AD-A247 360
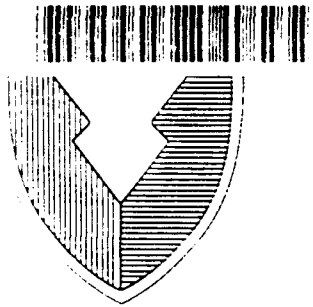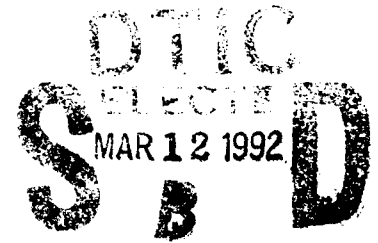
AD NUMBER _____

TECOM PROJECT NO. 7-CO-R90-EPO-005

FINAL REPORT

METHODOLOGY INVESTIGATION

OF

AI TEST OFFICER SUPPORT TOOL III

DTIC
ELECTE
MAR 12 1992
S B D

Robert Harder
CPT James Durham
Robert Williams
CPT(P) Tamara Stewart
SPC Gregory Adams
Curtis Massey
SPC Revis Farley

Software and Interoperability Division
Electronic Technology Test Directorate

U.S. ARMY ELECTRONIC PROVING GROUND
FORT HUACHUCA, AZ   85613-7110

1 December 1990

Prepared for:
U.S. Army Test and Evaluation Command
Aberdeen Proving Ground, MD   21005-5055

Approved for public release;
distribution unlimited.

92-06487

DISPOSITION INSTRUCTIONS

Destroy this report in accordance with appropriate regulations when no longer needed. Do not return it to the originator.

DISCLAIMER

Information and data contained in this document are based on input available at the time of preparation. Because the results may be subject to change, this document should not be construed to represent the official position of the United States Army Material Command (AMC) unless so stated.

The use of trade names in this report does not constitute an official endorsement or approval of the use of such commercial hardware or software. This report may not be cited for purposes of advertisement.

NOTICE

DEPARTMENT OF THE ARMY
U.S. ARMY ELECTRONIC PROVING GROUND
FORT HUACHUCA, ARIZONA 85613-7110

REPLY TO
ATTENTION OF
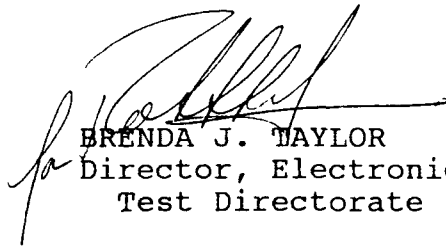
STEER-ET-S (70-10r)   24 January 1991

MEMORANDUM FOR Commander, U.S. Army Test and Evaluation Command,
               ATTN: AMSTE-TC-M, Aberdeen Proving Ground,
               MD 21005-5055

SUBJECT: Report, Methodology Investigation of AI Test Officer
Support Tool III, Project No. 7-CO-R90-EPO-005.

1. The subject methodology report is enclosed for your review
and approval.

2. Point of contact is Mr. Bob Williams, AUTOVON 821-8186.

FOR THE COMMANDER:

2 Encls   BRENDA J. TAYLOR
                                     Director, Electronic Technology
                                        Test Directorate

DEPARTMENT OF THE ARMY
HEADQUARTERS, U.S. ARMY TEST AND EVALUATION COMMAND
ABERDEEN PROVING GROUND, MARYLAND 21005-5055

AMSTE-TC-D (70-10p)

MEMORANDUM FOR Commander, U.S. Army Electronic Proving Ground,
ATTN: STEEP-ET, Fort Huachuca, AZ 85613-7110

SUBJECT: Final Report, Methodology Investigation of AI Test
Officer Support Tool III, TECOM Project No. 7-CO-R90-EP0-005

1. Subject report is approved.

2. Point of contact at this headquarters is Mr. Robert N. Bell,
AMSTE-TC-D, amstetcd@apg-9.apg.army.mil, DSN 298-7882.

FOR THE COMMANDER:

FREDERICK D. MABANTA
C, Technology Development Division
Directorate for Technology

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188
Exp. Date: Jun 30, 1986

1a. REPORT SECURITY CLASSIFICATION	1b. RESTRICTIVE MARKINGS
UNCLASSIFIED	

2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	Unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION	6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION
US Army Electronic Proving Ground	STEEP-TS-C	

6c. ADDRESS (City, State, and ZIP Code)	7b. ADDRESS (City, State, and ZIP Code)
Fort Huachuca, Arizona 65613-7110	

8a. NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
US Army Test & Eval Cmd		

8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS			
	PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO
Aberdeen Proving Ground, MD 21005				

11. TITLE (Include Security Classification)

Methodology Investigation of AI Test Officer Support Tool III

12. PERSONAL AUTHOR(S)
Robert Harder, CPT James Durham, Robert Williams, et. al.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM Nov 89 TO Oct 90	1990 December 1	79

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Artificial Intelligence, Expert Systems Automated Aids to Testing

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

This report covers the application of Artificial Intelligence (AI) techniques to the problem of creating automated tools to assist test officers. Phase III is described in detail, with a synopsis of previous efforts.

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

22a NAME OF RESPONSIBLE INDIVIDUAL	22b TELEPHONE (Include Area Code)	22c OFFICE SYMBOL
Mr. Robert Williams	(602) 533-8186	STEEP-TS-C

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted
All other editions are obsolete

TABLE OF CONTENTS

FOREWORD

ACKNOWLEDGMENTS

1

This Page Is Intentionally Left Blank

SECTION 1. SUMMARY

1.1 BACKGROUND

Test design and planning for modern Command, Control, Communications and Intelligence (C³I) systems require familiarity with a number of test operating procedures (TOPs) as well as detailed knowledge of specific test tool capabilities. A wide variety of tests must be designed, planned, and scheduled in order to efficiently conduct testing. Interrelationships among test groups and tools; common data requirements; data reduction and analysis requirements; lead time to prepare instrumentation; and required availability of the test item must be well understood in order to efficiently conduct tests within allocated time constraints.

The U.S. Army Electronic Proving Ground (USAEPG) has positioned itself to solve some of the problems faced by today's test officer by exploiting artificial intelligence (AI) technology and the quite capable microcomputer. Previous investigations at USAEPG, sponsored by the Department of Defense (DoD) Software Technology for Adaptable, Reliable Systems (STARS) program (reference 1), identified some aspects of AI which were sufficiently mature to insert in test tools. One of these technologies, AI expert (or knowledge-based) systems, was explored in depth. Others which are still under investigation are hypertext/hypermedia tools and artificial neural networks.

During the earlier projects, including Phases I and II of this investigation, prototype expert systems were developed to demonstrate capabilities and potential benefits. One of the first systems built to assess the suitability of AI technology for a proposed application is still being used to screen new proposals to eliminate those problems which are best addressed with conventional analysis methods. After the in-house skills were developed to build expert systems and differentiate between good and poor applications, a number of workshops were conducted.

The workshops produced many good ideas for expert system applications. Most applications were implemented during the workshops as "demonstration" level systems. A smaller number have evolved into more robust "prototype" versions. However, all of the systems shared the characteristics of being both developed on, and used in, a microcomputer environment. The viability and cost effectiveness of these microcomputer-based expert systems was shown during Phases I and II of the investigation (references 2 & 3). USAEPG continued to exploit this successful AI application methodology during Phase III, whose efforts are documented below.

1.2 PROBLEM

Testing C³I systems involves designing and planning tasks which are becoming increasingly complex. Advances in technology such as microprocessor design, distributed real-time architectures, artificial intelligence, and electro-optics are appearing in new C³I developments. While this sophisticated technology offers benefits to the developer, it is becoming a considerable burden to the tester. Test officers are required to identify appropriate test methods and associated instrumentation and data acquisition requirements for

3

each emerging technological area. This requires a level of expertise which is rarely found in any one individual. Besides being distributed among individuals, and therefore less available, this hard-earned expertise is frequently lost to the organization because of personnel reassignment or attrition.

1.3 OBJECTIVE

The objective of this investigation was to provide the test officer with automated support tools by inserting AI technology in appropriate applications. Objectives for the development of these tools included:

     a. Orientation toward the test officer as primary user.

     b. Wide usability to satisfy the needs of the approximately 100 test officers at the USAEPG.

     c. Ready availability (microcomputer based).

     d. Reduction in time to perform a given task and/or improved quality of the result.

     e. Education of the user (test officer); in addition to merely providing a solution, provide a means for the user to train new personnel by exposure to how a process is accomplished.

As testers, another objective of the investigation was to continue to identify test methodologies for the test and assessment of systems containing AI.

Finally, training potential developers and subject experts to identify good applications was perceived as a necessary adjunct to the widespread employment of AI.

1.4 PROCEDURES

Lessons learned from earlier work on expert system development were applied to restructure the original proposed approach. Rather than develop a single test officer tool on the one high end AI machine, an approach more in line with the objectives was established. This approach called for the development of a number of smaller tools, rather than risk all of the available resources on the success or failure of a single large tool. The smaller tools hosted on microcomputers provided a more flexible means of adjusting to resource constraints, while still benefiting from the technology.

This modified approach provided prototype versions of the Test Plan Drafter (TPD) and Environmental Impact Assessment Aid (EVA) systems. From this initial base, new ideas were developed in the areas of meteorological support, budget, security, safety, contract monitoring, human factors, and other supporting tools. Systems addressing these problem domains were developed using the workshop methodology: problem domain experts and knowledge engineers were paired to develop AI-based test officer support tools.

4

The issue of testing AI systems was investigated further. This effort encompassed basic methods of testing expert systems (with the idea of supporting testing of equipments that employ AI technology) and actual participation in formal AI test activities.

Training of new developers/subject experts was pursued by expanding on the concept of the workshops. To complement the workshops an apprentice program was developed to further educate and expose personnel outside of the AI office to AI techniques and application tools. This program was designed so that the apprentice would learn AI techniques and then apply them in the development of an expert system for his/her own office.

1.5 RESULTS

A number of AI expert systems were developed to aid the test officer in duties associated with testing. With respect to the application objectives outlined above, these systems satisfied those objectives as follows.

   a. The knowledge domains of the expert systems centered on areas of expertise for which an experienced test officer would be cognizant, but not necessarily an expert. In other words, a test officer might be familiar with certain security or contract monitoring requirements, but would still require considerable consultation with a domain expert to satisfy the requirements a new test. The systems built during this phase of the investigation were intended to assist test officers by providing the preliminary advice normally obtained from the domain expert during test planning.

   b. Most of the systems developed are still in the evaluation phase and therefore have been installed on a limited number of computer systems. A future consideration when these systems emerge from the prototype stage will be to examine the use of a central host computer or a personal computer (PC) local area network (LAN) for distribution and configuration management purposes.

   c. All of the systems were targeted for the microcomputers available at USAEPG. Because of the different configurations in use, some constraints exist as to which functions can be used while still retaining compatibility with a majority of the microcomputer base. Primarily these constraints have concerned disk and memory size, graphics capabilities, and hardware accelerators for floating point operations. From a practical standpoint, little functionality has been lost in conforming to the minimal configuration.

   d. An assessment of time savings or improved quality, due to the use of expert system aids, can only be done qualitatively, since all of the systems are just now being evaluated using actual test project parameters. Even after a production system is in place, cost saving and benefits will be hard to quantify as AI systems do not fit easily in a conventional software life cycle cost analysis. However, projected savings can be considerable in some cases; in one evaluation run, EVA assisted in identifying excessive and unnecessary test requirements. Other expert systems offer the potential of providing preliminary assistance in what can be complex or time consuming tasks. All of

the systems have demonstrated the ability to retain, and even combine, expertise from human domain experts.

e. The present suite of support tools all serve to train the test officer to some degree. After running the expert systems a few times, the officer begins to understand which parameters are significant for given situations. Also, all of the systems provide an online "help" function to inform the user of the nature of, and appropriate response to, the various queries encountered. Most of the advice offered by the systems provides both the necessary action and the reason for the action; e.g., use of incendiary devices requires filing a fire plan with the post fire marshal.

Test technology for AI expert systems exists in an embryonic stage. Participation in industry workshops has provided a forum for sharing ideas on possible approaches to testing AI. Also, participation in formal AI test activities has provided insight into what procedures are viable. Although some progress has been made in isolated areas, much remains to be done before AI test methodologies can be considered mature.

The first participant of the apprentice program allowed us to review and modify our process, thus improving the program. The program itself allowed the apprentice to provide his home division with a priority list of problems and types of solutions. This list was developed using knowledge engineering techniques and provided to the home division for review. After review, a prototype was developed of the selected problem. That system is presently being developed and is scheduled for completion early in 1991.

1.6 ANALYSIS

The development of various expert systems to aid the test officer demonstrates the usefulness of AI technology. The systems are still being evaluated, and will probably continue to evolve to support more of the domain knowledge. Besides the obvious benefits, such as retained knowledge and combined expertise of multiple experts, this methodology showed the feasibility of developing and using expert system technology with existing microcomputer resources. In addition, improved productivity and quality of work can be expected from test officers, along with an improvement in the testing process. With fewer resources available to essential mission functions, productivity and quality gains may overshadow other potential advantages of AI.

The systems developed for the investigation addressed individual problem domains within the testing arena. Many of these domains share commonality of information about test resources, techniques, and requirements - the infrastructure of testing. A broader analysis of this test support infrastructure requirements is appropriate. An early examination of the testing infrastructure, with subsequent incorporation of common requirements into a supporting structure (i.e., data bases, networks, geographic information systems, and standard information elements), could eventually lead to an integrated set of cooperating support tools.

Testing AI appears in two areas at the USAEPG: for systems embedded in test items (usually, Army systems) which must undergo developmental testing, and

6

for systems used in test support functions. The introduction of AI into test items makes it imperative that a test methodology be developed so USAEPG may perform its primary mission of testing. Almost equally important is the need to be able to validate test support tools which use AI. Until robust AI test methods emerge, the full potential of this promising technology will not be realized.

An apprenticeship program requires an investment in resources which could be devoted to development. Advantages, however, appear to outweigh any short term costs. At the end of the program another person is trained on the development of AI systems, a prototype will likely have been developed, and the usually underestimated burden of maintenance and further evolution of a system assumed by the apprentice's home office. Being outside the AI office, the apprentice will probably be able to identify possible AI applications which the AI office would neither be aware of nor have the resources to support. Also, the AI office would wish to maintain awareness of further developments by the apprentice, however, this would require far fewer resources than acquiring and maintaining expertise in the apprentice's domain.

1.7 CONCLUSIONS

The investigation was successful in demonstrating the capability of knowledge-based systems. This was accomplished with existing microcomputer resources, which increased the availability of the tools while minimizing costs. Further validation of this microcomputer-based expert system development methodology over a complete system life cycle would require that the prototype tools complete the ongoing evaluation phase. Following a favorable evaluation, the tools would then be fully developed and supported under production or instrumentation programs, for the remaining implementation maintenance portions of the life cycle.

Knowledge engineering techniques offer the possibility of supporting TQM activities. The end product of this approach would be an improved process and retention and documentation of corporate knowledge. An expert system byproduct of this approach would be merely a tool for examining current policy and procedures and modeling proposed changes.

Automating the entire test infrastructure is too ambitious an effort to be a part of this investigation. However, some consideration should be given to defining the infrastructure requirements for the production version of knowledge-based systems.

Since test items are already being developed which employ expert system technology, and knowledge-based test support systems have been shown to be beneficial, more emphasis should be placed upon initiating an AI test methodology investigation.

An AI apprenticeship program appears to be a viable methodology for acquiring the limited resources of an AI office. In the long term, more systems addressing a greater variety of domains could be developed and maintained with this approach compared to the alternative of developing all systems within the AI office.

7

1.8 RECOMMENDATIONS

Further investigation is recommended in the following areas:

a. Use of the prototype tools should continue through the evaluation phase to further validate the results obtained thus far. Distribution and operational considerations associated with the implementation phase of a system should be addressed, as well as maintenance issues. Further development of test officer support tools should also incorporate infrastructure requirements to the extent as possible.

b. A separate project should be undertaken to analyze the requirements for establishing and maintaining an automated testing infrastructure.

c. A project should be established to develop test procedures for AI. Efforts by industry workshop participants would continue to be monitored for new developments. Any practical techniques would then be applied to actual test situations. This effort would aid directly in accomplishing the primary mission of system testing, and would also offer a means to validate AI-based test support tools. Individuals and projects should be assisted by experienced testing personnel on site. TECOM/USAEPG could provide this type of consulting service, but it will be necessary to obtain Army sponsorship either through the DA AI Center or AMC.

d. Advances in hypermedia need to be further explored. This technology should provide several solutions in assisting the test officer in learning about the testing environment. One organization has tackled this problem, providing a study advisor. This effort would determine the feasibility of making regulations and local guidance documents, such as USAEPG's Test Officers Handbook, more readily accessible.

e. Advances in AI technology should be monitored to maintain cognizance of new developments in this maturing field. This should include those aspects of AI which have been explored only briefly during this investigation.

f. Methods to insert AI technology into the testing process at USAEPG should be aggressively pursued. The apprentice program should continue to be supported. It is an excellent vehicle for inserting AI technology into USAEPG. This methodology should also be applied at the test center level by providing apprentice (and mini-apprentice) programs to other U.S. Army Test and Evaluation Command (TECOM) test centers.

g. Use of AI technology to strengthen the development and retention of "corporate knowledge" should be explored on a larger scale. Possibilities for developing a testing infrastructure, exploiting hypermedia, and use of AI methods for Total Quality Management (TQM) deserve immediate attention.

SECTION 2. DETAILS OF INVESTIGATION

2.1 BACKGROUND

USAEPG is one of TECOM's nine test centers. TECOM established two goals for the use of AI technology, which are the primary goals of USAEPG's AI effort. One goal is to exploit AI technologies to enhance the ability to perform testing. The other, somewhat obvious goal, is to test systems which contain AI.

Testing is a complicated series of processes and is managed by individuals called test officers. Their primary duty is to oversee the activities associated with test directives. Besides test planning, the test officer is responsible for monitoring actual test conduct, and analyzing and reporting the results. With test items increasing in complexity due to the increased use of electronics, computers, and communications, the test officer's responsibilities are becoming more difficult. This would be sufficiently challenging even without the additional burden of reduced budgets and increased documentation requirements. At USAEPG alone, approximately 100 personnel are designated as test officers, with responsibility for conforming to all of the appropriate directives, regulations, and guidelines without losing sight of the primary mission. This can sometimes be a thankless job; the test officers must be constantly aware of the changing conditions and try to adjust to them.

This is the last phase of the three phase investigation to examine the potential of applying microcomputer-based AI technology to assist test officers in performing their job. AI has continued to evolve and change throughout this investigation. Neural networks have become more viable, expert systems have been integrated with conventional systems, and the methodology of applying AI has matured. This report details this year's work, updates previous efforts, and contains paragraphs describing a three-year perspective on sub-topics.

2.1.1 <u>Application Of AI</u>. The application of AI, or the technology insertion effort is almost an art in itself. It is not merely building expert systems, but involves managing the technology insertion and its effects on the organization. This approach requires that all aspects of an AI development infrastructure be addressed. Some of the essential ingredients of this methodology were the team organization, acquiring training and then training personnel at all levels in the organization, the development of various AI-based support tools, obtaining management support, and exploring AI testing issues. One method of training included an apprenticeship program located and supported in the AI office of the Software & Interoperability Division.

In the past year, USAEPG has emphasized TQM which afforded the AI efforts another subordinate role. That role has been to improve upon existing methods by examining existing processes, listening to experts/users, and in general defining and improving the job to be done - most of which are fundamental events in applying TQM.

9

2.1.2 <u>AI Background</u>. AI encompasses a large and somewhat diverse set of technologies, ranging from neural computing to robotics. Within that range exist expert systems, natural language processing, and vision systems. One of the more mature technologies of AI is that of expert, or knowledge-based, systems. AI developers have produced tools known as expert system shells that assist in the construction of rule-based expert systems. These shells allow a knowledge engineer to codify logical inferences (rules) about a given domain, then process the resulting knowledge base in order to provide expertise to the user.

Most non-trivial expert systems have been developed using a team consisting of AI and domain experts. It is the job of the knowledge engineer to obtain knowledge about a particular domain through consultation with one or more experts, documented information, or some combination of these sources. This knowledge is then incorporated into an automated tool which uses this expertise in solving problems within the domain. Expert system shells have considerably eased the task of developing expert systems, by providing a means to enter and exercise logical rules about a given domain. This leads to rapid prototyping of the knowledge of the domain and provides a proof of concept for the expertise developed.

Recent developments in expert system shells have resulted in a number of tools which are relatively easy to use and do not require extensive programming skills such as those normally associated with using symbolic programming languages. These shells have made it possible for some domain experts to build expert systems without assistance. However, knowledge engineering encompasses more than merely entering rules in the proper format. As a consequence, expert-built systems are usually small and expert-maintained and generally do not interface with conventional applications or data bases.

Presently AI seems to be experiencing a 'winter'. The explosion and interest in AI in the early and mid-1980's has slowed down as exotic promises of what AI could do remain unfulfilled. The trend in industry is to build fewer large AI systems and concentrate on integrating smaller ones into existing conventional programs. Sometimes, the AI portion assimilates facts, makes decisions, and then hands the information to conventional system components. Other times the knowledge-based system is embedded within an otherwise conventional system to make decisions. Some developers simply use AI techniques, but don't tell the user that they are getting an AI application. This trend can be seen by the emphasis placed on shells that allow integration with conventional systems and are portable across standard architectures.

There are also significant efforts to explore and retain corporate knowledge. No longer is it sufficient to simply 'capture' the expertise of an aging mentor; companies are seeking to capture and maintain the collective information designated as corporate knowledge.

2.1.3 <u>Problem Analysis</u>. In the past few years, most UASEPG AI applications have been elicited through workshops allowing USAEPG experts to explore their own ideas for improvement. This year the preliminary process varied in several ways. First, the AI team was approached several times to analyze problems for possible AI solutions. For these efforts, the overall process

was analyzed so that an appropriate 'fit' of an AI solution could be determined. Of note was the experience that using an AI solution resulted in improving the existing processes. Second, problem analysis was also utilized in the apprentice program (described below). Briefly, the apprentice and AI team met with several individuals of the apprentice's home division and discussed problems. These problems were then categorized into AI solutions, conventional (non-AI) solutions, and non-computer/TQM solutions. A rough effort/benefit analysis was performed, ranking the solutions for the apprentice's division chief to select the apprentice's project.

2.1.4 <u>Application Screening</u>. Expert System Selector (ES^2) was developed to assess the probable success of a proposed system by analyzing various parameters of the project. It was developed to provide additional screening for proposed applications and is itself an expert system. ES^2 examines such factors as the availability of expertise; supporting development and runtime tools; and the suitability and feasibility of an expert system solution to provide a qualitative score of the overall success potential. Proposed developmental concepts need to be well defined to allow ES^2 to provide a meaningful grade. ES^2 was used to screen ideas for workshops and was responsible for the elimination of what could have been poorly suited or overly ambitious suggestions. It was also used to help document some of the weaker aspects of proposed ideas.

The first method of screening to determine if an idea had merit was provided during the workshop brainstorming sessions. These sessions were structured to identify certain strengths and weaknesses of each idea. Participants were asked their opinion about characteristics of the idea and to give each idea a rating in two areas. The first area was how useful was the idea itself. The second was how difficult did the person feel it would be to accomplish; ratings ranged from easy to very difficult.

Another method used to continue problem analysis is to define the overall process. The system is viewed as the output of other efforts and as inputs into later processes. This requires delineation of questions such as:

     a. Who is the user?

     b. Who is the expert?

     c. How is it done now? (usually with work flow diagram)

     d. How would it be done in the future? (with diagram)

     e. Who will maintain it? (and to what extent can the proponent maintain it)

One of the most effective methods for obtaining answers to these questions was to have the proponent of the system visualize the existence of the system today and ask - "If the system were in existence today, how would it be used?" This technique has proven very effective in filtering out systems unacceptable to the user community or inconsistent with management practices.

2.1.5 <u>Microcomputer Development Environment</u>. The computing resources of
USAEPG include a variety of mainframe, mini, micro, and special-purpose AI
computers. However, only the ubiquitous microcomputer is readily available to
the test officer for planning functions. Earlier AI efforts demonstrated the
practicality of AI systems targeted for these machines. Microcomputer
applications were much more acceptable to the user. In some cases the
proponent was more comfortable with microcomputer applications as they were
more in control of their expertise.

Microcomputer implementations are not without their own unique challenges.

        a.   There is the need for practical methods to handle distribution and
configuration management. Much of the distribution problem will be aided by
the establishment of a LAN for USAEPG. Configuration management is a normal
problem in software development, especially in small expert systems
distributed by "floppy net".

        b.   New versions of the system may not get to all users of the system.

        c.   Bugs or improvements or changes can be made very easily by the
developer leading to many small version changes. For most systems, it is
vital that all test officers are using the same version.

2.1.6 <u>Test Knowledge Infrastructure</u>. Another problem, not strictly limited
to applications on small machines, is the need for production level systems to
access information and knowledge on the testing infrastructure. Automation of
the testing infrastructure within the context of a large organization requires
at least two types of knowledge.

        a.   The first type, knowledge of the domain in which the system is to
advise and assist, is termed domain expertise. This type of knowledge is
commonly described in AI literature.

        b.   The second type involves information concerning the administrative,
organizational, and regulatory environment in which the expert and system must
operate. Within USAEPG, as with most organizations, requisite information is
widely available, but from a variety of sources. At this time, there is no
central point for the maintenance of or access to this infrastructure
information.

One organization that has tackled this very effectively is the Concepts
Analysis Agency (CAA). The following short extract summarizes this system
(reference 5). Substitute "test officer" for "study director" to have a good
handle on the test knowledge infrastructure.

                "The CAA Study Director's Advisor was developed
                using hypertext capabilities and object-oriented
                programming to provide an effective working
                environment for study planning and management.

A study director at CAA has the responsibility to
perform analysis on a variety of issues ranging from
quick reaction assessments of limited topics to
year-long examinations of Army force level systems
in the context of joint or combined forces. The
onus is on the study director to carry the study
effort from establishing initial requirements to
conveying results and insights. This is a complicated
and difficult process. While the agency maintains
considerable documentation on the study process and
the resources available to shepherd the study
director, this guidance is not all centralized and
is often unknown or unavailable to the busy study
director. The Study Director's Advisor fills this
gap and serves as a primary focus for reference
material as well as a planning environment for study
directors at CAA. The system's objectives are to
provide an effective set of tools for:

Study planning and management.
Document preparation.
Study director training.
Improvement in study quality and consistency."

Items like the test officer handbook, USAEPG's missions and functions, could
be handled in DOCUVIEW. Items like briefings and diagrams would be best
handled in a more powerful hypertext environment. Windows 3.0 presently
includes as part of the package, a hypertext environment called Toolbook(tm).
This tool supports many of the same types of activities provided by the
MacIntosh product HYPERCARD(tm).

2.1.7 <u>Improvements In Application Development</u>. This year, hypertext was
added to some USAEPG applications and was found to be a very good companion
technology to expert systems. Hypertext can provide a good user interface.
Questions previously fed one at a time by the inference engine can be grouped
and defaults provided at the beginning of a session. Hypertext also handles a
lot of the knowledge that cannot be easily coded into rule-based or frame-
based systems such as the test knowledge infrastructure. Help files or
explanatory windows are done easier in hypertext. Many microcomputer AI tools
are beginning to include some hypertext and advanced user interface features.
With the age of windows, mice, menus, and LANs upon us, users are demanding
very sophisticated user interfaces. The acceptance factor for expert systems
is beginning to include the user interface, database interface, report
generation capabilities, and LAN access.

A need to review and improve on integrating common knowledge areas between
applications has proved to be interesting and to present several unique areas
to study. The possibility of using a common shell for all applications is a
possible solution, but does not necessarily allow use of the most suitable
tool for a particular situation. Identifying common knowledge first became

necessary as several applications were found to ask similar questions. An analysis of the similarities revealed these questions were of the form:

    a. Do you have _____?

    b. Are you using _____? (electrical generators for example)

Once the expert established the existence of an item (such as generators), the concerns were different. For example, security would be interested in preventing theft or vandalism to the generator. The environmental person was concerned about refueling the generator, the amount of noise it produced, and the amount of fumes. Developing these common infrastructure elements is certainly a worthwhile goal and should be considered in future efforts.

2.1.8 <u>Post Development Environment</u>. Several issues have begun to be raised as the prototypes from this methodology have matured. Noteworthy is the proponent or lack of one. Although several processes such as test planning have been identified, in some cases there is no organizational proponent for the process. This translates in to no real proponent for the AI system, and sometimes means several conflicting experts are involved.

Another major issue after development is the distribution and maintenance of the systems. Will the proponent, MIS shop, or the AI shop maintain the system? How will every test officer obtain the current version of the system? At the present time, USAEPG has no integrated LAN capability for maintaining a central copy of an application or application databases. Configuration management becomes a difficult task at best.

Other issues include the age of the AI shell and related software packages. Some AI software packages and shells do not age well. They are no longer available or supported by their developer. With multiple packages the issues become even more involved because a new version of one tool may not be interoperable with the current version of other tools.

2.1.9 <u>Team Structure</u>. USAEPG AI efforts are managed out of an office in the Software and Interoperability Division. The team consists of management, engineering, computer scientist, and apprenticeship personnel. These personnel are supplemented with an existing technical support contract. Upper management plays a key role in obtaining the commitment and resources so essential to the insertion of new technology. Because a successful technology development program requires both adequate tools and the management and technical skills to effectively use the technology, a considerable amount of emphasis is placed on training at all levels in the organization.

The USAEPG AI team now consists of two officers, two civilians, two enlisted persons (E4s), and 3-4 people on the technical support contract. The addition of one civilian over last year's level was for expertise and continuity. We have found the enlisted personnel to be quite valuable; adding testing experience and providing the soldier's perspective. Most AI teams in the Army have not utilized NCO or enlisted sources to this extent. Besides developing applications, they have been invaluable as computer technicians, keeping track of resources, and coordination with other elements of USAEPG.

It helps during the developing of AI applications, by any AI team, if it has a variety of skills. Multi-disciplines allow at least someone on the team to have some empathy with the proponent. It makes no difference if this background comes from previous experience; training; or for the military, from several MOSs.

The team should be multi-ranked to allow junior people to perform those aspects of a project commensurate with their experience. This decreases the time a senior knowledge engineer spends on mundane or easy tasks. The team should have a mixture of civilians and military to add continuity and purpose. Much of the contract effort has been indispensable, but it would be wise to keep a majority of the AI expertise in-house. Much can be lost otherwise.

2.1.10 Management Involvement. Management participation is an essential element of any new technology insertion effort. Management oversight was cultivated through the establishment of a steering committee. The goals of the committee meetings are to provide the communication channel to senior management from the AI cell and provide a forum for resource commitments and priorities to be assigned to proposed projects, based on command perspectives. If a steering committee is not possible or desirable, then open communications with upper management is still an essential element.

Training and education for management personnel is essential and a continuous effort. Seminars and small workshops were very effective in providing this education.

2.1.11 Interaction With Other Projects. AI projects under this methodology do not seem limited to this project. Synergism and leverage occurred in several unexpected ways. Under the Research and Development Instrumentation (RDI) effort, projects starting under this investigation were/are being developed to more robust and production systems. Previously, this was accomplished by taking a prototype tool developed in one year and trying to find funding to produce a production version the next. An example of being able to move an application from methodology to RDI was this year's effort on human factors engineering (HFE), the project was moved from methodology to RDI after the apprenticeship was over.

The TECOM AI Seed program was started to give other test centers some start-up capital. Projects designated to become TECOM-wide applications were the security and environmental systems as described in following the sections.

Under the Small Business Innovative Research (SBIR) program, USAEPG has had several projects that have looked at testing AI. This information, combined with the effort performed by this investigation, provides good coverage of the upcoming problems in testing systems containing AI.

2.1.12 Summary. In summary, AI is an emerging discipline, the application of which is a constant technology insertion effort. What is considered AI today, will become merely a standard discipline in the future. The most significant finding has been that AI is not stand-alone, nor the unique solution. AI and an AI shop require special skills and long term commitment from management to apply the emerging technologies. The AI applications and the AI shop must be

15

integrated within the existing organization. The best goal for AI applications and an AI shop is to help the organization pursue its mission by streamlining its processes. This is accomplished with the willingness to identify potential areas for improvement and accept a restructured process. Along the way, the best technology for a specific problem can be applied, and critical knowledge can be captured and put to work multi-fold.

2.2 NEW TECHNOLOGY

Several activities supported the exploitation of new technology in the AI field. The area of hypermedia was explored from the latest hypercard media developed by MacIntosh to a hypertext tool developed and supported in MicroSoft Windows 3.0 called ToolBook. One of our efforts developed a hypertext document handling tool that is presently being used in several small applications. Other efforts in this area were an examination of example-based AI development shells, a rule-based shell recently released by the National Aeronautics and Space Administration (NASA), visual programming environments, natural language processing, neural networks, and object-oriented development methodologies. More emphasis is being placed on the areas of natural language processing and neural networks. Several of the SBIRs that we presently are looking at have these two technologies.

2.2.1 <u>Three Year Perspective</u>. In the earlier phases of the investigation it was demonstrated that large, expensive minicomputers or specialized workstations were not necessary to exploit the benefits of AI. However, users are becoming increasingly more sophisticated and demanding better user interfaces, more functionality, and improved performance. The enhanced capabilities of conventional application packages - graphics, mouse selection of menu items, compatibility among packages, and more robust functionality - have caused users to expect these same characteristics of AI applications. Fortunately, the tools to address these increased requirements are entering the developers toolkits.

During the period of this investigation, considerable progress has been made toward adapting many of the features of conventional packages to AI development tools. The expected degradation in performance due to integration of these additional capabilities has continued to be offset by impressive cost/performance gains in the current generation of desktop microcomputers. Even development methodologies have improved with the general acceptance of techniques such as object-oriented programming. These trends can be expected to hold for the future, with conventional and AI tools exchanging successful methodologies and expanding their scope of suitability.

2.2.2 <u>Hypertext</u>.

2.2.2.1 <u>Introduction</u>. Hypertext is a method of preparing and presenting information linked together by an author for readers to retrieve and review in a non-linear fashion based on their needs or interest. The information is presented as pages or cards with linked items identified for additional information. Depending on the capabilities of a selected hypertext tool, the information may include additional text, graphics, sound, speech, animation, and execution of other programs. Hypertext documents allow the reader to dynamically alter the sequence in which information is presented. The result is that the reader is given complete control of the information, yet the author has had the opportunity to establish the structural links and control the detail and direction of a reader's explorations. This technique results in rapid assimilation of new information, without having to review nonessential, or already known, material. Several hypertext tools are available and applications have been written using hypertext techniques. This

17

effort was undertaken to obtain and review hypertext tools and sample applications for techniques applicable to future AI development efforts.

2.2.2.2 DocuView. DocuView was developed for USAEPG and was designed to be used on Zenith Data Systems, IBM, and IBM compatible PCs. DocuView is intended for displaying general textual information on PC displays. Hypertext expansion techniques are available through highlighted phrases within a document organized into pages. The author embeds various commands within the text of a document to provide flexibility in presentation of the document's information. DocuView has been distributed in AI workshops conducted by USAEPG and to other interested parties. Comments received from users have been for inclusion of an embedded editor and graphics capability within DocuView. DocuView was used to present results of an investigation into test issues for decision aids containing AI. These results identified and provided references to various methodologies for testing knowledge base system components. Additionally, results identified software quality factors and subfactors and identified which of the methodologies provided metrics for measuring each factor/subfactor.

2.2.2.3 HyperWriter. HyperWriter is a commercial product released in February 1990 by NTERGAID Inc. of Fairfield, CN. This product is an enhanced and rewritten version of their previous hypertext tool, Black Magic. HyperWriter was designed for IBM AT compatible machines in an MS-DOS environment. The product has an embedded editor and provides graphics and file manipulation among its features. ESGUIDE is an application developed with HyperWriter. ESGUIDE is a hypertext version of "Testing and Evaluating C^3I Systems That Employ AI, Volume 3: A Guide to Developing Small Expert Systems," a document prepared for USAEPG (reference 7). ESGUIDE provides rapid access to various sections of the document to a user needing an on-line reference.

2.2.2.4 1st-CLASS-HT. 1st-CLASS-HT is an expert system development tool from AICorp Inc. of Waltham, MA. The program is available for IBM PC compatible machines using either an MS-DOS or OS/2 environment and also Digital Equipment Corporation VAX machines running a VMS environment. The MS-DOS version has been obtained for use. Being an expert system development tool, instead of a stand-alone hypertext tool, this program provides hypertext features for developers to control large amounts of text and graphics for easy access by users in knowledge base applications. A hypertext editor is provided for rapid development of a user interface, allowing the user to access text and enter information through a logic tree-structured expert system. EVA, an environmental assessment expert system application under development for USAEPG, utilizes 1st-CLASS as will CPEA and SPA. Use of hypertext features has significantly reduced the number of rules required and simplified the input of data over that of EVA's prototype version. This was accomplished in addition to enhancing the availability of help information and allowing user control of the input data sequencing.

2.2.2.5 HyperCard. This program is a commercial product of Apple Computers and runs on Apple's MacIntosh personal computers. HyperCard is a toolkit program for creating, organizing, and linking information. The toolkit gives users the ability to create cards and organize the cards into stacks. The

18

user may also use, customize, and create new information types such as text, graphics, video, voice, and animation by referencing the cards/stacks. In addition, HyperCard provides a scripting language instead of a few commands for embedding into the information for presentation. HyperCard's scripting language is HyperTalk and gives users an opportunity to write their own programs for manipulating cards and stacks. The creation and manipulation of new information objects results in a user performing object-oriented programming.

USAEPG obtained the Study Director's Advisor application, which uses HyperCard, from USA Concepts Analysis Agency (CAA). This application provides information on CAA, guidelines and directions on conducting CAA studies, information on tools, models, and data bases used by CAA in studies, a file of memo and regulation references, and guidelines for preparing briefings and study reports. Additionally, the application provides a study information area for a study director to prepare and store administrative information; a work area for developing study objectives; places and examples for preparing a study directive and a study plan; and a variety of blank forms commonly required during conduct of a study. The forms section provides directions and assists in completing a form with data already collected.

This application is an excellent example of hypertext being used to present information for multiple reasons. The application performs as a good tool for briefing about CAA and how it conducts studies; a training tool for new study directors; a quick reference to a study director on guidelines and directions during conduct of a study; and as a tool to assist a study director in planning and conducting a study.

2.2.2.6 ToolBook. This program is a commercial product of Asymetrix Corp. of Bellevue, WA that runs on IBM AT compatible personal computers under an MS-DOS with Windows environment. ToolBook is a full-featured hypertext tool for authoring/reading books consisting of pages (corresponds to stacks consisting of cards in other hypertext terminology). ToolBook provides file management and an embedded text editor as well as a graphics editor supporting draw and paint objects. ToolBook has OpenScript as its scripting language. ToolBook is used to define objects and OpenScript is used to define the instructions of an object's response to specific events/actions. Thus OpenScript is an object-oriented programming language within ToolBook.

2.2.2.7 Conclusions. Hypertext is a highly useful and evolving technology for the presentation of and the interaction with information. A Hypertext tool's usefulness is related to the objects it supports and the fit of a problem into these objects. The Hypertext tools facilitate the creation and manipulation of its supported object types. Hypertext tools with a scripting language enriched with programming capabilities for defining new object types and the manipulation of these objects leads one into object-oriented programming.

2.2.3 Example-Based AI Development Tools. Example-based shells use an inductive inference methodology. This technique accepts objects of a known class (i.e., the "examples") with a fixed collection of attributes. The attributes are distinguishing characteristics which determine which set of

19

objects (class) a given object belongs to. After processing by an inductive algorithm, a decision tree with attributes is produced which may then be used to classify unknown objects. This methodology is well suited for classification, obviously, and diagnostic problems.

2.2.3.1 Investigation. To assess the potential of example-based tools, the Software Analyst's Assistant (SAA), was converted from a minicomputer to a microcomputer environment using the 1st-CLASS shell. The SAA was used in this capacity because the rules had been developed as examples. The conversion process itself was a trivial undertaking, even though the SAA is a medium-sized system (approximately 500 rules).

2.2.3.2 Investigation Results. This exercise provided much insight into the specific features of the 1st-CLASS tool. (These will not be described in detail, other than to mention that the tool offers considerable flexibility, an extremely user-friendly interface, and a classification algorithm with a linear time function.) Most important is the capability to graphically display the decision tree built by the inductive algorithm. This logic tree can be examined to avoid creation of extraneous inferences, and conversely, to identify situations unaccounted for. (The conversion of the SAA resulted in the discovery of one instance of the latter case, although the impact of this oversight in SAA operation turned out to be insignificant.)

2.2.3.3 Conclusion. Example-based tools can be extremely easy to use since the development environment builds the rules automatically, given example situations as input. Although the tools are easy to use, some caution must be exercised to ensure that a potential problem domain is amenable to use of inductive techniques. Examples, of course, must exist, or the domain expert must be able to provide examples. Less obvious is that attributes which distinguish one set of objects from another must be defined. The examples used to build a system must be representative of the domain, and cover all of the classes among which the system must distinguish. A good design methodology would also provide for ordering the attributes by the cost of obtaining the information. (An excellent paradigm of this last requirement is offered by a hypothetical medical diagnostic system. Attributes such as temperature, pulse rate, etc. would be used to identify a pathological condition, if at all possible, prior to requiring exploratory surgery.)

Example-based shells can provide other benefits as well. Some are good for discovering any underlying structure in low level data (i.e., they perform a "factor" or attribute analysis for the developer). Another useful feature of some shells is the ability to provide counter examples where examples are either too few or much too extensive (e.g., a medical diagnostic system which attempted to describe all the attributes of a well person).

Used properly, with appropriately structured problems, an example-based shell can be a tremendously effective development tool. Microcomputer versions can adequately handle real size problem domains with performance comparable to, or better than, rule-based shells. But the most interesting feature (at least for a testing organization) is the ability to validate a system by visual and automatic examination of the logic tree.

2.2.4 <u>NASA C Language Integrated Production System</u>. The NASA C Language Integrated Production System (CLIPS) tool is a development and delivery expert system tool which provides a complete environment for the construction of rule-based expert systems. (Tools such as M.1 require the user to provide his/her own text editor.) Versions are available for a number of computer environments, including a microcomputer environment which is compatible with USAEPG resources. The CLIPS distribution package has a number of potentially useful and unique features. Source code and documentation are available at no cost to government agencies and their contractors (call the CLIPS Help Line at (713) 280-2233). The system is currently limited to forward chaining; but it has a powerful rule syntax, is portable, can be embedded within conventional procedural code, and can be extended with the addition of user-defined functions. In addition, CLIPS comes with a utility to aid in verification and validation of rules by providing cross referencing of fact relations, style checking, and semantic error checking. An Ada version of CLIPS (current implementation is in C) is also being developed.

Experience with the CLIPS environment has been too limited to provide an assessment of the full potential of this otherwise promising tool. A distribution copy was obtained, and the tool was introduced to attenders at a mini-workshop. Since the initial reaction of users has been favorable, after further experience with CLIPS has been gained, a more extensive workshop featuring this tool will be conducted.

2.2.5 <u>Visual Programming Environment</u>. Modern graphical user interface (GUI) technology has led to exploration of computer-aided software engineering (CASE) tools that make far greater use of graphic representations of software objects and relationships. One subdomain of this field, visual programming, involves execution or compilation and execution of programs directly from some largely graphic representation of the program. This is termed visual programming. Such programming offers the potential of relatively rapid development and test of prototype applications as a result of the higher information capacity, or "bandwidth," for information transfer inherent in graphic representation. One implementation of this approach, selected as inexpensive and available for the microcomputer environment, is Matrix LAYOUT, from Matrix Software Technology Corporation of Boston, MA.

2.2.5.1 <u>Investigation Approach</u>. The package was installed on a Zenith Z-248. Initial efforts involved implementation of the tutorial example program outlined in the documentation provided. Several key features were investigated by developing sample applications requiring those features. These applications were exercised under a variety of conditions, and results noted informally.

2.2.5.2 <u>Characterization</u>. Matrix LAYOUT represents programs as simplified flowcharts. Flowchart elements are selected from a menu, and critical parameters, e.g., variable names, screen positions, repeat construct conditions, operators, etc., are entered as part of a dialogue or via mouse selection from lists of existing elements displayed for that purpose. The primary file structure supported is a cardfile, where cards correspond to records in conventional data base systems. LAYOUT is particularly strong in its implementation of the pull-down menu/window/dialogue box model of user

21

interface, and provides extensive and easy to use facilities for implementation of this type of application.

2.2.5.3 Investigation Results. Creation of simple prototype applications in LAYOUT is significantly faster than with other methods in use at USAEPG. Comparable facility is attainable in modern conventional programming environments, e.g. Turbo Pascal, only after some months of experience with the language, and with a significant number of previously developed libraries, either third party or in-house, available for special functions. This type of tool can be a vital adjunct to AI developments where an expert system may require significant conventional software support for information and data management. Unfortunately, the current version of LAYOUT is still immature, having design and implementation flaws in a number of areas. Syntax and semantics for operators on some data types are ambiguous or inconsistent, documentation of many flowchart element types is inadequate, and error handling for a number of conditions is non-existent.

2.2.5.4 Conclusions. Use of the current version of this tool for conventional support to expert system development is impractical due to the deficiencies in the current implementation. A more mature version of this tool, or an alternative visual programming system, is highly recommended for conventional software support for expert system development and implementation.

2.2.6 Natural Language Processing. As part of its responsibility, TECOM must evaluate Army draft Technical Manuals (TMs) accompanying equipment or systems to be tested by TECOM. An important feature of this evaluation is determining the reading grade level of the manual. Use of automated tools to determine reading grade level for military TMs was investigated.

2.2.6.1 Comparison of Grade Level Scoring Methods.

    a. Grade level is computed with the Flesch-Kincaid reading grade level formula to meet the requirements imposed by Military Standard (MIL)-M-38784A, Amendment 5. The reading grade level must be close to a predefined target value to be acceptable. The investigation covered the evaluation now being done by hand and the capability of commercially available software packages to meet the requirements of the standard. Also considered was what additional effort might be needed, and what savings could be expected, by using software packages.

    b. In accordance with the rules of MIL-M-38784A, a total of 29 samples were selected from a draft TM for the comparison of reading grade level scoring methods available. The study compared hand scoring against the Grammatik IV and RightWriter software packages. The results are shown in Table 2.2.6.1-I, which shows the grade level score and the total counts of syllables, words, and sentences found for each sample by each method. The samples referred to in the table are only those samples taken for this study.

Table 2.2.6.1-I. Comparison of Reading Grade Level Scoring Methods.

SAMPLE NO.	METHOD											
	Scored by hand				Grammatik				Rightwriter			
	rgl	syl	wds	sent	rgl	syl	wds	sent	rgl	syl	wds	sent
Sample01	9.6	375	219	17	9	375	219	18	9.19	377	222	18
Sample02	15.4	419	197	13	17	397	192	10	15.73	416	205	10
Sample03	13.8	445	218	16	11	406	216	21	10.63	418	220	20
Sample04	10.1	362	200	18	10	355	190	21	10.10	368	205	16
Sample05	8.1	331	200	19	11	380	193	23	9.19	392	215	21
Sample06	8.9	348	198	21	9	348	189	24	7.75	381	225	21

Sample10	10.6	338	203	12	10	305	164	16	8.64	353	208	13

Sample15	11.2	388	208	17	12	348	175	18	10.11	388	211	15
Sample16	8.5	333	202	17	8	292	172	20	7.75	345	206	17
Sample17	8.6	379	220	22	8	319	189	20	7.50	366	224	17
Sample18	7.7	340	209	20	8	310	180	22	7.16	339	213	17
Sample19	6.0	299	210	17	6	246	157	22	5.99	325	215	17

Legend
rgl -- reading grade level score
syl -- number of syllables
wds -- number of words
sent -- number of sentences
sample n -- Nth sample of text from the Technical Manual

2.2.6.1.1   Scoring by Hand. Scoring by hand has the advantage that human evaluators can judge what is a separate thought, and therefore what is a sentence, without depending solely on punctuation supplied by the writer. However, individual interpretation of the standard may vary from one evaluator to the next. Human evaluators are susceptible to errors in counts of syllables, words, and sentences. They are prone to errors in assembling, totaling, and manipulating counts and scores when in the process of making mathematical computations.

2.2.6.1.2   Scoring Using Rightwriter Software. Rightwriter produced reading grade levels and counts which are close to those produced by human evaluators. However, there are several problem areas. Determination of what is a sentence is dependent upon punctuation. Abbreviations or list elements of one or more words each, followed by periods, are mistaken as sentence ends. Constructs such as "FM 238" are counted as two words rather than as the one-word name "FM 238". Also it should be noted that paragraph numbers and other sequential

text identifiers or lists are included in the syllable, word, and sentence counts in variance with the standard.

2.2.6.1.3 _Scoring Using Grammatik IV Software_. Grammatik IV produced reading grade level scores and counts that varied from those produced by human evaluators. Some counts were considerably lower. This is caused, at least in part, by the fact that paragraph numbers and other sequential text identifiers or lists are not counted as words or syllables, in agreement with the standard. However, construct items which appear like paragraph numbers, other sequential text identifiers or lists, but are not these items, are also not counted. An example of such an occurrence is a reference in the text to a feature at location 1 in a diagram which might be written (1) or {1} or in yet another list form. These references are not counted, in variance with the standard.

2.2.6.1.4 _Questionable Results_. Other areas of questionable results were that determination of what is a sentence is dependent upon punctuation. Abbreviations or list elements of one or more words each, followed by periods, are mistaken as sentence ends.

2.2.6.2 _Improvement in Scores from Software_. Enhanced scores can be obtained from software packages by first doing some extra editing of input TM text. This extra editing would include removing periods after abbreviations or list elements, removing any spaces in names, and removing any periods or other punctuation characters used for spacers (e.g., in tables) and replacing them with blanks or other special characters.

2.2.6.3 _Methods of TM Text Entry for Software Analysis_. Draft manuals are currently received by the responsible agency in written form. It is necessary to have selected samples of text entered into disk files either by word processing personnel, or by some other appropriate means, before the text can be made available for software analysis. Word processing time required would be about one day at 10 - 15 minutes per sample, including error checking. The use of optical scanning equipment to enter text would take approximately three to five days for 30 samples. This time estimate is based in part on the knowledge that it is difficult to align text copy for optical character reading. Frequently the result must be edited or some part of the text reentered. The full sample text may have to be entered by typing, depending on the quality of draft manual print copy and the juxtaposition of any diagrams.

2.2.6.4 _Conclusions_. The most dependable means of determining grade level is through the use of software tools which perform natural language processing. Once tested to confirm its correct operation, software can be depended on to repeat the same error-free operation each time it is executed.

Grammatik IV operates close to the requirements of MIL-M-38784A, including the reading grade level using the Flesch-Kincaid formula. The software also allows additional rules to be added to its rule dictionary to enhance its use in TM evaluation. For these reasons, Grammatik IV is judged to be the best software to use at this time.

The draft TMs should continue to be sampled and tested for reading grade level to ensure that parts of the TM are not overly high or low in reading grade level score. This should be done even though obtaining the overall grade level (OGL) using the full TM text will produce an OGL score free of those sampling errors produced by poor choices of samples; for example, all easy-to-read samples or all hard-to-read samples. Careful sampling of the TM must continue to be a very important part of TM evaluation.

2.2.6.5 <u>Recommendations</u>. Software should be used to determine reading grade level. The preferred approach is that samples selected from manuals in accordance with MIL-M-38784A be entered onto diskette by word processing personnel, and each sample be analyzed for reading grade level using Grammatik IV. After all samples have been scored individually, samples should be combined into one total sample file to be scored for the overall reading grade level with Grammatik IV. This OGL score should be compared with the target grade level score for the TM. The analysis can be continued by scoring by hand, reading, and studying any questionable samples.

2.2.6.6 <u>How Future Grade Level Scoring Might Be Handled</u>. In the future, the TM evaluation process could be significantly improved by requiring that the manual be supplied by the developer on computer diskette. This would simplify manual handling whether software analysis is used or not, assuming that desktop computers are available to the evaluator. The following steps are suggested for the evaluation process:

     a. Select samples from the draft manual in accordance with MIL-M-38784A.

     b. Transfer selected samples onto appropriate files.

     c. Obtain reading grade level of samples using Grammatik IV such that grade levels of all the various manual sections are obtained.

     d. Obtain OGL using Grammatik IV operating on the full TM text. This obtains the OGL without sampling errors. The OGL could be obtained by analyzing the combined samples.

2.2.7 <u>Neural Nets</u>. Little effort was expended on examining neural networks during the last phase of this investigation. Previous effort had identified the suitability of this technology for certain classes of problems (primarily pattern recognition). The improved performance of microcomputers should make the application of simulated neural networks feasible for a wider range of problems. However other areas in TECOM such as CSTA and DPG have prototyped the use of neural nets for data validation during a test. This use is very promising for a testing organization.

2.2.8 <u>Object-Oriented Programming</u>.

2.2.8.1 <u>Introduction</u>. Programming paradigms are models of how to design and implement programs. Different models result in different techniques. Because techniques differ does not imply they are in conflict. Various techniques can be seen to complement one another. The common notions of programming models

are that the design should be based on abstractions corresponding to elements in the problem and that the implementation should be a collection of modules, preferably reusable ones. The paradigms differ on how to form abstractions and what constitutes a module.

      a. The methods of procedural programming are based on a model of building a program as a collection of functions. The techniques provide guidance on how to design, organize, and implement the functions that make up a program. The design method of functional decomposition identifies the functions that serve as the abstract operations for solving the problem. File organization allows functions to be grouped in separate modules, and structured programming techniques enhance the readability and maintainability of a function's implementation.

      b. Object-oriented programming is based on a model of building programs as a collection of abstract data type instances. The techniques provide guidance on applying data abstraction to the data structures of the problem. Access and manipulation of the structure is provided by sets of operations that are part of the data types. Object-oriented design identifies the types that represent objects in the programming problem. The operations in the object types are, like functions in the procedural programming model, the abstract operations that solve the problem. The object type serves as a module that can be reused for solving another problem in the same domain.

      c. Data abstraction focuses on the data structures that are neglected by procedure-oriented techniques. The model of data abstraction is that data structure should be defined by operations on it, rather than the structure of its implementation. Data abstraction complements the procedural programming view of functions as abstract operations because neither abstraction is complete without the other.

2.2.8.2 <u>C++</u>. The object-oriented programming approach is defining a collection of object types, creating instances of the objects for the specific problem, and invoking operations to do the processing. The major addition C++ makes to the C programming language is the introduction of class types. Classes allow a user to define aggregate data types that include not only data members but also member functions that operate on the type. The member function with the same name as the class is the constructor function which creates and initializes an instance of the class. Data hiding in classes provides the mechanism for data abstraction whereby the properties of an abstract data type are defined by its interface, not its structure or implementation. Class inheritance extends data abstraction by providing a mechanism for building new class types from other classes. In C++, classes serve as object types, and member functions provide the means for building operations into the type. Thus C++ provides a set of tools for the object-oriented programming approach.

2.2.8.3 <u>Common Lisp Object System</u>. Researchers have been experimenting with object-oriented extensions to Lisp for at least fifteen years. The ideas of Smalltalk were imported into Lisp several times and other researchers used Lisp to experiment with original ideas for how to organize object oriented programs. By the advent of Common Lisp as the standardized version of the

26

Lisp programming language, several object-oriented extensions to Lisp were available, some with widespread use. In 1986 at the Association for Computing Machinery (ACM) Lisp and Functional Programming Conference, Lisp users and implementors decided it was time to standardize on a set of extensions. The Common Lisp Object System (CLOS) specification was completed in June 1988 and comprises a tool set for developing object-oriented programs in Common Lisp. Implementations of this tool set are now entering the market. Use of CLOS should enhance portability of developed software as CLOS and Common Lisp are implemented on additional platforms.

2.2.8.4 Hypertext Scripting Languages. The scripting languages of HyperCard and ToolBook descriptions state that they provide for object-oriented programming. In a limited sense the scripting languages do provide for object-oriented programming if one constrains designs to the limited object types readily supported. These types include the buttons, cards, pages, graphics, stacks, and documents normally associated with hypertext applications. However, the scripting languages do not readily support the definition of more generic abstract data types not easily defined from the basic object types. Also the inheritance scheme only readily supports creating new object types from the basic types or types created from groupings of the basic types. This limitation allows for rapid creation of hypertext applications, but would soon prove difficult for more generalized applications. No single scripting language has been standardized to several product lines. Further developments in the scripting language products need to be monitored for emergence of a standard.

2.2.8.5 Conclusions. No single paradigm is suitable for solving all programming problems well. Programming techniques need to be applied flexibly, with an eye to how well they suit the problem at hand. Object-oriented programming does fit many AI problems because the problems consist of describing and manipulating knowledge of objects. Because object-oriented programming features are extensions to general programming environments, the choosing of an extension is restricted by the general environment. However, whenever a standardized set of extensions is available, it should be picked for the general environment. This will support the standard and will result in more readily portable software.

2.3 AI APPLICATION DEVELOPMENT

An AI expert system development methodology was synthesized from the lessons learned on previous projects, AI technology capabilities, computer resource availability, and elements of the testing infrastructure. This resulted in an approach similar to that used by industry for smaller AI applications:

a. Acquisition of microcomputer development tools and development of related personnel skills.

b. Identification of suitable applications and possible development tool.

c. Teaming of a knowledge engineer and domain expert(s).

d. Prototyping and iterative development of the expert systems.

2.3.1 Three Year Perspective. When this methodology was originally proposed several years ago, the concept was to produce one AI Test Office Support Tool to do all things for all people. The error of this approach was quickly realized and the result of adopting a more practical approach was the generation of a number of small expert systems which address the test officer's problems. These systems are described below and in the appendices.

a. Most of the systems deal with requirements during the planning phases of a test. This is not an indication that expert systems are not suitable for test conduct or reporting activities. Rather, it reflects the greater stability and better defined nature of the planning stage, and the unavailability of the expert tester. Test plans and environmental documentation are always required, regardless of other variations in the test conduct activity.

b. Another drawback to addressing test conduct requirements is that these types of applications are relatively large, require a lot of the expert tester's time, and would consume all of the present available resources of the AI team.

c. Changes that have occurred during this study are the increased emphasis on traditional software life cycle issues such as how the system fits within the organization, cost/benefits, maintenance, and user interfaces. Other changes include increased availability of microcomputers for testing personnel, and a general increase in computer literacy and interest in expert systems.

2.3.2 Human Factors Engineering - Prototype. HFE was designed and is being developed to assist the MANPRINT RAM Division in generating questionnaires for test items. It also assists in the training of novice users to create those same test item questionnaires. HFE was developed as an expert system using typical AI knowledge acquisition, rapid prototyping, and knowledge engineering methods, however, the production level version of the system will be implemented with conventional software tools. Because of the rather simple

28

structure of the rules of the domain and the large amounts of conventional
data items which must be manipulated in a microprocessor environment, a data
base management tool was chosen for both knowledge and data representation.

a. HFE is based on the questions contained in TOP 1-2-610 dated 30
November 1983. The data base was designed to take advantage of the categories
and sub-categories of the TOP with a further breakdown into groups to assist
the user's selection.

b. HFE is designed as a menu driven system that takes the user provided
input and creates a draft set of questions for review. Since the questions
are grouped as indicated above, the only questions included for review are
based on the input. The system is flexible, allowing the user to change or
add to the input parameters to create the questionnaire to review. The data
base structure also allows the user to add or delete categories, sub-
categories, or groups to allow for questions that are not included in the TOP.

2.3.3 Security Planning Aid - Upgrade. The Security Planning Aid (SPA) is a
rule-based system written in the expert system shell M.1. The system contains
knowledge from TECOM, Department of the Army (DA), AMC, and USAEPG regulations
as well as knowledge gathered from the experts from the Intelligence and
Security Division at USAEPG. The purpose of SPA is to assure realistic
security planning by test officers. It currently covers information security,
personnel security, physical security, operations security, and access control
for USAEPG. The system currently is at version 2.6, has supporting
documentation (Users Manual and Development notes), and is undergoing expert
validation using 20 test cases.

a. SPA through a series of questions to the test officer, determines
what security procedures, protective measures, actions, and memos that test
officers might need in order to insure adherence to security regulations. It
produces two sets of output: a check list for the test officer, and
documentation for the Intelligence and Security Division to review.

b. In the coming year we will attempt to implement SPA for as many of
the other TECOM proving grounds as possible. This will involve implementing
new modules, exporting the common core, and modification of certain current
modules. In order to do the above and to allow for help features, a better
user interface, and allow the program to be more powerful, we have recommended
that the application be moved to the expert system shell 1st-CLASS, with
hypertext capabilities. (The AI office and the contractor currently have a
licensed copy of the development environment of 1st-CLASS). This will still
allow for a runtime version of SPA to be distributed free, yet will provide
the power needed to upgrade from a usable prototype to a production system.

2.3.4 Environmental Impact Assessment Aid. The EVA analyzes various factors
of a proposed test (or training activity) to determine actions the test
officer should take to avoid unnecessary delays or increased costs due to
possible impacts to the environment. EVA considers location, nature of the
test, type and amount of off-road vehicles and troop activity, time of year,
and existing environmental documentation.

29

a. Programs of this nature are heavily dependent upon geographic data and will be much more acceptable when geographic information systems (GIS) become more commonplace.

b. In the coming year we will attempt to implement EVA for as many proving grounds as possible. An effort currently exists for Yuma Proving Ground.

2.3.5 Contract Performance Evaluation Advisor. The Contract Performance Evaluation Advisor (CPEA) provides assistance to test officers monitoring performance of a major cost-plus award-fee support contract. The contract delineates thirteen factors that must be evaluated each quarter for each task on the contract. Currently, eleven test officers monitor about 78 tasks. CPEA requests basic information from the project officer on each task, reasons about the answers, and presents a numeric range for the project officer to choose a score. Justification for the score is solicited for each factor. An evaluation report is generated for each task, signed by the project officer, and made part of the formal evaluation documentation. CPEA is an expert system designed to run on the IBM PC or compatible and is written in M.1. In the coming year we plan to look at exporting this product in some form to other USAEPG divisions that might need this type of assistance. Currently under consideration is the new EMETF contract. If this is done we will propose that it be done in a more powerful shell such as 1st-CLASS.

2.3.6 Software Analyst's Assistant - Upgrade. The Software Analyst's Assistant (SAA) is an expert system for testing software quality. The SAA provides the expertise to assess various quality factors through knowledge bases which incorporate the rules and criteria employed by expert software test engineers. The SAA comprises five major knowledge bases, covering descriptiveness and design issues. The SAA was hosted on a VAX minicomputer originally, with a limited capability microcomputer version available.

The high cost of minicomputer software packages and the increased capability of microcomputer development tools made it both desirable and feasible to rehost the SAA. Rehosting was performed using an example-based shell with hypertext features. A microcomputer environment supports both the development and the complete runtime system. The resulting product is now easier to support and use (both the development and end user interfaces are much more friendly), less expensive (by an order of magnitude), and provides additional capability with no reduction in performance. Although not originally built under this methodology, the upgrade of the SAA showed the benefits of using shells such as 1st-CLASS - see section 2.2.3.2 for more information.

2.3.7 Test Plan Drafter - Progress. The goal of the Test Plan Drafter (TPD) is to automate the current manual assembly of boilerplate for an initial draft of a detailed test plan (DTP). This is a time-consuming effort consisting of much cut-and-paste work from old test plans, but little real intellectual effort. TPD is intended to result in a strawman version DTP for distribution to specific subtest domain experts for further editing.

During the last phase of this investigation, the TPD's installation process was made simpler; reliability and documentation were improved; and TPD's use

increased. Despite the benefits gained from some use of the tool, it was determined that the total usage was insufficient to warrant additional development efforts.

A long term goal for test officer support tools is to integrate a core system like TPD with specialized systems. Test officers would then have an aid to test planning, a system to draft a test plan, and access to the specialized expert system aids for those areas required by their test (e.g., security and environmental planning). The concept of a core system with specialized component subsystems should continue to be investigated using the techniques now available.

2.4 TRAINING ASSISTANCE

The Army provides several good AI training programs. However, to perform a continual technology insertion effort of AI at USAEPG requires more than just training. Both management and users need to be aware of the potentials and limitations of the technology.

2.4.1 <u>Three Year Perspective</u>. The need for training was recognized early in the first phase of this methodology investigation. "How to increase AI literacy" for an entire command was the question. The following sections describe several approaches for this.

    a. The basic need has not changed much over the three years. What has been learned is that this is a continuous effort for two reasons.

        (1) The technology is constantly changing. What may have been difficult or expensive in 1985 is now relatively simple and inexpensive.

        (2) Potential benefactors of AI technology may change jobs but keep their 1985 perspective about AI. Therefore, they may "write off" AI as a solution due to the 1985 view of the technology.

    b. During the 5 years of its existence, several individuals have rotated through the AI shop as apprentices (described below). The training provided by this methodology has allowed those people to learn AI and provide long term benefits for the Army. One person is now in charge of an AI cell in ISC, while another intends to pursue a master's degree and work at the AI center at the Pentagon.

2.4.2 <u>TECOM AI Seed Program And TECOM AI Support Center</u>.

2.4.2.1 <u>Introduction</u>. The need for training and supporting funds was recognized at other test centers, but it was not feasible to attempt the "USAEPG effort" at each test center, especially the smaller ones. To assist these activities TECOM created an AI Support Center to be headed up by USAEPG, and initiated a TECOM AI Seed Program for AI.

2.4.2.2 <u>AI Seed Program</u>. The TECOM AI Seed Program has been viewed as a 3-year program of decreasing funding starting in FY91. This program is designed to allow interested test centers a chance to pursue AI by providing some startup funding for projects, training, and software/hardware. The major emphasis for this program is on joint applications. Similar ideas proposed by two or more test centers are considered joint applications. One test center would then take the lead and build the first prototype which would then be customized for the next test center, and so on. This concept as discussed could be risky, especially for test centers new to AI or which have few personnel resources.

2.4.2.3 <u>TECOM AI Support Center</u>. The idea of a support center was born during the first years of this methodology, although TECOM funding for it would not become reality until FY91. The support center has been functioning for several years, in the sense that USAEPG has hosted workshops, made

software and textbook buys for test centers, held conferences, distributed information, and provided advice to several activities on how to get started. With all this happening, TECOM decided to formalize and fund the support center.

2.4.2.4 Support Center Activities. The TECOM Support Center will continue to provide workshops in easy-to-use AI shells, advice in implementation, mini-apprenticeships (2-4 weeks, with assistance, to create a prototype), purchases where necessary of software and textbooks, and knowledge for testing AI systems. The following are descriptions of the type of functions, activities, and information the Support Center has, or will, provide.

a. TECOM AI Conference. An annual AI meeting of the test centers to discuss and demonstrate their AI activities during the year, usually held in conjunction with the TECOM AI Task force meeting.

b. Workshops. A one week workshop, with brief hands on training session of an easy knowledge system shell. Attendees would be given the chance to build their own prototype applications for review and consideration for further development.

c. Mini-Apprenticeship Program. A 2-4 week visit by test center personnel ("experts") to get a quick start at building their own knowledge system. This is organized as an intense hands-on individual training session, with one-on-one assistance in design and implementation.

d. Apprenticeship Program. Similar to the above, but of a 2-4 month duration and designed to create a finished system. This program is designed primarily for USAEPG personnel, but other test center personnel are welcome.

e. Road Shows. These would consist of two of the TECOM AI Support Center personnel traveling to various test centers. This three-day visit would include a short introduction to AI; discussions of AI in the Army and TECOM; and an overview of local activities and personnel resources. The last day would include about four feasibility sessions where personnel would perform a problem assessment for potential applications.

f. Resource Center. This is a means to provide support at a low level and by correspondence. The resource center activities include consolidating purchases of software and reference materials for all test centers, maintaining a database and an inventory of professional papers, studies, and guides from which test center personnel may draw, and making tools, applications and advice available over the phone.

2.4.2.5 Results. The TECOM AI Seed Program funding will not begin until FY91, however, the support center effort has been on-going and accomplishments are as follows:

a. The support center has purchased copies and site licenses of the Expert System Development package (EXSYS) knowledge system shell for all TECOM test centers.

b. A set of AI reference books was also purchased and sent to the other test centers.

c. An information distribution list of pertinent information developed by TECOM and obtained by USAEPG from AMC, DA, and other AI activities has been compiled and updated so that upon request, items can be mailed to interested parties.

d. A guidebook for the development of small rule-based expert systems (including testing considerations) is being reviewed, and will be made available.

e. A workshop in CLIPS, a NASA developed shell, was held in August 1990.

2.4.3 Apprenticeship Program.

2.4.3.1 Introduction. Another technique to insert AI technology is through the use of an apprenticeship program. This concept forces training to be application oriented and allows for a useful tool to be developed as a result. For AI team members, the apprenticeship represents part of their initial training. For others, the apprenticeship affords them experience using AI techniques and an opportunity to develop a project.

2.4.3.2 Concept. An apprentice is temporarily detailed to the AI Office. This minimizes the interruptions which would occur if they remained in their regular assignment. Although the actual period of training can vary with the individual's ability and program goals, the average time requested is four months. At the end of this period the apprentice will be familiar with the basics of developing rule-based expert systems. At the end of the program, the trainee will have developed at least one prototype application to satisfy some need at their home office.

2.4.3.3 Approach. Whenever possible, the apprentice begins training by attending a two week course in basic expert system building and/or participating in local workshops. While outside training is generally available to most personnel, the apprenticeship offers a number of advantages. Most people who attend long courses on their own return immediately to their home office and spend their time trying to catch up on work they missed. By the time they get around to applying the techniques they learned in the AI course, much of the effectiveness of the training will have been lost.

In the apprenticeship program, students are able to learn new concepts and tools and immediately begin to apply this knowledge. Not only does this greatly improve the education process, but it allows more advanced techniques to be assimilated within a shorter time. Augmenting the basic training by actual experience with concepts merely touched upon in the basic courses allows the apprentice to build better systems, more effectively, when they return to their home office. If the apprentice is sent to AI training by their home office, a shorter detail could be provided to allow for the immediate application of the knowledge.

34

2.4.3.4 <u>Objectives and Structure</u>. In FY90, a more formalized approach was developed and tested for the apprentice program. Eventually it may be made available to other test centers through the support center in the form of "modules of instruction."

    a. The objectives in the formal program are as follows:

        (1) Describe the potential uses of AI.

        (2) Understand and use the capabilities and assets available through the AI Office.

        (3) Evaluate ideas for potential applications.

        (4) Prototype at least one idea using an expert system shell.

        (5) Give briefings and status reports on the implementation aspects of AI and related technologies.

    b. The program was broken down into phases of:

        (1) Orientation.

        (2) Problem definition and analysis.

        (3) Presentation to home organization.

        (4) Prototype application.

        (5) Summary and final presentation.

    c. These phases meet the objectives and place boundaries on the various efforts. This structure is being revised and refined, but will require an on-site mentor to administer the program, even once modules are complete.

2.4.3.5 <u>Conclusions</u>. While the apprentice's home office serves to benefit directly from the program, the AI office is also compensated. One of the goals of the AI office is to educate as many people as possible on the benefits and capabilities of AI. Apprentices help achieve this goal by serving as tutors of AI to members of their home office. Also, while an apprentice, a person will be assigned to participate in the development of an expert system or expert system tools which support current efforts of the AI team. The synergism provided by this program makes this a good approach for using the limited resources of an organization.

The apprentice program in FY90 was more structured and one person went through the program. This structuring include more front-end analysis of possible ideas for expert systems, application cost/benefit analysis, and presentation of findings to the AI team and management. Although prototyping was not initiated until late in the fourth month, the apprentice's home division chief

commented that the analysis was invaluable and that the individual had progressed in speaking, writing, analysis, and computer literacy skills and recommended this person for an award.

2.4.4 <u>In-House Workshops</u>. In-house workshops have provided training on specific rule-based tools. The objectives of the workshops were to familiarize personnel with the technology and to solicit ideas for further development.

In 1989, one workshop consisted of approximately ten students (or student/expert teams), each of whom built a small expert system as a class exercise. From these exercises, some were selected for development of a prototype system based on a management review of the projects. A side benefit was the exposure of both management and test experts to the capabilities and limitations of expert systems. This year one workshop was held in CLIPS, a tool developed by NASA.

2.4.5 <u>Team Training and Literacy</u>. An aspect of AI development methodology is the intense need to keep the literacy level as close as possible to the state-of-the-art. This has been difficult to do, but several techniques have been found. Reading articles on AI technology, attending conferences and seminars, learning and applying new shells, and providing consultation on testing AI are just a few of the ways we are trying to keep abreast in this rapidly growing field.

2.5 TESTING AI ISSUES

This effort was undertaken as a survey of existing and proposed techniques for testing knowledge-based systems (KBS). The intent of the survey was to identify available techniques, assess their relationship to currently defined software quality factors, and make recommendations for their development and application.

2.5.1 <u>Specific achievements</u>. Achievements this year have included:

      a. Initial survey of existing and proposed techniques.

      b. Submission of the final report on the above-mentioned survey.

      c. Presentation of survey results to the TECOM-sponsored Test Technology III symposium.

      d. Construction of a data base reflecting technique to quality factor relationship.

      e. Update of a previously initiated bibliography data base of materials related to such techniques.

      f. Participation in organization and conduct of a workshop on validation and testing of KBS conducted at the 1990 American Association for Artificial Intelligence (AAAI-90).

      g. Acquisition of a commercial knowledge base performance validation tool to assess its applicability to test and test technology projects.

2.5.2 <u>Three Year Perspective</u>. One of the initial efforts of this investigation was to utilize the investment of resources by stimulating outside interest in the problem of testing AI. Members of the AI team were instrumental in establishing a new workshop held in conjunction with the AAAI annual conferences, and devoted exclusively to test issues for knowledge based systems. Although much remains to be accomplished in the AI test arena, the interest generated by the workshop continues to provide valuable research within industry, academia, and other government agencies.

2.5.3 <u>State-of-the-Art</u>. In the past three to four years there has been a significant increase in efforts devoted to development of approaches and techniques for verification, validation, and testing (VV&T) of KBS. Three broad categories of effort have been identified to date. The first category consists of those projects aimed at defining the KBS life cycle and the role and form of VV&T appropriate within that context. The second category consists of projects aimed at developing high-level KBS system or subsystem assessments from some combination of objective, external, performance measures and subjective performance ratings. The last category consists of projects aimed at development of detailed and generally automated procedures for measurement of technical characteristics of KBS.

37

a. Projects in the first category have been only superficially examined. Since testing done by USAEPG occurs at specified points in an externally-specified life cycle, projects of this type have limited applicability. Their primary contribution is in identification of characteristics and criteria for KBS evaluation, and, in some cases, of applicable techniques.

b. Projects in the second category constitute the bulk of techniques immediately available for application. These techniques are drawn, with little or no alteration, from VV&T procedures for decision support and command and control systems. They require very little tailoring for application to KBS, and in many cases parallel techniques in use now at USAEPG. These techniques suffer from the drawback of being oriented towards evaluation of operational effectiveness, and provide little, if any, of the technical specificity required for developmental testing or reliability, availability, and maintainability (RAM) assessment.

c. Projects in the third category have the greatest potential for application to developmental and RAM-related testing. Most of them focus on static analysis of a KBS knowledge base, although a few do or soon will include consideration of inference engine characteristics as well. All of these projects suffer from three principle drawbacks. All are narrowly focused on a specific knowledge representation, in a manner analogous to the limitation of many static analysis tools for conventional software to a single language, or even a single dialect. All but two of the tools found thus far are research efforts and not generally available production quality tools. All but three of the tools exist independent of the development and maintenance environment, and hence require additional, ad hoc procedures to obtain the necessary source or other output for their application.

d. The allocation of each of the techniques examined to one or more software quality factors leads to the overall KBS VV&T state-of-the-art rating given below:

Factor	Degree of Attention
Correctness	High
Reliability	High
Efficiency	Medium
Integrity	None
Usability	Low
Maintainability	Medium
Testability	Medium
Flexibility	Low
Portability	Low
Reusability	Low
Interoperability	Medium

e. The more detailed survey results are included in the copy of the workshop paper (reference 4).

2.5.4 Actual Test of an AI System.

2.5.4.1 PRIDE Expert System. PRIDE is an expert system developed by the US Army Ordnance, Missile, & Munitions Center and School (OMMCS). It is designed to assist maintenance personnel in diagnosis and repair of selected faults in the HAWK Pulse Acquisition Radar. In July, 1990, a test effort was conducted by the OMMCS. It involved the diagnosis and repair of previously inserted faults by a junior maintenance technician. Senior maintenance personnel observed and controlled the exercise to maintain safety, critique the system recommendations and results, and to provide immediate correction of observed errors in tool or instrumentation use or test procedures. The most frequent criticism of the system was that the level of expertise implicit in the system dialogue proved more demanding of junior personnel than anticipated. On two occasions, the system was successfully employed to diagnose unplanned failures in the test systems. The PRIDE system is currently being used in Operation Desert Shield (reference 6).

2.5.4.2 USAEPG Consultation. The test itself functioned as a validation of the existing knowledge base and as a tool to refine and extend the knowledge base. USAEPG personnel recommended that issues and criteria be explicitly defined and that the conduct of the test be more stringent. Some illustrative candidate technical issues were proposed and a brief subtest was also prepared to illustrate how those technical issues could be incorporated as part of a traditional test plan (Appendix H). These issues were not exhaustive but are an example of the type of test issues that will be needed in the future. The candidate issues were:

    a. Unreachable Objects.

    b. Knowledge Audit Trail.

    c. Development/Run-Time Comparison.

    d. Variable Usage.

    e. Object Development History.

2.5.4.3 Lessons Learned. The OMMCS AI development team consisted of 1-2 in-house personnel and 1-2 contractor personnel. Many Army AI projects are of this level. Most of these AI development efforts have trouble with implementing the technology, availability of domain experts, user cooperation, and immaturity of the software and hardware. Testing considerations are usually last, and the team has little or not testing experience.

These individuals and projects should be assisted by experienced testing personnel on site. TECOM/USAEPG could provide this type of consulting service, but it will be necessary to obtain Army sponsorship either through the DA AI Center or AMC.

2.6 INTERFACE WITH OTHER ORGANIZATIONS

This project investigated the application of AI to the testing process. Interfacing with other organizations became important for two reasons.

    a. Due to the "newness" of the field, discussion with other AI organizations was needed to exchange ideas on techniques, AI packages, and pitfalls.

    b. Other functional organizations were contacted for sources of information, and as users of the systems.

2.6.1 <u>Three Year Perspective</u>. AI interaction steadily increased over the three years of the methodology effort as Army technology insertion initiatives began to take hold. These included the efforts of the DA AI Center, AMC, and TECOM's AI Task Force. Industry and the academic community efforts also increased with AAAI workshops on testing AI and panel sessions on testing began to appear.

Although the functional interaction was expected from the proponent due to the nature of AI, much interest was obtained from other functional areas. For example, USAEPG has presented the Environmental Assessment Aid twice at environmental conferences at the request of the Environmental Protection Agency (EPA). In general, this type of interaction has resulted in a number of "unusual" connections for a typical testing organization.

2.6.2 <u>TECOM Involvement</u>. TECOM's involvement in AI technology insertion has increased over the years. They have supported that insertion in a number of ways. TECOM is the parent command for USAEPG and the other test centers. When those test centers interested in developing AI systems contacted USAEPG for lessons learned and advice, we were designated, by TECOM, as the support center for AI within the command, providing planning functions and training such as the workshops, mentioned above. Further, TECOM appointed the chief of the USAEPG AI Office as technical agent for AI matters within TECOM, and to represent them at higher level meetings. This considerable commitment on the part of TECOM to share resources has helped extend the limited assets of the individual test centers.

2.6.3 <u>Knowledge Commonality</u>. A direct result of the action above is the emergence of efforts on the part of the test centers to create "multi-test center" expert systems based on knowledge commonality among organizations having similar functions. For example, all test centers have security and environmental offices that place requirements on a test. It appears that a "common" expert system built for one organization can easily be exported to another, or that using the knowledge acquired on the first system can reduce the effort for similar systems significantly. Unfortunately, at the level of detail required for successful expert systems, this "common" system may be very difficult to build. The value of the expert systems we now have are that they provide more than just general information for the novice. The regulations being followed by different test centers are the same, but the existing internal processes may be different, making the processes difficult

to standardize. The ability to generalize and create a TECOM system is still a worthwhile goal, but every system may not be a good candidate.

2.6.4 <u>Value of Interfaces</u>. The value of interaction has been in opening discussions among organizations. The following are examples of these discussions:

    a. TPD, although still in the prototype stage, has opened dialogue with several agencies, such as the Operational Test and Evaluation Agency (OTEA), concerning test planning.

    b. USAEPG has been nominated to a TRI-service Working Group on Automation of Test Planning.

    c. At one of the EPA conferences, EVA was reviewed in detail by a desert environmental expert and suggestions provided.

    d. Other organizations with award-fee structure contracts have shown interest in CPEA.

    e. One of our smallest systems, the budgetary interpreter, shows the ease of creating an expert system for understanding numbers on a spreadsheet and showed the value of expert systems to managers. In turn, USAEPG has received expert systems from other agencies. Although they are not always directly useable, the knowledge can be used to define "similar" systems.

2.6.5 <u>Other Approaches</u>. Exchanges between test centers sometimes include the entire organizational structure for handling AI. For example:

    a. Dugway Proving Ground (DPG) has integrated their AI shop with their MIS and Test Programming Development shop which is along the trend used by industry. This consolidation has helped them develop systems that cover a wide range of applications for DPG.

    b. Combat Systems Test Activity (CSTA) has a separate shop for AI but has combined a tester and a software developer as a team. This team has concentrated on using LISP to develop test conduct applications.

    c. DCSPAL has created a team to teach others how to do AI with a management group overseeing the development of the systems.

2.6.6 <u>Critical Elements</u>. All organizations contacted have shown a need for "critical elements." One critical element is the number of individuals needed to maintain the AI effort - usually 2-3 for a critical mass. Management interest and support is another, with the technical advisor often cutting through the existing structure to keep interest in the projects. Another is an "AI mentor," which usually takes the form of an existing organization with experience in AI, such as the DA AI Center or a support contractor that lends guidance to government personnel. These critical elements have kept many of TECOM's smaller test centers out of AI, or at least on the outskirts. They have also hampered some of the start-up efforts in other agencies.

2.6.7 Distribution System. The USAEPG AI office has created several reports and documents besides the yearly methodology report. The number and interest in those reports has prompted the development of a "distribution order form." Visitors to USAEPG's AI Office can receive the information above, as well ordering by mail. Most of the tools developed during the investigation are available to other Government agencies or their contractors. For current information on the availability of a tool, contact:

> U.S. Army Electronic Proving Ground
> Software and Interoperability Division
> STEEP-ET-S Artificial Intelligence Office
> Fort Huachuca, AZ   85613-7110
> (602) 533-8183/8187

2.6.8 Conclusions. All research and development efforts need interfaces with other researchers and organizations. The application of AI appears to obtain more value from these interactions than other areas for many reasons such as:

   a.   The newness of the field.

   b.   At the outset of this project there were relatively few interfaces in this field, especially in the test community applying AI.

   c.   Application areas are so specific that some questions needing answers can only be answered by other experts in the field.

SECTION 3. APPENDICES

APPENDIX A

METHODOLOGY INVESTIGATION PROPOSAL

A-1 TITLE. AI Test Officer Support Tool   28 August 1987

A-2 INSTALLATION OR FIELD OPERATING ACTIVITY. US Army Electronic Proving Ground, Fort Huachuca, Arizona 85613-7110.

A-3 PRINCIPAL INVESTIGATOR. Mr. Robert Harder, Software and Interoperability Division, STEEP-ET-S, AUTOVON 821-8187.

A-4 BACKGROUND. Test design and planning for modern C^3I systems require familiarity with a number of test operating procedures (TOPs) as well as detailed knowledge of specific test tool capabilities. A wide variety of tests must be designed, planned, and scheduled in order to efficiently conduct testing. Interrelationships among test groups and tools, common data requirements and data reduction and analysis requirements, lead time to prepare instrumentation, and required availability of the test item must be well understood in order to efficiently conduct tests within allocated time constraints.

USAEPG has explored the feasibility of an automated system to support the test officer. Using Independent Laboratory In-house Research (ILIR) funds, a prototype system was developed using AI technology. The prototype addressed tests performed by the Simulation and Interference Branch primarily, but could be expanded for other test areas.

A-5 PROBLEM. Testing C^3I systems involves designing and planning tasks which are becoming increasingly complex. Advances in technology such as microprocessor design, distributed real-time architectures, artificial intelligence, and electro-optics are appearing in new C^3I developments. While this sophisticated technology offers benefits to the developer, it is becoming a considerable burden to the tester. Test officers are required to identify appropriate test methods and associated instrumentation and data acquisition requirements for each emerging technological area. This requires a level of expertise which is rarely found in any one individual. Besides being distributed among individuals, and therefore less available, this hard-earned expertise is frequently lost to the organization because of personnel reassignment or attrition.

A-6 OBJECTIVE. To improve test methodology by providing the test officer with an automated support tool.

A-7 MISSION AREA(S) SUPPORTED. All DA mission areas for systems containing embedded computer resources (ECR) are supported. The "Big 5" program categories (C^3, RSTA, etc.) are accommodated by the nonsystem-specific nature of the methodology.

A-8 PROCEDURES.

a. _Summary_. The investigation will draw upon previous ILIR efforts by expanding the level of detail and the scope. The result will be an enhanced tool supporting the test officer in specific domains such as electromagnetic compatibility, software testing, and general test mechanisms. Other domain categories will be explored as time permits.

b. _Detailed Approach_. The USAEPG will:

(1) Extract and codify knowledge from cognizant individuals in fields including electromagnetic and software testing.

(2) Examine other test areas to identify tests performed, responsible branches, test instrumentation capabilities, and characteristic test requirements. Commonality among these various factors will be identified to form a framework which will accommodate all test functions, instrumentation, and resources. Following implementation of the generalized framework, specific test areas (knowledge domains) will be analyzed in depth and incorporated into the tool.

c. _Final Product(s)_.

(1) An AI test officer support tool with enhanced capability -- more "smarts" in the existing area of coverage, and additional test areas covered.

(2) Requirements and recommendations for automation of test design and planning functions.

d. _Coordination_. Extensive coordination with the various test groups of the USAEPG is an inherent characteristic of the investigation. To the extent that test areas covered overlap the areas of interest of other I/FOAs, coordination will be accomplished through existing mechanisms such as the TECOM Software Technical Committee (TSOTEC).

e. _Environmental Impact Statement_. Execution of this task will not have an adverse impact on the quality of the environment.

f. _Health Hazard Statement_. Execution of this task will not involve health hazards to personnel.

A-9 JUSTIFICATION/IMPACT.

a. _Association with Mission_. This investigation directly supports USAEPG's mission relative to test and evaluation. Providing test officers with automated support tools will improve the efficiency and effectiveness of testing.

AI Test Officer Support Tool (Cont)

b. Association with Methodology/Instrumentation Program. This project
supports thrusts of the TECOM Methodology Program to improve the quality of
testing as well as test process.

c. Present Capability, Limitations, Improvement, and Impact on Testing if
not Approved.

(1) Present Capability. TOPs and guidelines, such as the
USAEPG Test Officers Handbook, provide static information on test methods and
checklists for test planning purposes.

(2) Limitations. Current guidelines often do not provide
the level of detail required for optimized application of scarce test
resources. Also, the information is static; status of test instrumentation,
competition for resources among different test items, and the impact of not
performing some test (or lack of test material such as certain documentation)
is poorly handled unless the test officer's experience has included similar
situations.

(3) Improvement. Using AI techniques to develop a support
tool can provide the test officer sufficiently detailed and flexible
guidelines. Beside being adaptable to the needs of a specific test item and
current with respect to test instrumentation availability, the proposed
approach would be sensitive to data requirements and be able to anticipate the
impact if tests are not performed. Supported over time, such a tool could
accumulate expertise which is presently distributed and too frequently lost.

(4) Impact on Testing if not Approved. The expertise
required of test officers is rapidly expanding in scope as innovative
technologies are increasingly employed by developers. The corresponding
increase in complexity of test methods and instrumentation demands a
commensurate improvement in support tools if test resources are to be
effectively and efficiently used. Also, without permanent storage and readily
available access to "lessons learned", the corporate memory of an activity
suffers each time an experienced individual leaves the organization.

A-10 DOLLAR SAVINGS. No directly supportable dollar savings can be projected
at this time. Indirect benefits include improving the quality of testing and
evaluation leading to improved quality of fielded systems. Equally difficult
to quantify is the retention, concentration, and increased availability of
expertise, which is potentially a significant amount.

A-11 RESOURCES.

a. Financial.

	Dollars (Thousands) FY88		Dollars (Thousands) FY89	
	In-House	Out-of-House	In-House	Out-of-House
Personnel Compensation	10.0		12.0	
Travel	3.5		4.0	
Contractual Support		84.5		42.5
Materials & Supplies	2.0		1.5	
Subtotals	15.5	84.5	17.5	42.5
FY Totals	100.0		60.0	

b. Explanation of Cost Categories.

(1) Personnel Compensation. This cost represents compensation chargeable to the investigation for using technical or other civilian personnel assigned to the investigation.

(2) Travel. This represents costs incurred while visiting government and industry facilities.

(3) Contractual Support. Performance of the investigation will be accomplished with resources provided under an existing support contract.

c. Obligation Plan (FY89).

FQ	1	2	3	4	Total
Obligation Rate (Thousands)	45.0	5.0	5.0	5.0	60.0

d. Man-Hours Required.

In-House:
Contract:

AI Test Officer Support Tool (Cont)

A-12 ASSOCIATION WITH TOP PROGRAM. This investigation will not directly
produce a TOP. However, various TOPs may require review and revision based on
the findings.

FOR THE COMMANDER:

                              (signed)
                              ROBERT E. REINER
                              Chief, Modernization and
                              Advanced Concepts Division

This Page Is Intentionally Left Blank

APPENDIX B

REFERENCES

1. Methodology Investigation for Software Technology for Adaptable, Reliable Systems (STARS), Final Report: Test Methodology with Artificial Intelligence, Harder, et al., U.S. Army Electronic Proving Ground, Fort Huachuca, Arizona, March 1987.

2. Methodology Investigation Final Report, AI Test Officer Support Tool, Volume I, U.S. Army Electronic Proving Ground, Fort Huachuca, Arizona, March 1989.

3. Methodology Investigation Final Report, AI Test Officer Support Tool, Volume II, U.S. Army Electronic Proving Ground, Fort Huachuca, Arizona, April 1990.

4. N. Sizemore, Test Technology for Knowledge Based Systems: A Survey. Proceedings of IJCAI-89 Workshop on Verification, Validation, and Testing of KBS, August 1989 [in preparation].

5. AI Exchange, Vol IV, No. 2, Spring 1990

6. AI Exchange, Vol IV, No. 4, Fall 1990

7. Software Package for ESGUIDE, U.S. Army Electronic Proving Ground, Fort Huachuca, Arizona, Sep 1990 (currently under development).

Note: Domain-specific references are available upon request.

This Page Is Intentionally Left Blank

APPENDIX C

ACRONYMS AND ABBREVIATIONS

The following is a list of acronyms and abbreviations used in the document.

AAAI........... American Association for Artificial Intelligence
ACM............ Association for Computing Machinery
AI............. Artificial Intelligence
AMC............ United States Army Material Command
AR............. Army Regulation
ASL............ Atmospheric Sciences Laboratory
BUD2........... Budget Spreadsheet Analysis Aid Expert System
C^3.............. Command, Control, and Communications
C^3I............ Command, Control, Communications and Intelligence
CAA............ Concepts Analysis Agency
CASE........... Computer - Aided Software Engineering
CLIPS.......... C Language Integrated Production System
CLOS........... Common Lisp Object System
CPEA........... Contract Performance Evaluation - Advisor Expert System
CSTEA.......... Combat Systems Test Activity
DA............. Department of the Army
DoD............ Department of Defense
DPG............ Dugway Proving Ground
DTP............ Detailed Test Plan
ECR............ Embedded Computer Resources
EPA............ Environmental Protection Agency
ES^2............. Expert System Selector
EVA............ Environmental Impact Assessment
EXSYS.......... Expert System Development Package
GIS............ Geographic Information Systems
GUI............ Graphical User Interface
HFE............ Human Factors Engineering
I/FOA.......... Installation/Field Operating Activity
IJCAI-89....... 1989 International Joint Conference on AI
ILIR........... Independent Laboratory In-House Research
KBS............ Knowledge Based Systems
LAN............ Local Area Network
MET............ Meteorological Expert System
MIL............ Military Standard
MS DOS......... Microsoft Disk Operating System
NASA........... National Aeronautics and Space Administration
OGL............ Overall Grade Level
OMMCS.......... Ordnance, Missile and Munitions Center and School
OTEA........... Operational Test and Evaluation Agency
PC............. Personal Computer
PRIDE.......... Pulse Radar Intelligent Diagnostic Environment
PT............. Physical Training
RAM............ Reliability, Availability, and Maintainability
RDI............ Research and Development Instrumentation
REC............ Record of Environmental Consideration

C-1

RSTA............ Reconnaissance, Surveillance, and Target Acquisition
SAA............. Software Analyst's Assistant Expert System
SBIR............ Small Business Innovative Research
SIMA............ Systems Integration and Management Activity
S&I............. Software and Interoperability Division (USAEPG)
SPA............. Security Planning Aid
STARS........... Software Technology for Adaptable, Reliable Systems
TECOM........... United States Army Test and Evaluation Command
TM.............. Technical Manual
TOP............. Test Operating Procedure
TPD............. Test Plan Drafter
TQM............. Total Quality Management
TSOTEC.......... TECOM Software Technical Committee
TTES............ Tape Test Expert System
UAV............. Unmanned Aerial Vehicle
USAEPG.......... United States Army Electronic Proving Ground
VV&T............ Verification, Validation, and Testing

APPENDIX D

DOCUVIEW HYPERTEXT TOOL

D-1 PURPOSE/GOALS.

The intended use of DocuView is for displaying general textual information on a computer screen. Hypertext expansion techniques are used for highlighting certain phrases within a document. Through selection of these phrases, such techniques allow a nonlinear traversal of the document.

D-2 DOMAIN/EXPERTISE.

This software program is capable of being used wherever documents or general text materials need to be separated into pages for display purposes. The document analyst, through various commands embedded in the text, has the flexibility to present the information in the most suitable manner for the particular domain being handled. The user, based on actual needs during presentation, has the control to dynamically alter the order in which the material is viewed. Both the analyst and user play a key role in the assimilation of hypertext information.

D-3 REQUIREMENTS.

This type of software tool is needed so that documents residing on the computer can be broken into logically defined pages for presentation as windows on a computer display screen. Documents must be stored in a form which allows modification by most text processors, yet must also be directly presentable by the document viewing tool.

D-4 DESCRIPTION.

The DocuView tool is a software package consisting of a main program and numerous subprograms and functions written in a conventional computer language. The software is designed to present the contents of a document file, referred to as an object file, on a microcomputer display screen in user-specified pages. Each page is defined to have its own window at a chosen location on the screen, and has a set of parameters which specify text and background color, window size, and other options. Pages are inserted into a text file by the addition of DocuView command lines. Other command lines are entered into the text to signify selected states or state changes. These commands make it possible for the DocuView user to work with varying document types and contents without experiencing conflicts between commands and the textual contents. The command words and delimiters used in the text are changeable as needed by the user. As an example, the exclamation point character used to delimit highlighted phrases can be changed to some other character when conflicts in the document text arise.

D-5 DESIGN/DEVELOPMENT CHARACTERISTICS.

The most significant feature of DocuView is that it allows the document being analyzed to be broken up into pages for presentation on a computer display screen, and that on these pages of text, chosen phrases can be highlighted for hypertext expansion into still more pages of commentary or description. The display of pages and hypertext expansion of selected phrases can be done recursively for page after page of textual information.

D-6 BENEFITS.

Information now stored on computer media or available in such form can be conveniently displayed on a computer screen. No significant changes to the original document are necessary. At the same time, any text processing of the document is readily achievable with conventional text editors. The real benefits, though, are to be realized with the display of only pertinent information and the resulting improvement in assimilation by the user of new information.

D-7 USE.

    a. DocuView has been used in a number of applications.

       1) One application is documentation of Test Issues for AI. The documentation includes over 200 Kbytes of detailed results. The system gives access to whatever portion of these results a person wants. Access is in four dimensions usable in a few seconds according to factor, subfactor, component, and methodology.

       2) A README file was prepared in DocuView for distribution with a software package. The package is used to prepare test plans. The information covers such topics as computer setup and software installation.

       3) A bibliography of AI publications has been produced using the hypertext capabilities of DocuView.

    b. Some of these applications of DocuView have been sent to various offices. Selected applications have been sent to the Pentagon AI Center, OTEA, Office of the Secretary of Defense, and the Office for Aviation Development and Test.

    c. Another use felt to be a good application for DocuView is the display of the Test Officers' Handbook. A user would be able to view only the information pertinent to a given need or test application.

D-8 DEVELOPMENT STATUS.

Development has reached a stage where an initial version of the DocuView tool was distributed at one of the AI workshops and to other interested parties. Currently, comments received from formal and informal reviews are being incorporated into a new version. Possible improvements include increasing the types of parameters which are user-defined, providing a selective print

function, and allowing easier use through such features as automatic sizing of text to fit a window.

D-9 AVAILABILITY.

DocuView Version 2.1 is currently available to interested government offices. Copies of the software can be obtained through:

U.S. Army Electronic Proving Ground
Software and Interoperability Division
STEEP-ET-S Artificial Intelligence Office
Fort Huachuca, Arizona 85613-7110
(602) 533-8183/8187

This Page Is Intentionally Left Blank

APPENDIX E

ENVIRONMENTAL IMPACT ASSISSMENT EXPERT SYSTEM.

E-1 PURPOSE/GOALS.

The purpose of EVA is to assist the test officer and environmental personnel in collecting accurate environmental information during the early planning phases of test activities, and in making appropriate recommendations based on characteristics of the proposed activities. Specific goals of the system were to:

     a. Identify tests with minimal or no environmental impacts, and streamline the documentation process.

     b. Identify possible environmental impacts and the resources that could be affected (e.g., water, wildlife, cultural, historical).

     c. Improve the quality, detail, and timeliness of information provided to environmental personnel during the initial stages of a test project.

     d. Incorporate environmental information into the initial decision-making stages of a project.

     e. Guide activity proponents through the environmental assessment process, and list points of contact for action items and regulatory requirements.

E-2 DOMAIN/EXPERTISE.

The domain of EVA covers that area of knowledge required to identify potential environmental impacts, recognize categorical exclusions from the rules for certain damaging activities, and perform a preliminary screening to determine the probable environmental documentation requirements. This expertise resides with the USAEPG environmental quality coordinator and environmental specialists attached to the post garrison. These experts in turn consult specialists in more narrow domains when necessary.

As EVA evolved through various prototype stages, additional information from documented sources was incorporated into the design. This information consisted more of quantitative impact factors, rather than intuitive knowledge about the domains. The inferences about this data were supplied by the human domain experts.

At the end of prototype development the following sources had been used in generating the data bases and rules of EVA:

     a. U.S. Army Corps of Engineers

         (1) Construction Engineering Research Laboratory reports

        (2) Archaeologist

    b. U.S. Department of Agriculture, Soil Conservation Service

    c. U.S. Army Environmental Hygiene Agency

    d. Fort Huachuca

        (1) Forester

        (2) Wildlife Biologist

        (3) Fish Biologist

        (4) Environmental Specialist

Much credit is due the post environmental specialist for identifying sources of information and eliciting knowledge from subdomain experts. This effort exceeded the scope of the normal participation of an expert, and aided tremendously in knowledge acquisition activities.

E-3 REQUIREMENTS.

USAEPG is required to conform to federal and state environmental regulations as well as Army and DoD policy in these matters. Every proponent of an exercise or test at Fort Huachuca is required to address the environmental issues associated with the activity. USAEPG test officers have the additional responsibility for assessing potential environmental impacts for any activity resulting from a test directive, regardless of the nature of the testing.

The result of the preliminary impact assessment is a record of environmental consideration (REC). The REC documents the consideration of environmental impacts; possible outcomes are that the activity is adequately covered by existing documentation, qualifies for an established categorical exclusion or other exemption, or requires an environmental assessment. Environmental assessments subsequently result in a "finding of no significant impact" or indicate that an extensive environmental impact statement is required.

Most of USAEPG's activities are conducted at locations specifically designated and documented for that type of activity, or are conducted entirely within an enclosed facility, as such computer simulation and modeling. The major requirement of a preliminary environmental screening is to discriminate as early as possible between typical situations requiring little further documentation, and those requiring a significant environmental impact study.

E-4 DESCRIPTION.

The EVA elicits information about a proposed activity from the test officer, and reaches a preliminary conclusion on the actions required. It then generates a report containing action items, and summary and detail characteristics of the activity, with corresponding environmental impacts. Activities which have already been documented or qualify for a categorical

exclusion are quickly identified (i.e., a minimum of user input is required), and the necessary REC report is generated. For activities where the potential environmental impact is greater, the user may elect to examine the environmental resources most affected and, if possible, modify characteristics of the proposed activity to minimize the impact and associated documentation. In any event, information from the report is used by the environmental quality coordinator in completing the environmental requirements.

E-5 DESIGN/DEVELOPMENT CHARACTERISTICS.

The EVA system consists of an expert system which provides the user interface, contains the rules used to make decisions, generates reports, and interfaces with other tools for additional capabilities. These other tools supply such functions as access to data bases and graphic display of map information. Other components include supporting information such as help, system parameter, map, point-of-contact, and report specification files. The expert system shell, EXSYS, allows a means to interface with the other tools and files so efficiently that the user is generally unaware of the individual components. To further isolate the user from having to contend with directory structures and operating system commands, a set of command files was created to simplify the installation and operation of EVA. The user merely enters one command to run the system and display and print the results.

The main expert component of EVA contains about 120 IF-THEN rules in the knowledge base. When processed by the EXSYS inference engine, the rules serve to collect the necessary information to reach the final conclusion on the environmental impacts of the proposed activity. Forward chaining, a technique which determines how the rules are processed, also allows some control over the sequence in which events take place. The user can be presented with queries in the same relative order, even though the knowledge base and supporting data bases may have changed from previous versions.

Although all of the rules may apply to a given scenario, only those which rely upon unknown information will request the user to enter needed data. Besides background information such as project number and description, which are always requested, firing (processing) of the rules may trigger queries on up to 150 numerical or textual variables, and up to 35 multiple-choice questions. For example, if the activity will include aircraft, then information is requested on the number of aircraft, number of flight hours, and time of day and altitude of the flights. Because only essential information is requested, an EVA session can last anywhere from 5 to 45 minutes.

Part of the development philosophy was to minimize the amount of knowledge to be included in the rules about a specific installation. Information on the location of sensitive resources, period of sensitivity if not constant throughout the year, and qualitative damage factors associated with particular activities, were placed in ten data bases. These data bases were designed to be readily understood and modified by the domain experts without first having to obtain knowledge engineering skills. Likewise, user help screens, point-of-contact information, etc., which contained installation-specific material, were kept in separate files. This approach may provide a ready means of porting the system to other installations, but was chosen primarily to reduce

development and maintenance costs. Information contained in the various data bases and files could have easily been encoded into rules, and some expert development packages provide the capability to do just that when fed tabular data. The problem with a pure expert system solution, (with all of the knowledge embedded in rules), concerns the size of the resultant rule base. It was estimated that to incorporate the knowledge in the data bases alone into rules would add another three to four hundred rules. Further development and maintenance of such an unwieldy knowledge base would have significantly impeded progress, with no known advantages.

E-6 BENEFITS/USE.

EVA offers benefits to the test officer, environmental quality coordinator, and program manager. Test officers are given the opportunity to compare environmental effects of different activities at various locations and times. With little prior knowledge of environmental concerns, the test officer using EVA can quickly gain an appreciation of the relative impact of various activities through the questions asked, the associated help text, and the outcome of the proposed scenarios. Less experienced test officers also benefit from the action items and notes related to the proposed activity; e.g., contacting the fire marshal and filing a fire plan if incendiary devices are used, or coordinating tree and brush removal with the post forester. These serve as reminders even for seasoned test officers, and both inexperienced and experienced users of the system benefit from reduced paperwork and coordination.

EVA does not make complicated environmental decisions, write environmental assessments or environmental impact statements, or replace environmental personnel. In fact, environmental quality coordinators themselves can use EVA to refine the work initiated by test officers, or as a method of automating and documenting activities in a standard fashion. Tests with minimal environmental impact are identified with a savings of paperwork and time. Even for large activities not fully handled by EVA, the quality, consistency, and detail of information presented to environmental personnel is greatly improved. Without EVA, many preliminary meetings are required between the test officer and environmental quality coordinator, merely to establish what information is needed, and then the data is rarely available in an organized format.

Sponsors of testing activities may gain the most from the use of EVA, albeit somewhat indirectly. Because extensive environmental documentation requirements can cause lengthy and expensive delays, it is important to identify potential impacts as early as possible, and develop alternative test scenarios which are more environmentally benign. Advance warning of potentially expensive activities, such as disposal of hazardous materials (e.g., expended batteries), may, if given in time, allow implementation of more cost-effective solutions.

E-4

E-7 DEVELOPMENT STATUS.

EVA is currently installed on several microcomputer systems at Fort Huachuca; about 20 test officers have been formally trained in its use. Presently the system is in an evaluation phase, where feedback is being obtained concerning its use in test operations.

E-8 FUTURE.

A number of ideas for further development of EVA have been proposed. During its construction, the development team identified a number of desirable features which could not be implemented because of time constraints. Other valuable ideas emerged from the test officer training sessions. The actual usefulness and benefits to be realized must be determined from the results of the ongoing evaluation. Some of the more significant limitations and improvements to be considered in future efforts are the following:

    a. Some of the knowledge in EVA is in a preliminary state, having been added to determine the feasibility and desirability of certain features (e.g., a component to address hazardous materials). Those features deemed desirable should be expanded, along with the rest of the system, into a fully operational form.

    b. The potential for porting the system to other installations should be explored further. This would require an initial analysis of the requirements of other installations, to see if enough commonality exists in the knowledge domains to make this approach feasible. Such an investigation might also shed some light on the commonality of other requirements, such as test resource management and safety.

    c. The prototype system has the limitation that only one map area can be entered as the location of activity. Although areas may be arbitrarily defined as large or small as desired, a cumbersome situation occurs with activities consisting of 100 or more sites with minimal impact at each location. Even smaller activities may be handled better if multiple locations, or if unrestricted boundaries are allowed.

    d. A feature which would allow saving all of the input information, to be used later to examine the impact of different test scenarios, is desirable. Such a capability was partially implemented, but had to be disabled because of a software discrepancy in the expert system tool. Along these same lines, many users expressed the desire to be able to modify an entry that had just been made. Both seem to be necessary features for practical use in an operational environment.

    e. Most of the data bases of EVA are indexed by location. Geographic information also plays an important part in many other functions at Fort Huachuca. A solution to many of these needs for information associated with geographic position would be a geographic information system. This is also a requirement of many other proposed test tools. While implementation and maintenance of such a system is well beyond the scope of this investigation,

the potential usefulness is great enough to warrant development by other means.

       f. The actual users of EVA range from inexperienced test officers to qualified environmental personnel. Because of the disparity in experience, a system tailored to a given skill level will be somewhat frustrating for users of a different level. Experienced users quickly tire of a system oriented toward the novice, while inexperienced users may find a system written for the expert to be much too difficult. A possible solution to this dilemma was discovered during the EVA development, but too late to fully evaluate. Basically, this approach, if implemented, would call for multiple levels of rules, help, and queries. A "don't understand" option is provided on higher level queries. When invoked by the novice, this option fires lower level rules which elicit a number of simpler details from the user. These details are then formulated by the lower level rules into facts which satisfy the original, "difficult" query. Such an approach is best implemented on mature knowledge bases because of the growth in size and commensurate decline in maintainability. For a system with a diverse user base, further examination of this technique may prove useful.

APPENDIX F

TEST PLAN DRAFTER

F-1 PURPOSE/GOALS.

The basic function of the TPD is to automate the current manual assembly of boilerplate for an initial draft of a test plan. This is a time-consuming effort consisting of much cut-and-paste work from old test plans, but little real intellectual effort. It is intended to result in a strawman version for distribution to specific subtest domain experts for further editing.

In addition to its basic function, TPD provides a framework for maintaining three types of information. These types are generic composition of test plans, test requirements for a specific item, and general information needed in test planning, such as EPG's organizational structure of domain experts.

Using TPD's framework of information, a prototype expert system, called coordination notes, has been linked with TPD and successfully used. Although the value of coordination notes to test officers is limited, the primary value of coordination notes is that it shows how expert systems could be integrated with TPD.

F-2 DOMAIN/EXPERTISE.

The primary domain of the TPD is to assist test officers in producing test plans. The initial sources of knowledge about test plan composition and the overall test and evaluation process were Army, AMC, TECOM, and USAEPG publications. Further details were provided through review of local policy, interviews with test officers, and interviews with USAEPG's Technical Publications Division, whose personnel prepare and publish test plans. Also, previously drafted subtest plans were acquired and reviewed.

     a. Test planning is becoming increasingly complex. It requires familiarity with test operational procedures TOPs and detailed knowledge of specific test tool capabilities. It requires technical knowledge of the systems being tested and their test requirements, specifications, and issues. It requires knowledge of generic composition of test plans and other general information. Test planning requires consideration of the availability of the test items, lead time required to prepare test instrumentation, and common requirements for data reduction and analysis.

     b. TPD's emphasis on assisting the test officers is well placed. Three facts support this:

        (1) Test officers need something to assist them with the growing complexity of their tasks.

        (2) Nearly all test officers have the computer literacy required to use TPD.

(3) The percentage of test plans that are published in the test officers division is increasing.

Furthermore, because of TPD's potential to raise the general level of test officer competence in handling the complexity of their tasks, TPD could contribute to consistent excellence in test planning.

   c. Before TPD can make a significant contribution toward this goal, it needs to be integrated with an expert system that prompts test officers through the process of transforming generic test plans to system specific test plans. More specifically, before development of TPD would be complete, it needs to be integrated with a system that addresses the operational part of test planning. The recently appointed TPD proponent claims test plans contain two parts: individual subtests, which TPD addresses, and an operational part. The operational part contains both procedures to check proper operation of a single unit (i.e. for testing the environmental specifications), and scenarios to check operation of a group of units (i.e. for testing higher level evaluation issues).

F-3 REQUIREMENTS.

The general requirement was that TPD be of wide utility and also aid in training of personnel. The specific requirements were that it reduce the manual and telephonic work required to reach the strawman stage for a test plan, that it provide information on test plan structure and component descriptions, and that it assist the novice test officer in understanding the test and evaluation process. These capabilities have been demonstrated.

F-4 DESCRIPTION.

TPD organization and operation is summarized below. Inputs and outputs are described briefly.

   a. Inputs.

      (1) TECOM project number. This software helps the test officer add the TECOM project number to TPD.

      (2) Test information. This software helps the test officer add information to TPD, including item nomenclature and type of test.

      (3) Plan administrative information. This software helps the test officer add information to TPD, including the test officer's name.

      (4) Agency information. This software helps the test officer add information to TPD, including the name and address for agency sponsoring the test.

      (5) Subtest selection. This software helps the test officer specify which subtests are required or excluded.

(6) Appendix selection. This software helps the test officer specify which appendices are required or excluded.

    b. Outputs.

(1) Administrative details. This software produces a paper sheet that shows, for a specific subtest plan, the test officers name and the date of the last TPD update.

(2) TECOM project number breakout. This software produces a paper copy of a chart that shows what the digits in the TECOM project number mean.

(3) Subtest status chart. This software produces a paper copy of a chart that shows which subtests are required by who and which subtests are excluded.

(4) Coordination notes. This software creates three files. Upon a second user input, EXSYS is loaded and used to execute an expert system, (coordination notes). The resulting file is automatically sent to the DOS operating system and printed. The printed result provides test officers with information about which subtests require coordination and it provides references so the test officer can read more about what is required.

(5) Cover sheet. This software produces a paper copy of the test plan's cover sheet based on information provided by the test officer.

(6) Table of contents. This software is not yet developed.

(7) Introduction. This software is not yet developed.

(8) Details of Subtest(s). This software produces paper copies of all the subtests which the test officer has specified are required. Also, this software can link to DocuComp to compare a subtest with its generic version.

(9) Subtest floppy disks. This software copies the generic versions of required subtests to floppy disks. Because TPD contains information on which office is responsible for each subtest, this software produces one floppy for each office with that office's subtests.

(10) Appendices. This software is not yet developed.

F-5 DESIGN/DEVELOPMENT CHARACTERISTICS.

The initial TPD prototype consisted of a data framework (dBASE III and hypertext files), software to create and maintain test plan information, and software to support help and explanation functions. The dBASE III tool software drives the application. In this environment, changes in the data framework require knowledge of dBASE.

TPD was targeted for use on the microcomputers that are widely available to test officers. These microcomputers varied widely in configuration (DOS

version, amount of memory, and floppy drives). Since TPD has several functional pieces and provides a data framework that can be used by expert systems such as coordination notes, some constraints were found concerning which TPD functions were compatible with the diverse configurations.

Alternatively, these difficulties could be overcome by targeting TPD for use on a local area network. TPD could be installed on a single network server, which would avoid the difficulties caused by installation of TPD on diverse microcomputers. Through the local area network, TPD could be available to nearly all test officers.

Two AI-related tools have been linked to TPD. A hypertext tool provides help and explanation functions to TPD users, using the hypertext paradigm. This tool links each screen of information to related screens of information. In this way, information can be displayed from only one level, while hiding information at lower or higher levels unless the user chooses to look at them.

A rule-based EXSYS supports development and use of expert systems. For example, EXSYS supported development and use of coordination notes, an expert system that is linked to TPD. EXSYS could support additional expert systems, which could also be linked to TPD.

One further tool, DOCUCOMP, from Advanced Software, Inc., has been linked to TPD, and provides a document comparison facility for identifying changes made to a standard subtest to tailor it for a specific system. This could provide the test officer with limited assistance in the test plan review process.

F-6 BENEFITS/USE.

One of TPD's benefits is quicker production of test plans. Current users and others to whom TPD has been demonstrated indicate that the current manual method of strawman draft plan composition can take from two days to two weeks. TPD can be used to produce strawman subtest plans in one hour or less. While the resulting product is not complete, it accounts for perhaps as much as 40 percent of the content of such a strawman. Some increase in this percentage will accrue from growth in the archive of generic subtests, while some increase must await implementation of further functions.

Another of TPD's benefits is training and standardization of the test planning process. TPD's help and explanation functions provide information previously available only by experience, word of mouth, or perusal of regulations and pamphlets. Also, TPD indicates sources for further information. Moreover, while the diverse backgrounds and levels of experience among test officers have sometimes led to irregularities in subtest format, more widespread use of a single tool offers the promise of improved adherence to TECOM and local guidance with less administrative review and rewriting effort. This could allow test officers, test engineers, and managers to devote more of their time and effort to substantive test issues.

Yet another potential benefit is helping test officers deal with the increasingly complexity of their test planning mission. The combination of TPD's basic data framework and its proven capability to link with expert

systems offers the promise of a tool that can deal with complexity in a way that is usable for test officers.

Previous to the fourth quarter of FY 90, the TPD prototype had been used in production of several strawman draft test plans by test officers in the Command and Control Division of the C³ Test Directorate. These test officers made several suggestions for improvement.

The fourth quarter of FY 90, a TPD instructor helped five test officers use TPD. These test officers made positive comments about the usefulness of TPD. Also, they talked about the existence of other tools that could help them produce quality test plans. However, despite the perceived usefulness of a completed TPD, the frequency of use was estimated to be relatively low.

F-7 DEVELOPMENT STATUS.

TPD's prototype functionality is largely complete and reached the point where a decision was made on whether to devote the resources necessary to produce a production version. Because of the priority of other proposed projects and the projected frequency of use, further development of the current prototype was postponed. The structure and operational characteristics of the system have much in common with related applications though, and consideration has been given to adopting the TPD design to satisfy part of the requirements of other proposed systems.

F-8 FUTURE.

If resources permit further development and the anticipated usage justifies completing a production version TPD, three major lines of development are foreseen. The first line could be to increase the use of TPD in order to obtain additional information concerning what test officers require. Increasing the use of TPD would involve making TPD easier to learn and understand, and assisting experienced test officers in using TPD. These accomplishments would require actions like writing a users pamphlet and sending a TPD instructor to work one-on-one with test officers. If, as anticipated, a local area network becomes widely available to test officers, a further incremental increase of TPD use could be achieved by placing TPD on a network server.

The second line of TPD development could be to expand and refine the knowledge-based portion of the system, i.e., the hypertext and expert system components. This line offers potential for increasing the support for test officers, such as assisting them in drafting the operational part of test plans.

The third line of development could be to expand TPD's capabilities. New expert systems could be developed and targeted for use with TPD. Both these new systems and existing expert systems that have proved their value could be integrated with the TPD. This might require expanding the TPD data framework and modifying the shell for passing information from the TPD data framework to expert systems. The long term goal of this line could be integration of several support tools into a single package for use by test officers.

This Page Is Intentionally Left Blank

APPENDIX G

METEOROLOGICAL EXPERT SYSTEM

G-1 PURPOSE/GOALS.

The Meteorological Expert System (MET) began originally as a manual paper checklist for test officers to use in preparing for upcoming tests at Fort Huachuca. It is designed to emphasize the need for meteorological data in planning and reporting tests within USAEPG. MET also indicates that various meteorological measurements and advisories are available from the Atmospheric Sciences Laboratory (ASL) weather station at Fort Huachuca, and from other sites located on the Fort Huachuca ranges.

G-2 DOMAIN/EXPERTISE.

This expert system deals with the knowledge encompassing meteorological measurements and/or those weather events which affect test operations on the ground or in the atmosphere where testing will take place. Generally these measurements or observations are provided by ASL.

G-3 REQUIREMENTS.

From the standpoint of the test officer, the need for an expert system on weather is that it can educate and inform the test officer about meteorological data requirements and available resources for a test. The need for such data comes primarily when the test will be conducted outdoors. The expert system will make clear that the officer will need to have weather predictions before the test in order to plan for conditions such as cold or heat, rain or snow, and wind or lightning. Weather advisories and weather alerts from ASL can warn the test officer in the field of impending sudden weather changes that could endanger personnel and equipment.

G-4 DESCRIPTION. The MET system educates the test officer as to possible weather-related needs, and informs the officer on how to obtain needed measurements to prepare for the test, how to run the test more effectively, and how to obtain weather station support in reporting the test outcome.

Measurements and predictions of temperature, dew point, rain, snow, thunderstorm activity, and winds in the lower atmosphere, may be needed. Predictions may be needed as to meteorological conditions such as sunspot activity and atmospheric index of refraction. MET informs the test officer whether, during on-site test activities, weather advisories and reports of selected meteorological values are available and may be needed. Also the ASL weather station's ability to support test reporting is covered.

The result of using the MET system is that the test officer can produce better test data by being prepared with needed meteorological data, both in measurements that directly supply parameters needed in the calibration of

equipment such as radar, and in supplying measurements for the test, as well as weather advisories that assist in day-to-day running of test operations.

Without MET, the test officer must know to inform ASL of test requirements far enough in advance to prepare them to supply information needed for the test. ASL may need to prepare ahead of time to be able to make measurements during the test, and will need to know what data are needed for the test report. ASL can supply reports of the meteorological conditions that existed during testing.

G-5 DESIGN/DEVELOPMENT CHARACTERISTICS.

The MET system is composed of a series of questions which are presented to a test officer from within the EXSYS shell. The questions asked in this prototype version of MET determine, for example, whether lasers will be used in the atmosphere, whether any radar or unmanned aerial vehicles (UAVs) will be used, whether personnel and/or equipment will be in the field, and whether heavy rain or snow will be a problem. From such factors, MET can then advise that meteorological measurements will be needed to support these activities. For example:

     a. Aerosol density in the atmosphere or optical scintillation measurements may be needed for a test involving lasers.

     b. Meteorological data used in radar calibration may be needed for a test using or testing radar.

     c. Measurements of upper air winds and turbulence could be needed for a test using UAVs.

     d. Weather advisories would be wise to have during test activities.

MET automates the original weather/meteorological checklist into a system in which the questions are presented on the computer monitor for decision, help is provided by way of a computer-stored text file for each question, and the answers are stored in computer memory until the sessions end, when a report including all input answers is produced. The report is displayed on the computer monitor and printed on the line printer, under operator control.

G-6 BENEFITS/USE.

The benefit of using the MET system is that the test officer becomes better informed about available support from the ASL weather station, and learns what weather conditions require special preparation. The test officer can then more likely plan the test so as to produce a more accurate result, and will be able to write a more correct and informative report. This all adds up to savings in time and money.

G-7 DEVELOPMENT STATUS.

MET has been developed only to the initial evaluation stage. In this prototype version, MET has been placed on 10 microcomputers in the USAEPG and ASL offices at Fort Huachuca, so as to be available for use by all test officers. Statistics on system usage and comments on deficiencies or possible improvements have not yet been collected.

G-8 FUTURE.

After evaluation, the MET prototype will be modified to eliminate any discrepancies found, and to enhance the system's capabilities to better serve test officer needs. Questions will be improved to clarify their meaning. The MET help file will be changed, as needed, to make explanations more useful to the user. The sequence of questions presented to each test officer will be determined by previous answers to prevent redundancies.

This Page Is Intentionally Left Blank

APPENDIX H

CANDIDATE TECHNICAL TEST ISSUES AND SAMPLE SUBTEST FOR PRIDE

H-1   CANDIDATE TECHNICAL TEST ISSUES.

H-1.1 UNREACHABLE OBJECTS.

Description: All knowledge base objects are reachable and contribute to some solution path.

Rationale: Unreachable objects add to maintenance problems. They may appear to encode important knowledge but in fact may never be used due to dependence on sets of conditions which cannot be met. They are comparable to dead code in conventional, procedural software.

Procedure: A list of all object names is created. This list is matched against trace results from execution, using either built-in trace facilities or a version of the knowledge base modified to provide such a trace. The comparison may be automated with conventional code. As test problems are executed, the objects invoked are compared against the list and those traversed are marked or removed from the list. Untested objects become the target of new test problems or of analysis to determine why they were excluded from their intended solution paths. The resultant suite of test problems becomes a part of the development environment for use in regression testing.

H-1.2 KNOWLEDGE AUDIT TRAIL.

Description: All knowledge structures in the knowledge base have knowledge source information entered.

Rationale: As the knowledge base and the problem domain evolve, changes may introduce inconsistencies or other conflicts. These are extremely difficult to resolve without source information. Ideally this information will resolve sources to individual expert, but must at least identify knowledge as originating either from (specific) publications, subject matter experts, testing, or field or other change requests.

Procedure: At a minimum this may be verified by examination of source slots in the knowledge base structures. It may extend to verification of source data through review of publications or formal approval by involved subject matter experts.

H-1.3 DEVELOPMENT/RUNTIME COMPARISON.

Description: All test problems produce identical results in the development and run-time environments.

Rationale: Development environments are typically much more complex software tools than the corresponding run-time environments. This arises from the

editing, debugging, and other development facilities provided. It also means that the potential for error in this software is greater, and may mean that the two environments are actually designed differently. It is necessary to ensure that testing done in the development environment produces results directly comparable to the run-time environment.

Procedure: This issue may be resolved by running test case suites in both environments and comparing results. Once a procedure has been established, this may be largely automated, assuming the respective tools have a capability for executing test cases in stand-alone mode, as opposed to interactive mode. Result comparison may be achieved by comparing output files. If a stand-alone mode does not exist it may be necessary to create one using a commercial software 'test harness' which allows keystroke capture and system execution using the capture file.

H-1.4 VARIABLE USAGE.

Description: All variables are used, and all variable value ranges are accounted for within the knowledge base.

Rationale: This is similar in intent to the first issue on knowledge objects. Each variable should be part of some path in the system, otherwise it is either wasted resource or important data which is unused. More stringent is the requirement that all ranges be accounted for. This is important as various combinations of variable values not in the knowledge base may lead to anomalous behavior, or to failure of the system. In any event they will leave the maintenance technician either unaided or possibly misguided, unacceptable in a time-critical situation. While a prototype system may be excused such gaps, a fielded system cannot afford them. If a diagnostic rule requires knowing whether a reading is between -10 and +10, there should be an object in the system which provides some action, at least an operator notice of system ignorance, when the value is not in the range.

Procedure: Testing for this issue may be done by using conventional software to extract variable occurrences and associated values from the knowledge base. Initially these may be used in constructing test cases, and comparing the variable list to trace output to determine which variables have not been used. Test cases should also be constructed to determine or verify system behavior for variable values not found in the existing knowledge objects. Where necessary objects may be created to provide a behavior for such values.

H-1.5 OBJECT DEVELOPMENT HISTORY.

Description: Each object in the knowledge base has some development history and configuration or version information as part of its documentation.

Rationale: Tracking versions as the knowledge base evolves is critical to efficient use of development resources. It should be possible to determine at a glance whether a given object has been updated as part of a new version, or since a given change was entered. This will generally involve both a simple history line entered as part of a given change and, for a new version to be distributed, some version identification with each object. Ideally this could

be automated in part as a function of the editing process. The resolution, i.e., whether each object would be tracked by a version number or each group of objects, or the knowledge base as a whole, depends on the tools available and degree of control required.

Procedure: This is a visual inspection test, much like the source information. If suitable numbering conventions are established it may become possible to automate this inspection with software which scans objects for specified date or version strings.

The following is an example of a portion of a test related to maintainability and flexibility. Maintainability is concerned with how easy the software is to repair and flexibility is how easy it is to change.

H-2 SAMPLE SUBTEST.

The following is an example of a portion of a test plan related to maintainability and flexibility as assessed by issue two, above. Maintainability is concerned with how easy the software is to repair and flexibility is how easy it is to change.

H-3   KNOWLEDGE AUDIT TRAIL

This issue addresses how well the complete, documented system allows a person to follow the acquisition and incorporation of the knowledge to perform a change to the software.

    a. Criteria: The maintainer should be able to trace the source of knowledge for objects and other knowledge constructs to be able to effectively incorporate a change or update to the system.

    b. Data Required:

        (1) Comments or documentation of objects.

        (2) Log of knowledge acquisition sessions

        (3) Sample of changes that might occur in the future.

    c. Data Acquisition Procedure:

        (1) Visual examination of the comment fields/slots of all objects.

        (2) A program that examines the existing code checking for blank, empty, or unused comment slots of objects.

    d. Analytical Procedure. Count any object without a comment field:

        (1) A sampling of x percent of the objects will be reviewed to see if the comments relate to knowledge acquisition session.

(2) A mock change will be attempted using the sample changes to see if the changes could be made in context of the original reasoning used in constructing the knowledge base and objects.

    e. Evaluation Criteria

(NOTE: Evaluators are usually distinct from the testers - testers collect the data - evaluators determine if the system met the criteria.)

System is considered deficient if any object is not commented and/or comments are not helpful in performing changes to the system.

APPENDIX I

DISTRIBUTION

Addressee	Number of Copies
Director	
U.S. Army Material Systems Analysis Activity	
ATTN: AMXSY-CA	1
Aberdeen Proving Ground, MD 21005-5071	
Commander	
U.S. Army Test and Evaluation Command	
ATTN: AMSTE-TA-W	1
ATTN: AMSTE-TC-M	3
ATTN: AMSTE-TA	6
ATTN: AMSTE-TO	2
Aberdeen Proving Ground, MD 21005-5055	
Commander	
Defense Technical Information Center	
ATTN: FDAC	2
Cameron Station	
Alexandria, VA 22304-6145	
Commander	
U.S. Army Cold Regions Test Center	
ATTN: STECR-TM	1
APO Seattle, WA 98733-5000	
Commander	
U.S. Army Combat Systems Test Activity	
ATTN: STECS-DA-M	2
Aberdeen Proving Ground, MD 21005-5000	
Commander	
U.S. Army Dugway Proving Ground	
ATTN: STEDP-PO-P	1
Dugway, UT 84022-5000	
Commander	
U.S. Army Electronic Proving Ground	
ATTN: STEEP-TD	1
ATTN: STEEP-ET	1
ATTN: STEEP-DT	1
ATTN: STEEP-MO	4
Fort Huachuca, AZ 85613-7110	
Commander	
U.S. Army Jefferson Proving Ground	
ATTN: STEJP-TD-E	1
Madison, IN 47250-5000	

Addressee	Number of Copies
Commander U.S. Army Tropic Test Center ATTN: STETC-TD-AB APO Miami, FL 34004-5000	1
Commander U.S. Army White Sands Missile Range ATTN: STEWS-TE-A ATTN: STEWS-TE-M ATTN: STEWS-TE-O ATTN: STEWS-TE-PY White Sands Missile Range, NM 88002-5000	1 1 1 4
Commander U.S. Army Yuma Proving Ground ATTN: STEYP-MSA Yuma, AZ 85634-5000	2

This Page Is Intentionally Left Blank