimmer, W.D., and VanLandingham, C.W. (1987) Criterion-referenced testing in the US Army Service Schools. Proceedings of the 29th Annual Conference of the Military Testing Association, 509-512. Paper presents a overview of the importance of testing (CRT) in Tradoc schools/field and provides a listing of military test design references (23) from 1962-1982.

2. Domain of Content Specification, Front-end Analysis

Albert, W.L. (1990) Development of generalized equations for predicting testing importance of tasks. Proceedings of the 32nd Annual Conference of the Military Testing Association, 310-315. A method was developed by which "testing importance ratings" could be assigned to tasks without the expense of SME- conducted survey. Part of the Air Forces automated test outline (ATO) work for test development.

Baker, G.H., and Laabs, G.J. (1988) Issues in job sample testing. Proceedings of the 30th Annual Conference of the Military Testing Association, 571-575. Paper discusses a number of practical and technical issues in the development and administration of a hands-on performance test of work samples (Part of the Joint-Service Job Performance Measurement and Enlistment Standard Project.)

Bart, W.M. (1985) How qualitatively informative are test items?: a dense item analysis. Proceedings of the 27th Annual Conference of the Military Testing Association. A psychometric exposition on the definition and attributes of dense test items. A dense test item indicates why students provide incorrect answers and indicates the sequence of remedial instructional.

Buck, L.S. (1987) Procedures for the development of trade-skill tests. Proceedings of the 29th Annual Conference of the Military Testing Association. Describes the procedures developed and implemented for the production of content valid written and performance tests designed to assess the skill of navy shipyard workers in 17 skill areas.

Distefano, M.K., Pryer, M.W., and Erffmeyer, R.C. (1983) Application of content validity methods to the development of a job related performance rating criterion. Personnel Psychology, 36, 621-631. Notes the procedure used to: a) develop a content valid set of job requirements and b) use of that set of requirements to evaluate prospective worker's performance.

Dittmar, M.J., Hand, D.K., and Phalen, W.J. (1990) Estimating the importance of tasks by direct task factor weighing. Proceeding of the 32nd Annual Conference of the Military Testing Association, 316-321.

Gifford, J.A., and Hambleton, R.K. (1981) Construction and use of criterion-reference tests in program evaluation studies. Academic Psychology Bulletin, 3, 411-436. Technical considerations associated with item selection sampling, and reliability assessment are considered when using a CRT to evaluate the effectiveness of a program of instruction rather than the performance of an individual.

Laabs, G.L., and Baker, H.G. (1989) Selection of critical tasks for Navy job performance measures. Military Psychology, 1(1), 3-16. Describes a method for selecting job tasks which when assembled form the content area for work sample performance test. This hands-on performance test would be used as a "benchmark test" for various predictor tests of job performance. The selection of job tasks involved the use of SMEs, job incumbents, and their supervisors.

Maier, M. H. (1985) On the content and measurement validity of hands-on job performance tests. Proceedings of the 27th Annual Conference of the Military Testing Association, 311-316. Paper examines the content and measurement validity of prototype hands-on performance tests for three Marine Corps specialties. Research is part of the Job Performance Measurement Project.

Mann, W.G. (1989) External validation of job analysis results. Proceedings of the 31st Annual Conference of the Military Testing Association, 205-208. Author notes the frequent lack of predictive validity of CRTs which are purported to be content valid. Performed research showing an r=.78 between measures of content and predictive validity.

Phalen, W.J., Albert, W.G., Hand, D.K., and Dittmar, M.J. Estimating testing importance of tasks by direct task factor weighing. Proceedings of the the 32 Annual Conference of the Military Testing Association, 316-321. Paper presents a possible procedure to select tasks for inclusion into an automated task-data-based outline for the development of Air Force Specialty Knowledge Tests (SKTs). Procedure had SMEs rate the importance of each of the seven factors which would then be used to rate specific tasks for possible inclusion into the testbed.

Popham, W.J. (1984) Specifying the domain of content or behaviors. In R.A. Berk (Ed.) A Guide to Criterion-Referenced Test Construction. Baltimore, MD: The Johns Hopkins University Press.

3. Test Item Writing

Millman, J., and Westman, R.S. (1984) Computer-assisted writing of achievement test items: toward a future technology. Journal of Educational Measurement, Summer, 26(2), 177-190.

Roid, G.H. (1984) Generating the test items. In R.A. Berk (Ed.) A Guide to Criterion-Referenced Test Construction. Baltimore, MD: The Johns Hopkins University Press.

Roid, G.H., and Haladyna, T.M. (1982) A Technology for Test-Item Writing. New York: Academic Press.

Vineberg, R., and Joyner, J.N. (1985) Simulation of hands-on testing for Navy machinist's mates. Proceedings of the 27th Annual Conference of the Military Testing Association, 323-326. Discussed the rationale and developmental concerns for constructing a simulated (written) test in lieu of a hands-on performance test.

4. Test Construction and Administration

Buck, L.S. (1987) Procedures for the development of trade-skill tests. Proceedings of the 29th Annual Conference of the Military Testing Association, 380-384.

Millman, J. Individualizing test construction and administration by computer. In R.A. Berk (Ed.) (1984) A Guide to Criterion-Referenced Test Construction. Baltimore, MD: The Johns Hopkins University Press.

5. Item Analysis

Kalisch, S.J. Jr (1989) Use of item response patterns to predict examine performance. Proceedings of the 31st Annual Conference of the Military Testing Association, 163-166. Author presents the case for using item-response patterns on items to increase the efficiency of testing in both adaptive testing (appropriate branching) and non-adaptive testing (termination of session) situations.

Rushano, T., Williams, J.E., and Stanley, P.P. Item content validity: its relationship with item discrimination and difficulty. Proceedings of the 32nd Annual Conference of the Military Testing Association, 386-391. Paper describes the relationship between item content validity ratings to item discrimination and difficulty. Part of Air Force Specialty Knowledge Test (SKT) development.

Seddon, G.M. (1987) A method of item-analysis and item-selection for the construction of criterion-referenced tests. British Journal of Educational Psychology, 57, 371-379. The author presents a method (which uses the basic item statistics of point biserial correlation, mean, and standard deviation) for selecting individual test items from the domain which minimizes sampling error. The theoretical basis and an empirical application are presented.

6. Standard Setting

Arabian, J.M., McHenry, J.J., and Wise, L.L. (1988) Synthetic validation procedures for identifying selection composites and cut scores. Proceedings of the 30th Annual Conference of the Military Testing Association, 434-439.

Berk, R.A. (1986) A consumer's guide to setting performance standards on criterion-referenced tests. Review of Educational Research, Spring, 56(1), 137-172.
This review provides a trilevel categorization process for 38 methods of standard setting. Ten criteria (Technical/Practical) for method evaluation are presented. The review is intended to facilitate the selection of the suitable method appropriate for the intended application.

Cantor, J.A., and Hobson, E.N. (1986) The development of a model for construction of criterion-referenced system achievement tests for the strategic weapon system training program. Paper presented at the 70th Annual Meeting of the American Educational Research Association. (ED 268-178)

Kulik, C.C., Kulik, J.A., and Bangert-Drowns, R.L. (1990) Effectiveness of mastery learning programs: a meta-analysis. Review of Educational Research, Summer, 60(2), 265-299.
This review is broader in scope than the 1986 review and does not yield and findings germane to the area of CRT.

Kulik, C.C., and Kulik, J.A. (1986) Mastery testing and student learning: a meta-analysis. Journal of Educational Technology Systems, 15(3), 325-345.
Review indicates that end of course scores are highly influenced by criterion level of performance required to progress through course. Implications are evident for setting high cut scores for CRT quizes within course.

Pettie, A.L. (1985) Standard setting methods for skill qualification tests. Proceedings of the 27th Annual Conference of the Military Testing Association, 391-394.
Notes earlier work on standard setting approaches for the SQT, (largely superseded by author's 1987 MTA paper).

Pettie, A.L., and Brittain, C.V. (1987) Establishing minimum pass scores for Skill Qualification Tests. Proceedings of the 29th Annual Conference of the Military Testing Association, 391-394. Paper notes several approaches to setting minimum passing scores (MPS) on the SQT which has became largely a paper and pencil multiple choice test. Method used in FY 87.

Rudolph, S.A. (1990) Test design and minimum cutoff scores. Paper presented at the 32nd annual meeting of the Military Testing Association (Abstract only: 401).

Walker, C.L., and Cantor, J.A. (1987) Alternative performance standard methodologies: a comparison of results on a strategic weapon system (SWS) missile technical C-R SAT. Proceedings of the 29th Annual Conference of the Military Testing Association, 498-503.
Paper explains the procedures of using three different approaches to set performance standards for a naval weapons training program.

7. Reliability

Arabian, J.M., McHenry, J.J., and Wise, L.L. (1988) Synthetic validation procedures for identifying selection-composites and cut scores. Proceedings of the 3oth Annual Conference of the Military Testing Association, 434-439.
Presents a procedure for identifying selection-composites for military occupational skills candidate placement. Notes efficiency in cost savings of analysis and personnel savings as a function of accurate setting of standards.

Kane, M.T. (1986) The role of reliability in criterion-referenced tests. Journal of Educational Measurement, Fall, 23(3), 221-224.
This article focuses upon the importance of a minimal level of internal reliability in a CRT. Generally, any CRT with a reliability of less than .5 should be seriously questioned.

Raju, N.S. (1982) The reliability of a criterion-referenced composite with the parts of the composite having different cutting scores. Educational and Psychological Measurement, 42, 113-129.
The author proposes a method of determining the internal consistency (reliability) for a CRT having subtests with different cut scores. This is not a measure of the reliability of the mastery classification.

Subkoviak, M.J. (1988) A practicner's guide to computation and interpretation of reliability indices for mastery tests. Journal of Educational Measurement, 25(1), 47-55.
This article provides guidance on the meaning, computation, and use of the coefficients of agreement and kappa to determine CRT reliability. Tables are presented which allow these coefficients to be determined from a single test administration.

8. Validity

Brittain, C.V., and Vaughan, P.R. (1987) A comparison of hands-on and written common task test (CTT) scores. Proceedings of the 29th Annual Conference of the Military Testing Association, 385-128.
Notes the relationship between a hands-on performance test of 17 basic soldiering tasks and an alternative paper and pencil test of these tasks. Notes lack of comparability of several task tests.

Buck, L.S. (1989) Are performance tests necessary? Proceedings of the 31st Annual Conference of the Military Testing Association, 123-128.
Focus of research is evaluation of the contribution of written and performance tests to the assessment of a job-incumbents abilities. Used Navy personnel from 17 shipyard trades.

Campbell, C.H., and Campbell, R.C. 1990 Job performance measures for Non-commissioned officers. Proceedings of the 32nd Annual Conference of the Military Testing Association, 541-596. Developed a test battery to measure performance in three job components (supervisory, common, MOS-specific) via three measurement modes (written, hands-on, and ratings). Measurement instruments developed for nine military occupational skills (MOS); intercorrelation of test mode results presented.

Carretta, T.R. (1988) Cross-validation of an experimental pilot selection and classification test battery. Proceedings of the 30th Annual Conference of the Military Testing Association, 559-564.
Use of the basic attributes tests (BAT) in conjunction with the standard Air Force Officer Qualifying Test (AFOQT) will allow a predetermination of flying specialty (fighter vs non-fighter) prior to, rather than following the 52 week undergraduate training program.

Doyle, E.L., and Campbell, R.C. 1990 Hands-on and knowledge tests for the Navy radioman. Proceedings of the 32nd Annual Conference of the Military Testing Association, 529-534.

Paper notes the development, administration and results of a benchmark hands-on performance tests which would guide the development of written tests which could be used as substitute measures of hands-on job proficiency.

Heneman, R.L. (1986) The relationships between supervisory rating and results-oriented measures of performance: a meta-analysis. Personnel Psychology, 39, 811-826.
Results of a meta-analysis found a low correlation between supervisors' rating of worker performance and objective measures of workers' performance. Author notes limitations of study and advocates the use of composite (multiple rating items) ratings and a relative, rather than absolute, rating format.

Intano, G.P., and Howse, W.R. Validation of the Army aviation classification process. Paper presented at the 32nd annual meeting of the Military Testing Association (abstract only: 404).
Paper discusses the development of a test battery to assign novice rotary wing pilots to one of four aircraft for combat training.

Maier, M. H. (1985) On the content and measurement validity of hands-on job performance tests. Proceedings of the 27th Annual Conference of the Military Testing Association, 311-316.
Paper examines the content and measurement validity of prototype hands-on performance tests for three Marine Corps specialties. Research is part of the Job Performance Measurement Project.

Vineberg, R., and Joyner, J.N. (1985) Simulation of hands-on testing for Navy machinist's mates. Proceedings of thee 27th Annual Conference of the Military Testing Association, 323-326.
Discussed the rationale and developmental concerns for constructing a simulated (written) test in lieu of a hands-on performance test.

Williams, J.E., Stanley, P.P., and Perry, C.M. 1990 Implementation of content validity ratings in Air Force promotion test construction. Proceedings of the 32nd Annual Conference of the Military Testing Association, 235-240.
Paper reviews the historical issues concerning item content validity ratings and integration of these ratings into current test development procedures. This Air Force research is part of the current Specialty Knowledge test (SKT) program.

17.0 APPENDIX J
Demonstration Program (DEMPRO)
for Performance Support Technology (PST) System

Demonstration Tutorial: Using "Examiner" to Create Self Development Tests (SDT)

WELCOME TO A DEMONSTRATION

JOB AID FOR CREATING

MOS TESTS

CHOOSE THE KIND OF TEST DO YOU WANT TO DEVELOP.

1. PRACTICE EXERCISES FOR A COURSE ANNEX.

2. END - OF - ANNEX TEST.

3. END - OF - COURSE TEST.

4. SELF-DEVELOPMENT TEST (SDT).

Demonstration Tutorial: Using "Examiner" to Create Self Development Tests (SDTs)

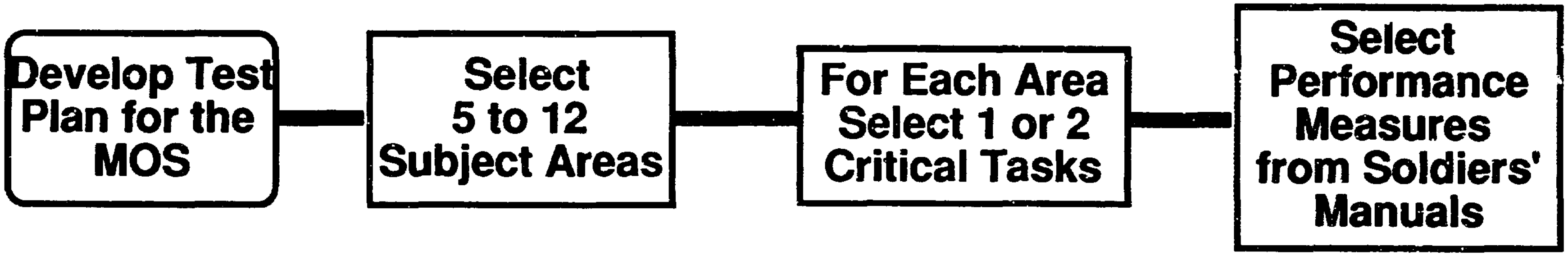
WELCOME TO THIS DEMONSTRATION TUTORIAL

ITS PURPOSE IS TO SHOW HOW TO CREATE SELF DEVELOPMENT TESTS (SDTs).

WE'LL EXPLAIN HOW TO PLAN THE TEST (WRITE A TEST PLAN), THEN LET YOU PRACTICE.

NEXT WE'LL EXPLAIN HOW TO WRITE ITEMS, THEN LET YOU PRACTICE .

Demonstration Tutorial: Using "Examiner" to Create Self Development Tests (SDTs)


Develop Test Plan for the MOS
Select 5 to 12 Subject Areas
For Each Area Select 1 or 2 Critical Tasks
Select Performance Measures from Soldiers' Manuals

Example of part of a plan from soldiers' manuals for MOS 76C & 76Y, Tables of Contents

SUBJECT AREA	TASK	PMs
1. Requesting Supplies	101-521-1133: Use of ARMS	PM 2
2. Manual PLL	101-539-1111: Repair Parts	PM 4
	101-539-1101: Maintain PLL	PM 3
3. TAMMS	101-539-1301: Maint. Con Log	PM 1

NOW PRACTICE ENTERING THIS PARTIAL TEST PLAN INTO EXAMINER

LIGHT PEN THE SQUARE WHICH SAYS 'EXAMINER'.

EXAMINER

Demonstration Tutorial: Using "Examiner" to Create Self Development Tests (SDTs)

Student shifts to Examiner and enters the 2 subject areas and 4 tasks for the sample test plan.

Then the student shifts back to the tutorial.

Demonstration Tutorial: Using "Examiner" to Create Self Development Tests (SDTs)

YOU'VE PUT PART OF A TEST PLAN INTO EXAMINER.

NOW PRACTICE WRITING SOME ITEMS FOR ONE OF THE TASKS IN YOUR PLAN.

BUT FIRST, FOUR BASIC RULES FOR WRITING MULTIPLE CHOICE ITEMS:

1. Test performance ability, not rote recall of facts.
2. Don't give away the answers.
3. Be fair: no curve balls, no trick questions, please.
4. Be clear: 9th grade reading level is about right.

NOW LET'S TAKE THESE ONE AT A TIME.

"Demonstration Tutorial: Using "Examiner" to Create Self Development Tests (SDTs)

RULE 1. TEST PERFORMANCE ABILITY, NOT ROTE RECALL OF FACTS

HERE'S HOW:

1. Ask soldier to analyze a job -related problem.

Example:

You're inspecting a 2 1/2 ton truck. You find the headlights don't work. Which of the following status symbols would you put on DA Form 2404 (Equipment Inspection and Maintenance Worksheet)?

A. -

B. /

C. X

D. ⊗

Demonstration Tutorial: Using "Examiner" to Create Self Development Tests (SDTs)

RULE 1. TEST PERFORMANCE ABILITY, NOT ROTE RECALL OF FACTS

HERE'S HOW:

2. Ask soldier to get information or solve a problem

Example:

Using the Harlem, Ga. map sheet and grid coordinate scale find the 8-digit coordinates for ARRINGTON CEMETARY in grid square 8180.

A. LG 81518051

B. LG 81648064

C. LG 81758080

D. LG 81808070

Demonstration Tutorial: Using Examiner to Create Self Development Tests (SDTs)

RULE 1. TEST PERFORMANCE ABILITY, NOT ROTE RECALL OF FACTS

HERE'S HOW:

3. Ask soldier to identify a critical part of a task

Example:

Your unit has been operating in a very cold area.
You are not in a chemical/toxic environment

What first aid should you give to a soldier's frostbitten face?

A. massage it with your hands

B. cover it with your hands

C. heat it near an open fire

D. Apply a very warm object to it.

Demonstration Tutorial: Using "Examiner" to Create Self Development Tests (SDTs)

RULE 1. TEST PERFORMANCE ABILITY, NOT ROTE RECALL OF FACTS

HERE'S HOW:

4. Ask the soldier how to fix or prevent a malfunction.

EXAMPLE:

You are doing operator maintenance on an M16A1 rifle. You see NO dirt or corrosion in the vent holes. Which procedure would you skip?

A. clean firing pin hole

B. remove bolt carrier and bolt

C. remove handguards

D. clean carrier key.

Demonstration Tutorial: Using "Examiner" to Create Self Development Tests (SDTs)

RULE 1. TEST PERFORMANCE ABILITY, NOT ROTE RECALL OF FACTS.

HERE'S HOW:

5. Ask a what or when question

Example:

Given an M17 series protective mask and the order is given to mask, what step should you take first:

A. stop breathing and close your eyes

B. open the mask carrier with your left hand

C. remove your headgear and place it in a conventional location

D. grasp the mask just below the eyepiece.

Demonstration Tutorial: Using Examiner to Create Self Development Tests (SDTs)

RULE 2. DON'T GIVE AWAY THE ANSWERS .

HERE'S HOW:

1. **Make the incorrect alternatives believable and logical. Avoid ridiculous choices.**

Example of a bad question:

As a PLL Clerk, when would you do a review of the PLL?

A. Whenever I could.

B. Every 60 days.

C. Whenever I felt like it.

D. Every 90 days

Demonstration Tutorial: Using "Examiner" to Create Self Development Tests (SDTs)

RULE 2. DON'T GIVE AWAY THE ANSWERS .

HERE'S HOW:

2. Situations and illustrations for one item should not give away answers to other items.

Example: (The following item gives away the answer to preceding item.)

As a PLL clerk, you do a quarterly review of the PLL to:

A. Adjust stockage quantities of repair parts.

B. Verify location and condition of repair parts.

C. Increase stockage quantities of repair parts.

D. Reduce stockage quantities of repair parts.

Demonstration Tutorial: Using "Examiner" to Create Self Development Tests (SDTs)

RULE 3 . BE FAIR, NO CURVE BALLS, NO TRICK QUESTIONS, PLEASE.

HERE'S HOW:

There are many ways to write fair questions. Here are just few ideas.

a. Make correct choices absolutely correct,
Incorrect choices absolutely incorrect.

b. Avoid judgement calls, e.g. 'most important'.

c. Avoid 'All of the above' and 'None of the above' choices.

Demonstration Tutorial: Using "Examiner" to Create Self Development Tests (SDTs)

RULE 4. BE CLEAR. A 9TH GRADE READING LEVEL IS ABOUT RIGHT

HERE'S HOW:

1. Use job language

2. If two or more words say the same thing, pick the easiest word.

3. Don't ask in 20 words what you can ask in 10 words.

Go the the next screen for examples:

Demonstration Tutorial: Using "Examiner" to Create Self Development Tests (SDTs)

Below are 2 SDT questions taken from TRADOC documents. Each is followed by by a simplified question.

Before

You are performing operator maintenance on an M16A1 rifle.Your inspection reveals that there is NO dirt or corrosion to be seen through the vent holes. Because of the fact that there is no dirt or corrosion, which of the following procedures would you omit from your maintenance. ? [48 words]

After

You are doing operator maintenance on an M16A1 rifle. You see NO dirt or corrosion through the vent holes. Which of the following procedures would you omit from your maintenance. [30 words].

Before

You have been given the task of computing subsistence requirements in a theater of operations. It is necessary that you compute stockage strength. Using the following data, compute what the stockage strength should be at the supply point. [38 words]

After

What is the stockage strength at the supply point in a theater of operations, given the following data? [18 words]

Demonstration Tutorial for Using Examiner to Develop Self Development Tests (SDTs)

Test your understanding. What's wrong with the following item? How can you improve it? Check your answers with ours on the next 2 screens.

What does the Recoverability Code "A" mean?

A. Special procedures and handling must be followed.

B. Always trash.

C. Nonreparable

D. Allowance priority

Demonstration Tutorial for Using Examiner to Develop Self Development Tests (SDTs)

Here's our answer to "Test your understanding."

First what's wrong with the item?

1. It measures rote recall, not performance.

2. The alternatives are not well designed:

A, the correct choice, is longer than the other choices.

B is not plausible

C is O.K.

D is fictitious

3. Finally, the choices are grammatically different. A and B are sentences. C is an adjective. D is noun phrase. Choices should have the same grammatical form, e.g. all sentences or all noun phrases.

Go to the next screen to see how we've improved the question.

Demonstration Tutorial: Using "Examiner to Create Self Development Tests (SDT)

Our improvement.

You're the unit's PLL clerk. A mechanic turns in a water pumped he's replaced an M1 Tank. The pump is a PLL item. What do you first?

A. Deterime the recoverability code from the AMDF.

B. Find the recoverability code on the DA Form 3318.

C. Take the old pump to the SSA.

D. Put the old pump in the scrap.

Demonstration Tutorial for Using Examiner to Develop Self Development Tests (SDTs)

Now go to EXAMINER and practice writing items for the test plan you entered earlier.

Write one item for each of the four tasks in your plan. Write items for the PMs specified in the test plan which you entered earlier.

Next, generate a test from those items and compare your test with ours. You can then revise your test if you want to.

EXAMINER

Demonstration Tutorial: Using "Examiner to Create Self Development Tests (SDT)

Student shifts to Examiner and enters items for a sample test.

Then the student shifts back to the tutorial to compare his/her items against our items

Finally, student shifts back to Examiner to revise his/her items.

Demonstration Tutorial: Using "Examiner to Create Self Development Tests (SDT)

Student is using examiner to write items for:

SUBJECT AREA	TASK	PMs
1. Requesting Supplies	101-521-1133: Use of ARMS	PM 2
2. Manual PLL	101-539-1111: Repair Parts	PM 4
	101-539-1101: Maintain PLL	PM 3
3. TAMMS	101-539-1301: Maint. Con Log	PM 1

Next four screens show tutorial solution for writing items.

Demonstration Tutorial: Using "Examiner to Create Self Development Tests (SDT)

101-521-1133: Use of ARMS PM 2

1. Using the microfiche viewer and the AMDF, determine which repair part has NSN 2910-00-176-8928.

   A. Generator (*)
   B. Water Pump
   C. Battery
   D. Fuel Filter

Demonstration Tutorial: Using "Examiner to Create Self Development Tests (SDT)

101-539-1111: Repair Parts PM 4

2. As the unit's PLL Clerk, you requested 3 water pumps from the SSA. Later, the SSA calls and tells you they have 1 pump for you. After receiving the pump, how do you post your document register for this request?

   A . In ink, enter quantity received and julian date of receipt in columns J and M respectively

   B. In pencil, enter quantity received and julian date of receipt in columns J and M, respectively (*)

   C. In pencil, enter the quantity due-in and the julian date of the expected delivery in columns J and M, respectively

   D. In ink, enter the quantity due-in and the julian date of the expected delivery in columns J and M, respectively

Demonstration Tutorial: Using "Examiner to Create Self Development Tests (SDT)

101-539-1101: Maintain PLL   PM 3

3. You are the unit's PLL Clerk. You've just finished reviewing the PLL and have determined that the authorized stockage for one part should be increased. What action is not necessary for increasing the stockage level?

   A. Get the commander's approval (*)

   B. Note the increase on the DA Form 3318

   C. Request the item from the SSA

   D. Enter the increase on the DA Form 2063-R